A SEMANTICS-BASED APPROACH TO CONSTRUCTION COST ESTIMATING

by

Mehrdad Niknam

A Dissertation submitted to the Faculty of the Graduate School,
Marquette University,
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy

Milwaukee, Wisconsin

December 2015

ABSTRACT

A SEMANTICS-BASED APPROACH TO CONSTRUCTION COST ESTIMATING

Mehrdad Niknam

Marquette University, 2015

A construction project requires collaboration of different organizations such as owner, designer, contractor, and resource suppliers. These organizations need to exchange information to improve their teamwork. Understanding the information created in other organizations requires specialized human resources. Construction cost estimating is one of the processes that requires information collected from several sources including a building information model (BIM) created by designers, estimating assembly and work item information maintained by contractors, and construction resource cost information provided by resource suppliers. Currently, it is not easy for computers to integrate the information for construction cost estimating over the Internet.

This study discusses a new approach to construction cost estimating that uses the Semantic Web technology. The Semantic Web technology provides a data modeling format and the required infrastructure that enables accessing, combining, and sharing information over the Internet in a machine processable format. The estimating approach presented in this study relies on BIM, estimating knowledge, and construction material cost data to be represented in the Semantic Web. The approach presented in this study makes the various sources of cost estimating data accessible as Simple Protocol and Resource Description Framework Query Language (SPARQL) endpoints or semantic web services. This study presents an estimating approach that integrates distributed information provided by project designers, contractors, and material suppliers for preparing cost estimates. The purpose of this study is not to fully automate the estimating process but to streamline it by reducing human involvement in repetitive cost estimating activities.

ACKNOWLEDGMENTS

Mehrdad Niknam

This dissertation has been made possible with the guidance and support of many individuals. I owe my deepest gratitude to them all who contributed to the completion of this work.

I would like to sincerely thank my advisor, Dr. Saeed Karshenas, for his guidance, instruction, understanding, patience, encouragement, and providing an excellent atmosphere for research work. I would like to mention that the result of this dissertation would not be possible without his help and his ideas among which is the idea of defining a shared ontology for building information modeling that acts as a semantic bridge between different domains. I would also like to thank him for always being there for me through the toughest and the happiest times in my life.

I would like to extend my thanks to my dissertation committee members, Dr. Mani Golparvar Fard, Dr. Iqbal Sheikh Ahamed, and Dr. Javier Irizarry. I am privileged to have worked with this excellent interdisciplinary team of researchers. This work has been made much better because of their thoughtful suggestions and recommendations.

I must also thank the Graduate School and the Department of Civil, Construction and Environmental Engineering at Marquette University for their financial support.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

The construction cost estimating process is a time consuming process that includes a number of repetitive activities to integrate information from different sources. Automating some of the human-involved activities in the cost estimating process can improve the efficiency of cost estimators. In order to make the repetitive work involved in the cost estimating process computer processable, it is argued in this dissertation that the formats for representing construction data must change. Currently construction data are represented in text, relational, object-oriented, or XML formats. To facilitate exchanging and integrating information, the semantic data format is suggested and applied to the estimating data. The purpose of this dissertation is not to fully automate the estimating process but to streamline it by reducing human involvement in repetitive cost estimating activities. In order to better understand the time consuming aspects of construction cost estimating, in the following section the current computerized cost estimating process is briefly explained.

1.1 Current Estimating Applications

Current computerized construction cost estimating applications break a construction project into assemblies of work items. An estimating assembly represents a building element and includes the work items that must be completed for the construction of the building element. For example, an estimating assembly for a footing element includes work items such as forming, reinforcing, and placing concrete. The cost of an assembly is the sum of the work item costs included in the assembly. Figure 1-1 shows the general architecture of the current estimating applications. As Figure 1-1 shows, a cost estimating application keeps built-in databases of estimating assemblies and work items along with the work item properties such as crew make-ups, crew productivities, and resource (material, equipment, and labor) costs.

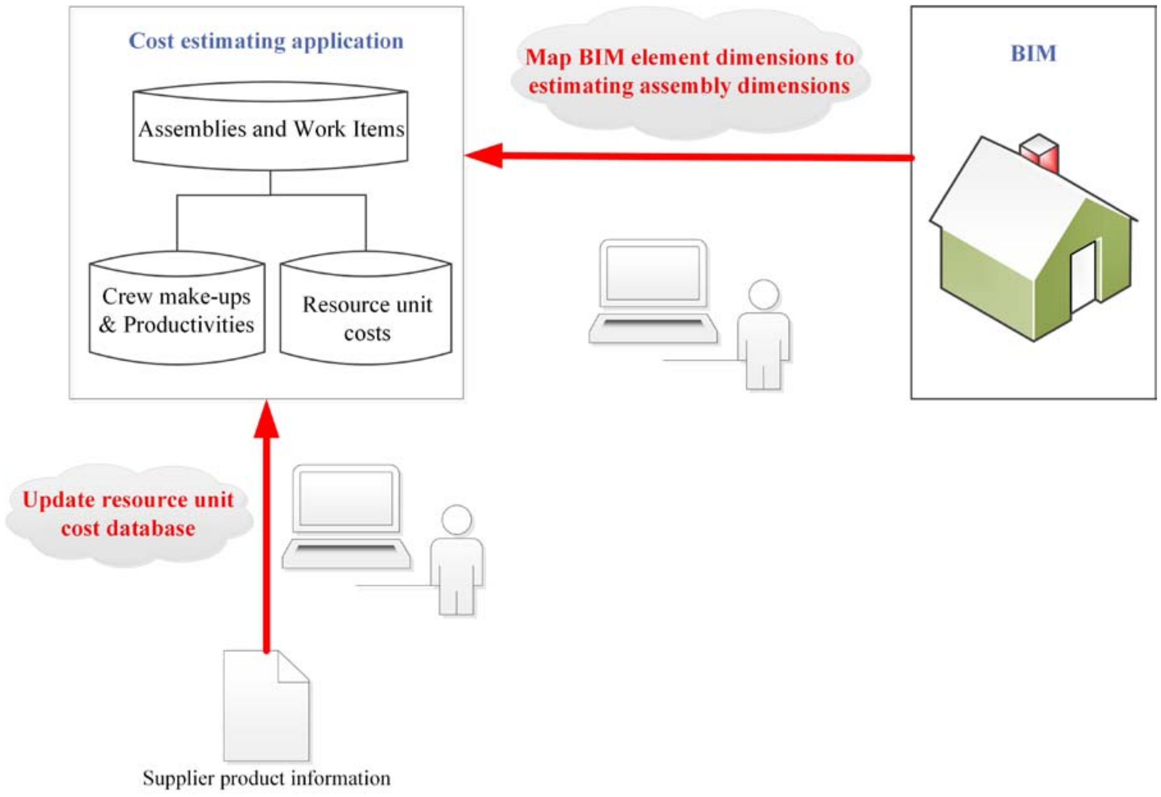

Cost estimating application
Assemblies and Work Items
Crew make-ups & Productivities
Resource unit costs
Map BIM element dimensions to estimating assembly dimensions
BIM
Update resource unit cost database
Supplier product information

Figure 1-1: Current estimating application

To estimate the cost of a work item, an estimator needs to (A) obtain the required information from the design model of the project to calculate work item quantities and (B) update unit costs of the work item resources based on the information from suppliers. These steps are time consuming as explained below:

A) Obtaining the required information from the design model of the project to calculate work item quantities

With the advent of 3D modeling, a model of a building can be digitally represented which is referred to as a Building Information Model (BIM) (Azhar 2011; Cerovsek 2011; Teicholz et al. 2011; Gu and London 2010). Currently, estimators can digitally extract information from a BIM and transfer it to an estimating application in order to calculate work item quantities. Current commercial estimating applications use proprietary add-in programs for this purpose. An add-in program is designed to retrieve information from an application. For example, WinEst (WinEst 2015) estimating software has an add-in program that transfers BIM element properties from an Autodesk Revit (Autodesk 2015a) model to a WinEst spreadsheet.

To transfer information from BIM to an estimating software, an estimator maps the information as shown in Figure 1-2. The mapping process involves 2 steps as follow:

1) Map BIM elements to their corresponding estimating assemblies. An example is mapping a rectangular footing element in BIM to a spread footing assembly in the estimating software.
2) Map BIM element dimensions to the estimating assembly dimensions for calculating work item quantities. For example, mapping the dimensions of a BIM footing element to the corresponding estimating assembly dimensions. The estimator needs to repeat the mapping process for each building element

and element dimension (e.g. length, width, and thickness) separately which is a time consuming process.

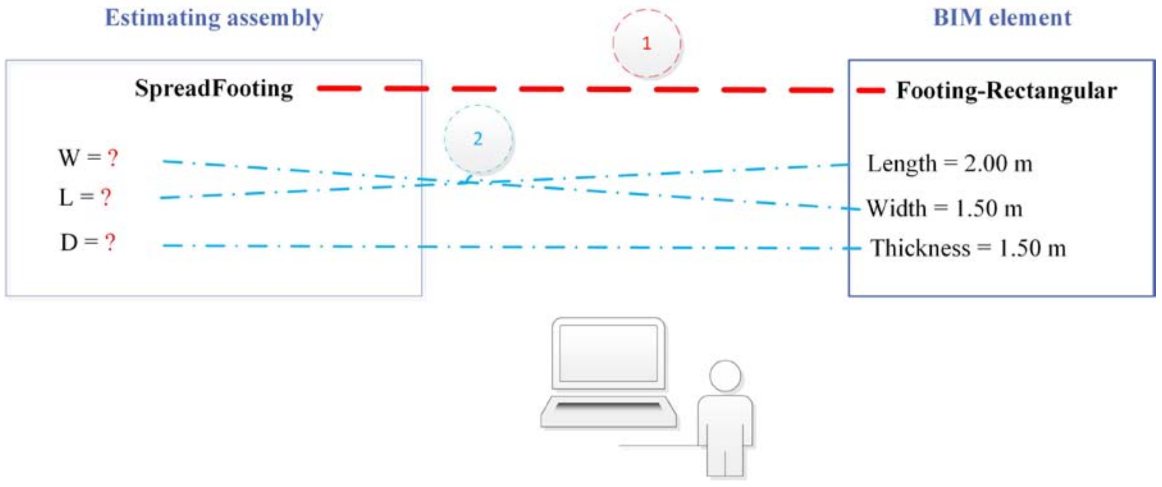


Figure 1-2: Mapping information from a BIM element to an estimating assembly

Among the 2 steps mentioned above, step 1 requires estimator judgement and cannot be easily automated. For example, there might be some rectangular footings in a building project which may or may not need formwork. This means that there should be 2 different assemblies in the estimating application one with and one without formwork work item. Therefore, generally human judgement is required for executing step 1. On the other hand, step 2 can be easily automated if estimating assemblies and BIM elements are semantically defined, which is one of the objectives of this study.

B) **Updating unit costs of work item resources based on information from material suppliers**

Another time consuming activity in construction cost estimating is updating an estimating application's resource cost databases. Current estimating applications keep built-in databases of resource unit costs. Since resource costs are affected by economic conditions and change based on supply and demand, an estimating application's unit-cost databases must be updated before performing a new estimate. Currently, resource suppliers' information is provided in a format that is suitable for human consumption and cannot be directly used by computers. That is why current estimating application resource cost databases are manually updated. The process of updating an estimating application resource cost database is time consuming and requires estimator involvement for obtaining the latest unit costs from various suppliers. One of the objectives of this study is to semantically define suppliers' product information in order to allow automated updating of estimating unit cost databases.

1.2 Current Challenges for Integrating Information from Different Sources

The estimating process requires accessing and combining data from different sources. These sources include BIM created by designers, estimating assemblies and work items created by contractors, and resource cost data provided by suppliers. Among the factors that make data sharing and integration difficult are the variety of formats used for storing data and the ambiguities created when several synonym words refer to the same entity or a term refers to several concepts (Baker and Cheung 2007).

Currently, sources of data used in estimating are created in heterogeneous data formats. For example, BIM data are in relational (e.g., ODBC), object (e.g., IFC), or XML (e.g., ifcXML, gbXML) databases, estimating assembly and work item data are usually in relational databases, and material suppliers' product data are provided in text, HTML or in XML formats. In addition to heterogeneous data formats, there are many synonyms for the same entity. For example, different databases may use spread footing, column footing, or rectangular footing to refer to a footing. Another example is the height dimension of a footing that may also be referred to as depth or thickness. Besides synonyms, the same term can be used to represent different concepts which can cause ambiguity. For example, the term "area" may be used to refer to an element's surface area or the element's formwork area.

The above mentioned challenges cause problems in integrating information from different sources.

1.3 Using the Semantic Web for Information Modeling

Representing construction cost estimating information in a computer processable format can greatly improve estimator's efficiency (Niknam and Karshenas 2013; Niknam and Karshenas 2015a). The Semantic Web (W3C Standard 2015b) aims to solve the information integration problem (Baker and Cheung 2007). The Semantic Web uses a graph structure to represent information. When information is represented as a graph, it is easy to link and merge graphs of information from different sources. Uniform Resource Identifiers (URIs) (W3C 2005) are used in the Semantic Web to globally identify each piece of data. The graph data structure of the Semantic Web along with URIs provide a global information space of interlinked data distributed on the web (Cyganiak and Jentzsch 2010). The Semantic Web technology provides the infrastructure and the data modeling format which allows computers to process, merge and combine information over the Internet.

In this dissertation, a semantics-based estimating approach is presented that combines information from a semantically defined BIM knowledge base, an estimating assembly knowledge base, and resource suppliers' semantic web services to prepare a cost estimate. This study investigates how such knowledge bases and semantic web services can be developed and how a semantics-based estimating application can access these distributed sources of information over the Internet to prepare a cost estimate. This study also investigates how a semantics-based estimating application can reduce human involvement in the estimating process by streamlining the mapping of BIM elements to estimating assemblies, and performing estimating material resource cost database updates.

1.4 Organization of Dissertation

This dissertation is organized into 7 chapters. Following the present introductory chapter, Chapter 2 will present current challenges for integrating distributed sources of data for construction cost estimating. Chapter 2 introduces relational databases, object-oriented databases, and Extensible Markup Language (XML) as the current practices for information modeling. The limitations of the current approaches for transferring and integrating information between different domains will be discussed. In chapter 2, the advantages of the Semantic Web technology for integrating distributed sources of data will be discussed. Also, architecture of a semantics-based estimating application will be explained which accesses distributed domain knowledge bases that are created by different domain experts: BIM knowledge bases created by designers, estimating assembly and work item knowledge bases created by cost estimators, and material suppliers' product knowledge bases created by material suppliers. Chapters 3, 4 and 5 will discuss the required ontologies and the methodologies for creating the required knowledge bases.

Chapter 6 will describe a prototype estimating application based on the estimating architecture presented in chapter 2. It will explain software implementation and use cases developed in this study. Chapter 6 will also validate the prototype estimating application by comparing it to WinEst estimating software for the time it takes to prepare a cost estimate. A summary of the dissertation, the main conclusions as well as recommendations for future work will be given in Chapter 7.

CHAPTER 2
INTEGRATING DISTRIBUTED SOURCES OF DATA FOR CONSTRUCTION COST ESTIMATING

The estimating process requires accessing and combining data from different sources. These sources include BIM created by designers, estimating assemblies and work items created by contractors, and resource cost data provided by suppliers. Currently, different sources of data used in cost estimating are created in heterogeneous data formats. For example, BIM data are stored in relational (e.g., ODBC), object (e.g., IFC), or XML (e.g., ifcXML, gbXML) databases; estimating assembly and work item data are usually kept in relational databases; and material suppliers' product data are provided in text, HTML, or XML formats. When distributed sources of data are in heterogeneous data formats, computers cannot easily combine and integrate the data.

Another factor that complicates integration of data from several sources is the use of different terminologies for referring to the same entity. For example, different databases may use spread footing, column footing, or rectangular footing to refer to a footing element in a building. Another example is the height dimension of a footing that may be referred to as depth or thickness. One more factor that complicates the integration of distributed data is using the same word to refer to different concepts. For example, the term "area" may be used to refer to an element's surface area or the element's formwork area.

In this chapter, sections 2.1, 2.2 and 2.3 discuss relational, object-oriented, and Extensible Markup Language (XML) formats for representing data and their limitations for integrating data from different domains. Section 2.4 presents the Semantic Web approach to data modeling and its advantages for combining data from several distributed sources. Section 2.5 discusses the architecture of a semantics-based cost estimating application developed in this study.

2.1 Data Stored in Relational Databases

A relational database is a digital database that organizes data into one or more tables of rows and columns. Each entity type in a database has its own table where rows represent instances of that type, columns represent properties, and cells are particular values for those properties. Relational databases usually include relations among database tables. Relational databases are useful for storing large volumes of data and easily adding or updating the data using transactions.

However, schema rigidity of relational databases causes problems for integrating data from different sources (Bergman 2009). Once the data schema and data relationships are set, it is not easy to change them. Different sources of data in different domains usually have different data schemas. This means that if one needs to integrate data from two different domains, he/she cannot easily transfer data from one domain database to the other because of differences in database schemas.

To integrate data from two domains when data are stored in two separate relational databases, one needs to know the schemas of both databases. This is not always possible considering the fact that the schemas of relational databases are local and cannot be easily accessed. Even if one knows schemas of different domains, he/she has to write very complicated queries to combine and integrate the databases (Alexander 2013), which is very time consuming and cannot be easily automated. It is also worth mentioning that combining different domain databases would result in data duplication.

According to W3C, some of the challenges for integrating data from relational databases are (Malhotra 2007):

- Relational databases are relatively flat with little semantic information. Data is often fragmented and although Key and Foreign Key relationships are indicated in the catalog, these are merely hints about the semantics of the data.
- Relational databases are often insular, built for a single purpose with access restricted to a particular group.
- Corresponding or analogous data in one database might be represented in a different number of tables in other databases.
- Corresponding or analogous data in one database might be represented in a different number of columns in other databases.
- Corresponding or analogous data in one database might be represented in a different number of rows in other databases.
- Corresponding or analogous data in one database might be represented using different values or vocabularies in another database; or the same values in the various databases might mean different things.

The above mentioned challenges mean that even if all distributed sources of data needed for construction cost estimating are stored in relational databases, combining and integrating the data would be very challenging.

2.2 Data Stored in Object-Oriented Databases

An object-oriented database is a digital database in which information is represented as objects where each object has some attributes (properties). Object-oriented databases usually map very well to the object models of the programs that use them (Cunningham & Cunningham 2014). So, the programmer can maintain consistency between a database management system and the programming language. However, object-oriented databases have limitations. According to W3C, Some of the limitations of object-oriented systems for information sharing are (SWBPD 2006):

- The domain schema is local and cannot be shared on the Internet and among computer applications.
- It is not easy to dynamically modify data schemas because data schemas should map very well to the object models of the programs that use them. Any changes in data schema require revising the software programming code written for the schema.
- To model the same objects in two different domains, each domain develops its own class hierarchy. For example, the same building would be modeled with two different class hierarchies in the BIM and the estimating domains. So, the same element in the same building would belong to two different classes that cannot share instances and their properties.

The above-mentioned limitations make it difficult to exchange and combine AEC information stored in object databases.

2.3 Data Stored as Extensible Markup Language (XML)

Internet is a platform for exchanging information. When two organizations work with each other, they establish Internet connections for different purposes. Extensible Markup Language (XML) (http://www.w3.org/XML/) is a serialization format that enables different programs and computers to communicate with each other over the Internet. XML allows business partners to develop fast and reliable communication platforms. In AEC domains, XML standards have been developed to facilitate information exchange. Some of the common standards are listed below:

- **ISO 10303-28 (STEP-XML)** (http://www.steptools.com/library/standard/) is a standard data model for exchanging product digital data. 3D objects in Computer Aided Design (CAD) can be represented in STEP. It enables businesses to describe products and exchange product information independent of the system being used.
- **ifcXML** (http://www.buildingsmart-tech.org/) is developed by buildingSMART to describe building and construction industry data. It's a commonly used collaboration format in building information modeling. buildingSMART publishes IFC specifications in different formats including XML. IFC is an official ISO standard.
- **AecXML** working group was formed to develop schemas for the exchange of AEC-specific business-to-business information (http://xml.coverpages.org/aecXML.html). Later, Associated General Contractors of America (AGC) funded AgcXML (http://agcxml.org) project as part of AecXML domain to ensure compatibility with related efforts. AgcXML project, managed by National Institute of Building

Sciences, enables efficient and reliable exchange of transactional data between architects, engineers, contractors, subcontractors, material suppliers, and building owners (http://agcxml.org). AgcXML allows design and construction professionals to exchange information such as owner/contractor agreements, schedules of values, requests for information (RFIs), requests for proposals (RFPs), architect/engineer supplemental instructions, change orders, change directives, submittals, applications for payment, and addenda.

- **Construction IT Alliance eXchange (CITAX)** was an Irish construction industry project to define universal set of XML message standards to allow suppliers and contractors exchange information with each other (Hore and West 2007). The overall aim of the project was to facilitate business transactions between companies in the Irish construction industry.
- **Construction Operations Building Information Exchange (COBie)** (http://www.nibs.org/?page=bsa_cobie) is an information exchange specification for the life-cycle capture and delivery of information needed by facility managers. COBie can be used during design, construction and maintenance of a project. Aim of COBie is to assist the facility manager to maintain, operate, and track assets within the building. COBieLite is a lightweight XML format for COBie, which was published by buildingSMART in April 2013.
- **CityGML** (http://www.opengeospatial.org/standards/citygml) is an XML data format for storing and exchanging 3D city models. CityGML defines

> entities, attributes, and relations of a 3D city model with respect to their geometrical and topological properties. It can be used in different applications such as simulations, urban data mining, facility management, and thematic inquiries.

XML standards address structural and syntactic issues but not the semantics of information. XML is a serialization format for encoding information so that it can be parsed when it is passed between machines; the challenge is that other programs in order to be able to read an XML file require a special programming code (Cambridge 2015). Using XML standards can facilitate the information integration task; however, standards have built-in flexibilities that require developers' great effort to come to an agreement on how to exactly implement the standard in XML (Bussler 2002). XML allows representing the same data in different ways (Sequeda 2012; Berners-Lee 1998). For example, consider a wall element that has a length property. An XML representation of the wall is:

```
<element>
<title>wall001</title>
<length> L1</length>
</element>
```

Other XML representations of the same information are:

```
<element title="wall001">
<length> L1</length>
</element>
```

or,

```
<length>
<element>wall001</element>
<name> L1</name>
</length>
```

or,

```
<document href="wall001">
<length> L1</length>
</document>
```

or,

```
<document>
<details>
 <element>href="wall001"</element>
 <length>
 <name> L1</name>
</length>
</details>
</document>
```

or,

```
<document>
<length>
<element>href="wall001"</element>
<details>
<name> L1</name>
</details>
</length>
</document>
```

or,

```
<document href="wall001" length=" L1" />
```

This means that if one domain creates an XML representation of their information, other domains require custom program coding to read that information. The temporary nature of construction projects makes this even more complicated. Organizations involved in construction projects reorganize from project to project, which makes it difficult to establish and maintain XML based Internet connections between organizations working together on a temporary basis.

In XML documents, information is modeled as a tree with a root. This makes it difficult to link data across XML documents. Xlink was an XML linking language to link information between XML documents that failed to gain adoption. Figure 2-1 shows an example of integrating XML information between two different domains. At the top of Figure 2-1, two domains named domain A and domain B represent their information in XML; in the case of construction projects, these domains may be the design and cost estimating domains. In Figure 2-1, node X may represent a footing element with design properties such as length, width, and thickness. The same footing has estimating properties such as costs of required resources. To integrate XML information related to object X between two domains, custom programming is necessary. One reason is that synonyms may be used in two different domains to represent the same entity. Also, integrating XML information from two sources requires transferring information from both sources to a new document which results in information duplication.

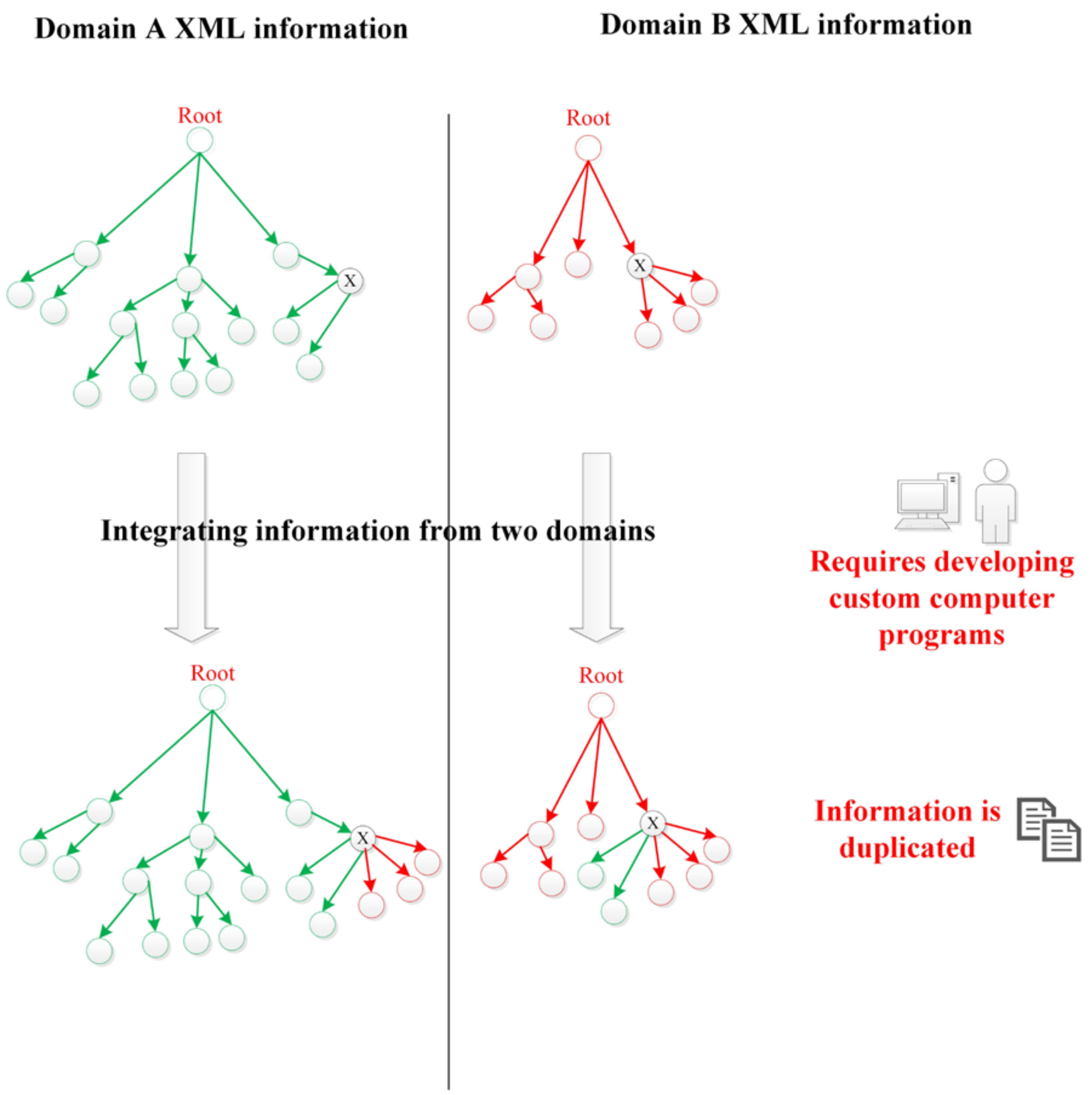

Domain A XML information
Domain B XML information
Root
Root
X
X
Integrating information from two domains
Requires developing custom computer programs
Root
Root
X
X
Information is duplicated

Figure 2-1: XML-Integrating information from two domains

The following section introduces the Semantic Web technology and the advantages of the Semantic Web for integrating information from different domains without information duplication.

2.4 The Semantic Web

The current web provides the infrastructure for distributed network of web pages that refer to each other using Uniform Resource Locators (URLs), which is suitable for human consumption. The Semantic Web is a network of linked (connected) data that are machine processable (Allemang and Hendler 2011). The Semantic Web uses a graph data structure in which each node is an instance that is pointing to other nodes. So, a semantic representation of a construction project enables project participants to represent their information in a graph data structure which allows easily connecting and combining their information about the project (Niknam and Karshenas 2014). The Semantic Web creates linked distributed information and enables computers to search for and find data distributed on the Internet. The Semantic Web allows creating data models, drawing meaningful conclusions, and sharing information on the web and between computer applications (Hitzler et al. 2011). It allows sharing model schemas and enables computer applications to process and draw conclusions on data that are created in other sources (Knublauch et al. 2006).

Resource Description Framework (RDF) (W3C RDF Working Group 2014) is a standard data model in the Semantic Web. There has always been misconception between XML and RDF. Both XML and RDF are used to represent structured data on the Internet and move data between computer applications. But, there is a difference between XML and RDF: XML is a syntax; whereas, RDF is a data model (Sequeda 2012). RDF has several syntaxes such as Turtle, N3, N-Triples, and XML. The XML syntax is referred to as RDF/XML (W3C 2004). RDF/XML was the first standard format for serializing RDF. The followings are the advantages of RDF (Sequeda 2012):

A) RDF data models are interpretable by computers.

RDF represents information in triples. When information is represented in triples, there is only one way of representing information. The following shows RDF triple for the wall element length example that was given in XML:

Subject	**Predicate**	**Object**
Wall001	hasLength	L1.

When information is represented in RDF triples, it eliminates the need for custom programming to access information and interpret changes in RDF triples. In RDF, even the data schema are represented by RDF triples. This makes it possible to modify and extend data schema or add different attributes without requiring changes in programming code of applications that access the information.

B) RDF information can be easily linked and combined

RDF triples can be seen as a graph structure to represent information. When information is represented as a graph, it is easy to link graphs of information from different sources. RDF uses Uniform Resource Identifiers (URIs) (W3C 2005) to uniquely identify each piece of data. The graph structure of RDF along with data URIs provides a global information space of interlinked data distributed on the web (Cyganiak and Jentzsch 2010).

Figure 2-2 shows an example of integrating information between two different domains in RDF. Using RDF, no custom programming is necessary to integrate information. When X has the same URI in two different domains, graphs of RDF information automatically merge without creating any duplication.

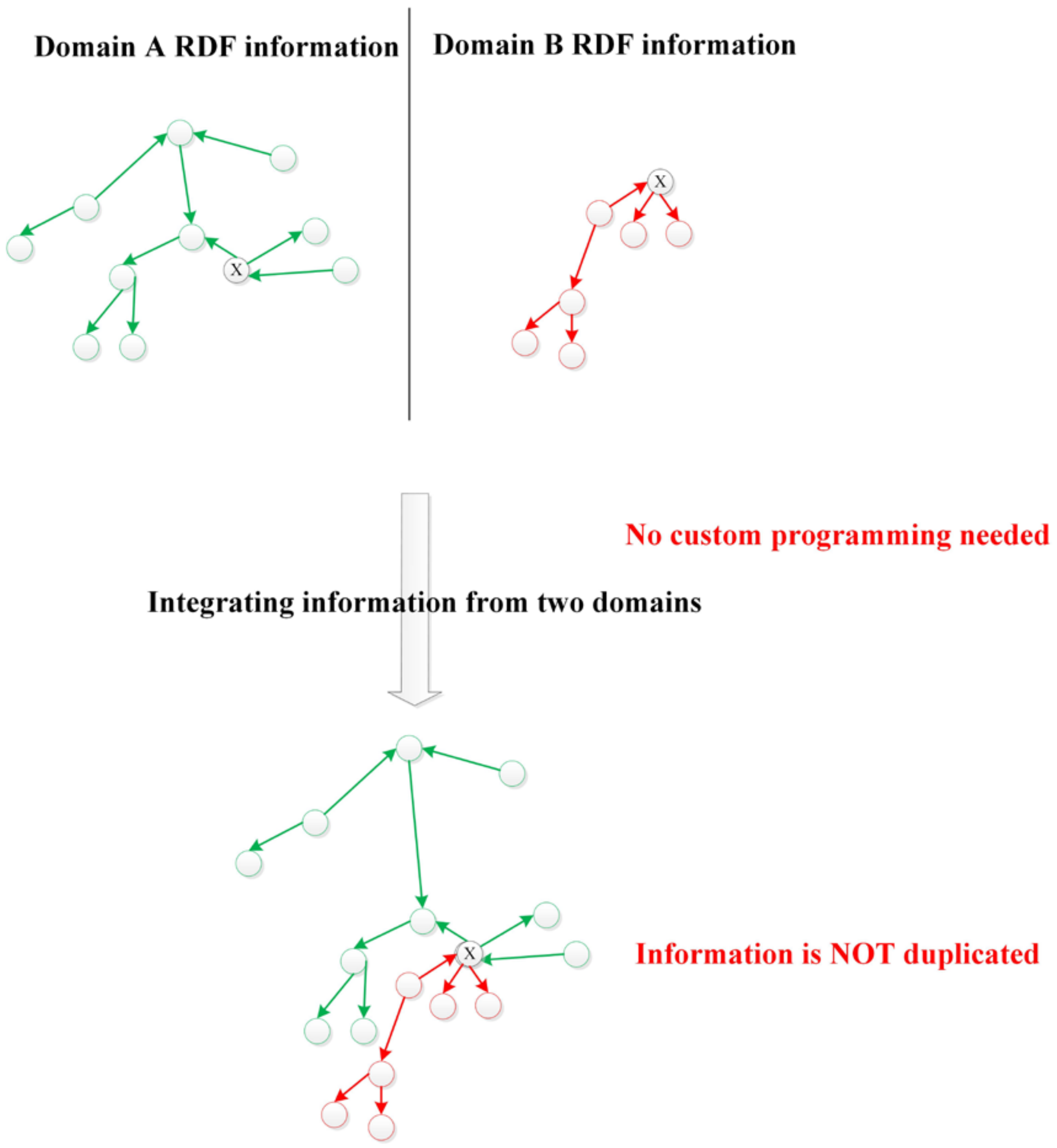

Domain A RDF information
Domain B RDF information
X
X
No custom programming needed
Integrating information from two domains
X
Information is NOT duplicated

Figure 2-2: RDF-Integrating information from two domains

In the Semantic Web, ontologies (W3C Standard 2015a) are used to describe the organization of information in a domain. Ontologies explicitly define the concepts, relationships among the concepts, and the terminology used in a domain of interest (Gruber 1993). Ontologies can be imported and used for knowledge representation, which gives computers awareness of the organizations of the information distributed over the web. A domain ontology together with a set of the domain instances constitute a domain knowledge base (Noy and McGuinness 2001); what can be expressed with an ontology is stored and used in a knowledge base. The Semantic Web is designed to create distributed knowledge-based systems (Obitko 2007).

2.5 A Semantics-Based Cost Estimating Approach

An estimating application must be able to access and use design, estimating, and resource information across organizational, specialty, and geographic divides. In the Architecture, Engineering and Construction (AEC) industry, design knowledge is provided by design companies, cost estimating knowledge is maintained by construction companies, and material resource knowledge is provided by material suppliers. So, a flexible estimating architecture must allow accessing and using independently created domain knowledge distributed over the Internet.

The Semantic Web (W3C Standard 2015b) technology is used in this study for sharing and integrating information distributed over the Internet. Figure 2-3 shows the Semantic Web based estimating architecture developed in this study. The estimating application shown in Figure 2-3 uses distributed domain knowledge bases that are created by different domain experts; for example, designers create BIM knowledge bases, construction cost estimators create estimating assembly and work item knowledge bases, and material suppliers provide material cost knowledge bases.

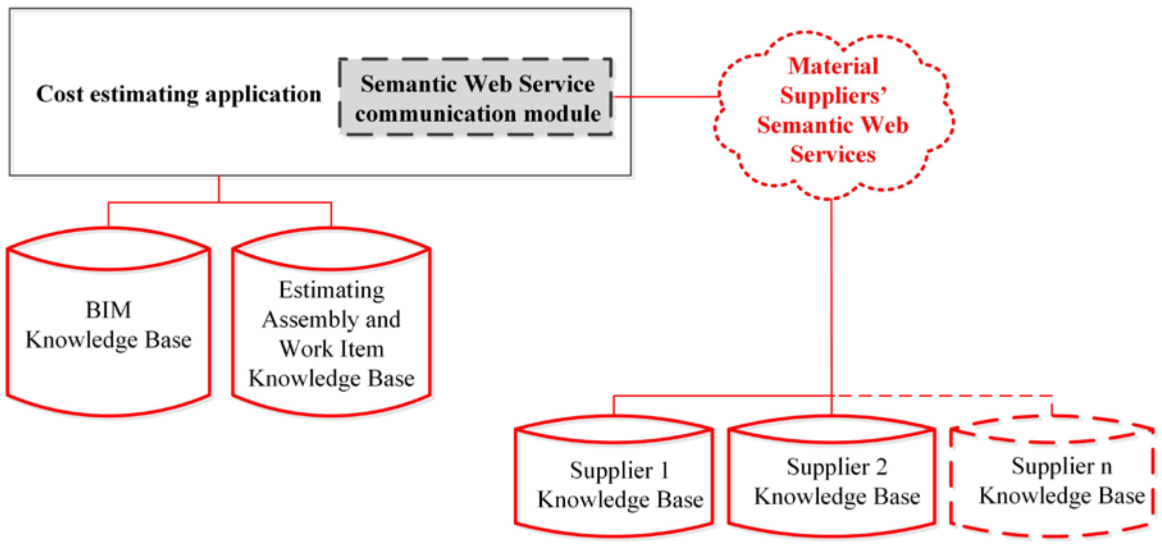

Cost estimating application
Semantic Web Service communication module
Material Suppliers' Semantic Web Services
BIM Knowledge Base
Estimating Assembly and Work Item Knowledge Base
Supplier 1 Knowledge Base
Supplier 2 Knowledge Base
Supplier n Knowledge Base

Figure 2-3: Semantics-based estimating architecture developed in this study

A domain knowledge base can be developed and accessed as a Simple Protocol and Resource Description Framework Query Language (SPARQL) endpoint (Prud'Hommeaux and Seaborne 2008) or a semantic web service (Martin et al. 2004b). In this study, BIM and estimating assembly and work item knowledge bases are created as SPARQL endpoints and material suppliers' knowledge bases are created as semantic web services. SPARQL is a semantic query language that is able to retrieve and manipulate data in a knowledge base. A SPARQL endpoint knowledge base is directly queried by the estimating application whereas accessing a semantic web service requires a communication module. A prototype estimating application is developed in this study that can access these knowledge bases to create an estimate.

Ontologies are needed to define the organization of information used by the semantics-based estimating application. A number of methods have been proposed to create ontologies such as Uschold and King (Uschold and King 1995), Gruber (Gruber

1995), METHONTOLOGY (Fernández-López et al. 1997), On-To-Knowledge (Sure et al. 2004), DILIGENT (Pinto et al. 2004), and NeOn (Suárez-Figueroa 2012; Suárez-Figueroa et al. 2012) methodologies. In this study, the NeOn methodology is used because of the flexibility it provides for a variety of scenarios instead of prescribing a rigid workflow. NeOn methodology is a scenario based methodology which emphasizes reuse of ontological and non-ontological resources, the reengineering and merging, and taking into account collaboration and dynamism (Suárez-Figueroa et al. 2011). The NeOn methodology divides the general problem of ontology development into nine sub-problems known as NeOn scenarios (Suárez-Figueroa 2012). Solutions to different NeOn scenarios are combined to obtain a solution to a general problem.

The estimating architecture shown in Figure 2-3 requires ontologies that define the estimating domain knowledge. Instead of developing a large ontology that covers all estimating domain concepts, a separate ontology for each of the knowledge bases shown in Figure 2-3 is developed. An ontology network is developed that consists of an ontology for the BIM knowledge base, an ontology for the estimating assembly and work item knowledge base, and a set of ontologies for material suppliers' semantic web services.

NeOn scenario 1 documents the ontology requirements that define the purpose, scope and implementation language of the ontology. The requirements for the ontologies developed in this study are discussed in chapters 3, 4, and 5 of this dissertation. Ontologies are implemented in Resource Description Framework (RDF) (W3C RDF Working Group 2014) and Web Ontology Language (OWL) (W3C Standard 2015a). RDF is a standard data model in the Semantic Web and is used for data interchange on the web. RDF uses subject-predicate-object triples to represent information which can be seen as a graph of data; it facilitates merging data from sources with different underlying schemas and supports the evolution of data schemas over time without requiring changes in the applications consuming data. OWL extends RDF to build ontologies and to enhance the Semantic Web with reasoning power of description logic. RDF and OWL are different layers of the semantic web that work together to create ontologies, merge data distributed over the internet, support the evolution of data schemas, and reason on data from different sources. RDF and OWL enable creating and connecting knowledge bases distributed over the Internet. In this study, Protégé (Stanford University 2015) software is used to code the ontologies in RDF and OWL. Protégé is an open source java tool that provides an extensible architecture for the creation of customized ontologies and knowledge bases.

NeOn scenario 2 requires using non-ontological resources available when developing a domain ontology. Non-ontological resources include published documents in the domain of interest for which the ontology should be developed. In this study, non-ontological resources related to the estimating domain are used, which include cost estimating books, and cost estimating references such as RSMeans reference books (RSMeans 2015), UNIFORMAT II classification system (ASTM standard 2015), and CSI MasterFormat (CSI 2015).

NeOn scenarios 3, 4, and 5 require reusing, reengineering, and merging existing ontological resources available, respectively. Scenario 6 suggests both reengineering and merging existing ontological resources if necessary. In this study, there was no need to reengineer or merge any existing ontologies. The existing ontological resources that are used are:

A) **QUDT ontology** (Hodgson et al. 2011): QUDT expresses quantities and units of measurement. It was developed for the NASA Exploration Initiatives Ontology Models (NExIOM) project. QUDT provides a standardized and consistent vocabulary for the terminology used in science and engineering for representing units of measurements.

B) **Free Class OWL ontology (FC)** (BauDataWeb 2015): FC is developed by the European Building and Construction Materials Database to describe construction materials and services.

C) **Good Relations ontology (GR)** (Hepp 2008; Hepp 2015): GR was developed to allow businesses semantically defining their product offerings and publishing them on the web. GR provides a conceptual model for general concepts such as company, product descriptions, offer, price, payment, store location, shipment, and warranty information.

D) **OWL-S ontology** (Martin et al. 2004b): OWL-S ontology provides computer-interpretable descriptions of web services and the means by which they are accessed.

E) **Organization ontology** (W3C 2014). This ontology is designed to enable representation of information on organizations and organizational structures. Organization ontology provides a generic, reusable core ontology that can be extended or specialized for use in particular situations

The above mentioned ontologies were imported and reused as will be discussed in different chapters of this dissertation. Protégé user interface allows importing ontologies available on the Internet and reusing them when developing a new ontology. The above mentioned ontologies are mapped to the ontologies developed in this study when necessary.

NeOn scenario 7 suggests using Ontology Design Patterns (ODPs) as a guide when developing new ontologies. For ontology development in this study, no ontological design patterns were used but ontology development benefitted from guidelines and suggestions in (Stanford University 2015; Allemang and Hendler 2011; Hebeler et al. 2011a; Segaran et al. 2009a; Noy and McGuinness 2001; Hobbs and Pan 2006).

NeOn scenario 8 is needed when restructuring ontological resources. In this study, it was not needed to restructure ontological resources. NeOn scenario 9 is required when localizing ontological resources to other natural languages. Ontologies in this study are developed in English and there was no need to localize it to other natural languages.

Chapters 3, 4 and 5 will discuss the ontologies and the methodologies for creating the required knowledge bases shown in Figure 2-3.

CHAPTER 3
BIM KNOWLEDGE BASE ARCHITECTURE

This chapter presents how a building information model (BIM) can be semantically defined and saved as a BIM knowledge base. The methodology for creating the knowledge base is explained.

3.1 BIM Ontology

BIM includes information about building elements such as type, location, level, material, and geometry. Currently, designers create BIM in a BIM platform (e.g., Autodesk Revit) and then its information can be converted and stored in XML (e.g., ifcXML, gbXML), relational (e.g., ODBC) or object-oriented (e.g., IFC) databases. The aim of this chapter is to represent BIM in the Semantic Web. Ontologies are needed to define the organization of information in the Semantic Web.

The AEC-FM industry needs a standard ontology that can be used for representing BIM in the Semantic Web. A number of studies have used EXPRESS-to-OWL conversion procedures for developing an ifcOWL ontology (Karan et al. 2015; Karan and Irizarry 2015; Pauwels and Terkaj 2014; De Farias et al. 2014; Pauwels et al. 2011b; Pauwels et al. 2011a; Demir et al. 2010; Beetz et al. 2009); however, none of the ifcOWL ontologies have become a standard yet. IFC schema is also limited and does not cover all the concepts in various AEC-FM domains.

A large number of individuals and organizations are involved in AEC-FM projects. When independent individuals and organizations need to share and exchange information to achieve interoperability, no single ontology can cover all the concepts in every domain

(O'Leary 1997). Instead of developing a single large ontology to cover all concepts in various AEC-FM domains, several independent domain ontologies can be developed for different domains and used for information sharing. To convert information from one domain to another domain ontology, there must be a set of mappings between domain ontologies. Ontology alignment is the process of finding the correspondences between different domain ontologies (Segaran et al. 2009b). But, mapping a large number of concepts in several domains is not an easy task. Instead of mapping several domain ontologies directly to each other, one can define a shared ontology to map various related domains to the shared ontology (Hebeler et al. 2011b). In this method, different domains must share a set of commonly understood concepts. A shared ontology, also referred to as a foundation ontology or an upper level ontology, acts as a semantic bridge in the ontology alignment process (Mascardi et al. 2010). Integrating information from different domains can happen by establishing connections and relationships between domain ontologies and a shared ontology (Karshenas and Niknam 2013). Figure 3-1 shows how a shared ontology can be used as a semantic bridge between multiple domains. In Figure 3-1, the design and estimating domain ontologies are developed by extending a shared ontology. This facilitates the integration and connection of various AEC-FM domain ontologies.

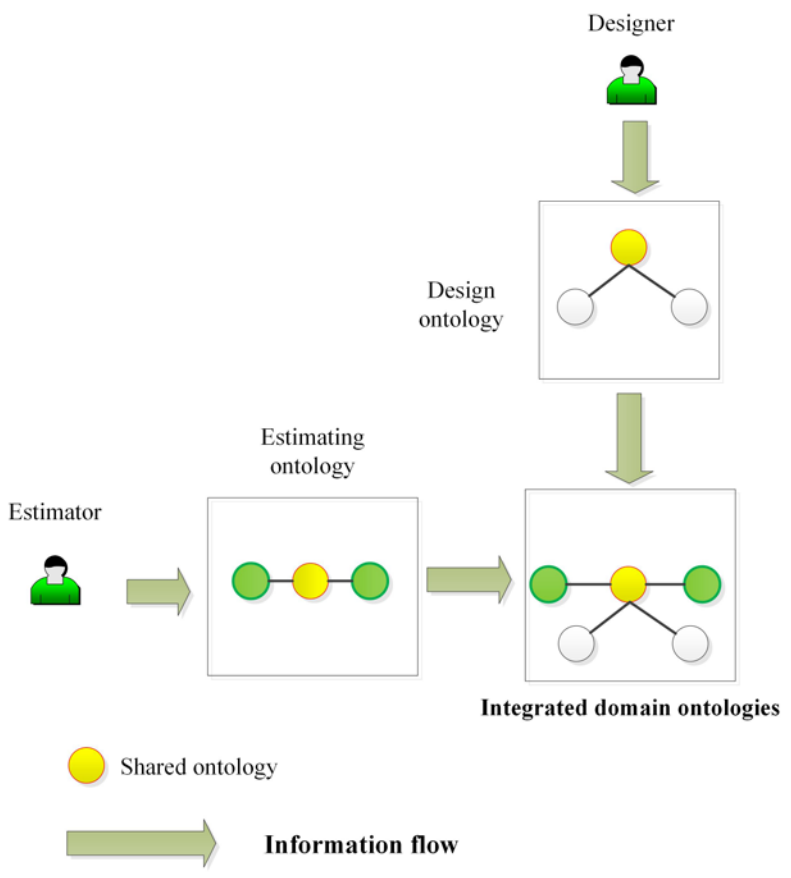


Figure 3-1: Integrating domain ontologies using a shared ontology

In this study, a shared ontology for building information modeling is developed. The developed BIM shared ontology (BIMSO) can be used as a foundation for creating other AEC-FM domain ontologies. For example, BIMSO is used as a base to develop a BIM design ontology (BIMDO) for buildings which represents BIM element design properties. The following sections describe the methodology used for ontology development and the structures of BIMSO and BIMDO.

3.2 BIM Shared Ontology (BIMSO)

To develop an ontology, the ontology requirements should be defined. The followings are the requirements for BIMSO:

- **Purpose:** The purpose of BIMSO is providing a conceptual knowledge model for building information modeling that can be used by different building domains for developing domain ontologies. BIMSO is a foundation ontology that can be extended to create various AEC-FM domain ontologies.
- **Scope:** BIMSO is limited to sharing and exchanging building information among various AEC-FM domains. The scope provides answers to competency questions related to building elements, levels, spaces, and construction phases.
- **Implementation language:** The ontology is implemented in RDF/OWL.
- **Intended end-users:** Various AEC-FM domains are considered as the end users.
- **Intended use:** The intended use is to provide a semantic bridge for integrating and exchanging AEC-FM domain information.

In building projects, different AEC-FM domains need to exchange information about an element or a group of elements in a building. Figure 3-2 shows the main concepts in BIM shared ontology (BIMSO) developed in this study. The BIMSO ontology can be used for creating knowledge bases about one or more building elements included in various phases, levels and spaces of a building project.

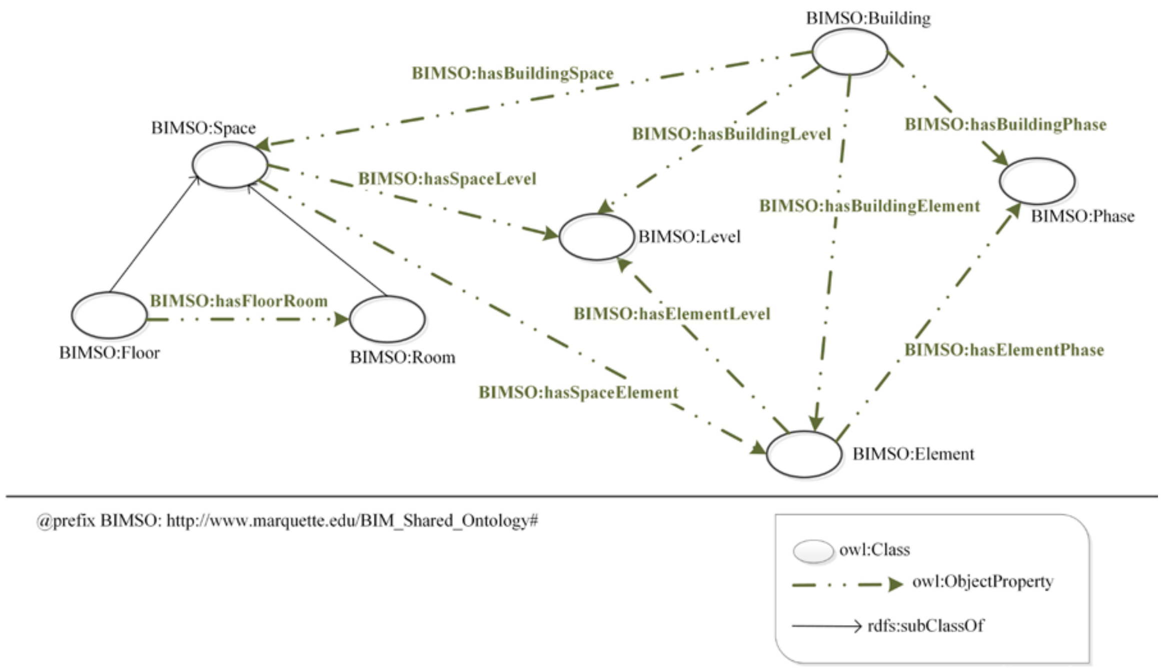


Figure 3-2: BIMSO general view

Every concept and property must be uniquely identified in the Semantic Web. A Uniform Resource Identifier (URI) (W3C 2005) is used for this purpose. To uniquely identify concepts in BIMSO, the URI http://www.marquette.edu/BIM_Shared_Ontology# is defined with the prefix BIMSO. For example, the concept Element is shown in Figure 3-2 as BIMSO:Element which is equal to the URI as http://www.marquette.edu/BIM_Shared_Ontology#Element.

A building project includes a large number of elements such as footings, walls, windows, and doors. Most of a building project information is about its elements; each AEC-FM domain creates different types of information about building elements. For example, designers specify elements' material properties and geometry, while project schedulers provide elements' construction schedule, and suppliers provide element's material properties and cost information. If every domain extends BIMSO to develop their domain ontologies, it would allow integrating different domain information about building elements.

To organize building element types in BIMSO, UNIFORMAT II classification system (ASTM standard 2015) is used. UNIFORMAT II is an ASTM standard which has been revised by Construction Specifications Institute (CSI) and Construction Specifications Canada (CSC). UNIFORMAT II has 4 levels that subdivide building element types as follows:

Level 1: Major Group Element Types

Level 2: Group Element Types

Level 3: Sub-Group Element Types

Level 4: Individual Element Types

UNIFORMAT II level 1 has 7 major groups: A-Substructure, B-Shell, C-Interiors, D-Services, E-Equipment & Furnishes, F-Special Construction, and G-Building Site Work. Figure 3-3 shows the top level concepts in BIMSO:Element organized according to the UNIFORMAT II classification Level 1. A screenshot of BIMSO in protégé software is also shown in Figure 3-3.

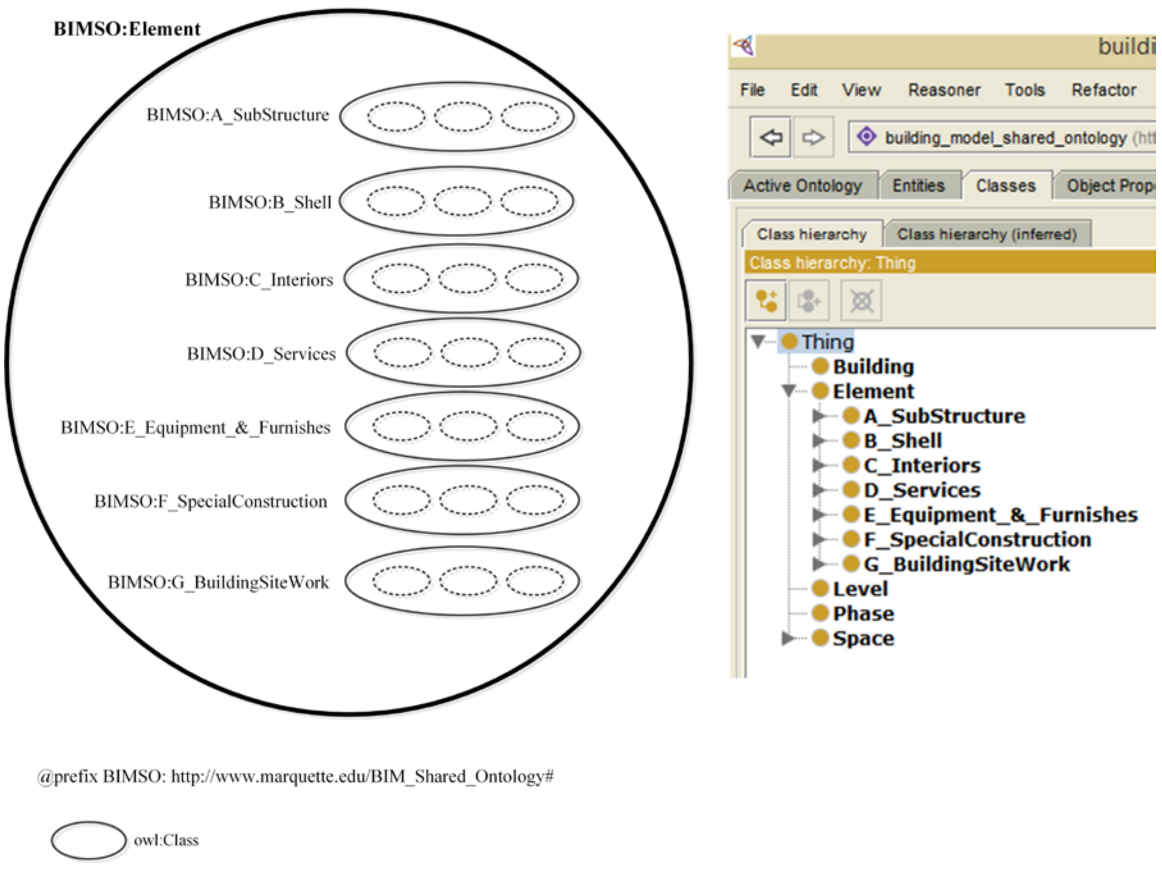
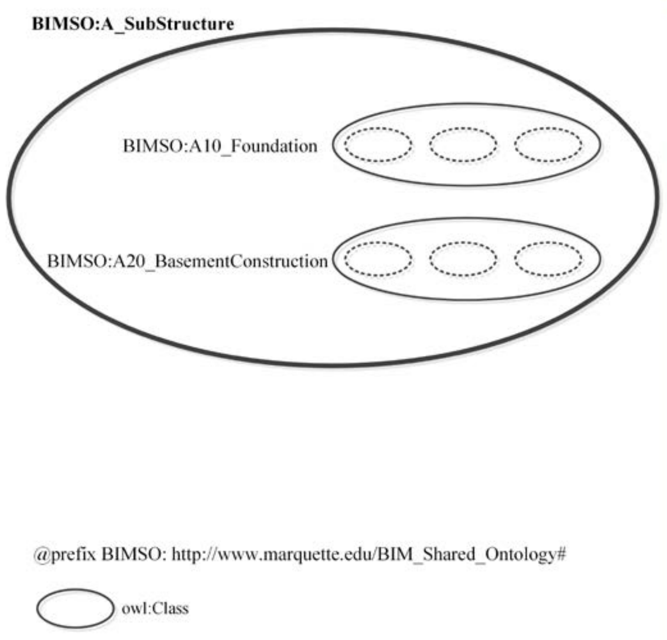
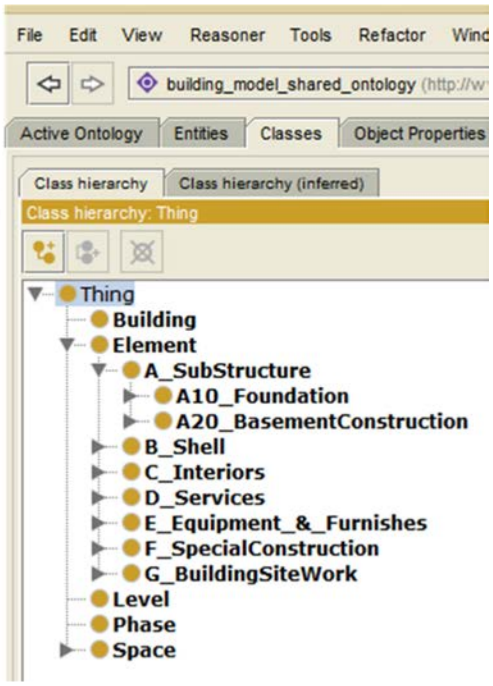


Figure 3-3: Top level concepts in BIMSO:Element

UNIFORMAT II level 2 divides each major group in level 1 into a number of element type groups. For example, major group A-Substructure is divided into A10-Foundation and A20-Basement Construction. Sub-classes for the top level concepts in BIMSO:Element are created corresponding to level 2 UNIFORMAT II classification system. Figure 3-4 shows element type groups defined for major group A-Substructure. In a similar method, other level 1 major groups are divided into sub-classes.


BIMSO:A_SubStructure
BIMSO:A10_Foundation
BIMSO:A20_BasementConstruction
@prefix BIMSO: http://www.marquette.edu/BIM_Shared_Ontology#
owl:Class



Thing
- Building
- Element
  - A_SubStructure
    - A10_Foundation
    - A20_BasementConstruction
  - B_Shell
  - C_Interiors
  - D_Services
  - E_Equipment_&_Furnishes
  - F_SpecialConstruction
  - G_BuildingSiteWork
- Level
- Phase
- Space

Figure 3-4: Sub-classes of top level concept A-Substructure in BIMSO

In level 3 of the UNIFORMAT II, every element type group of level 2 is divided into sub-groups. For example, A10-Foundation is divided into A1010-Standard Foundation, A1020-Special Foundations, and A1030-Slab on Grade. Subclasses of element type groups in BIMSO were created according to the UNIFORMAT II level 3. Figure 3-5 shows sub-groups for A10-Foundation. In a similar method, other sub-groups for the UNIFORMAT II level 3 element type groups are created.

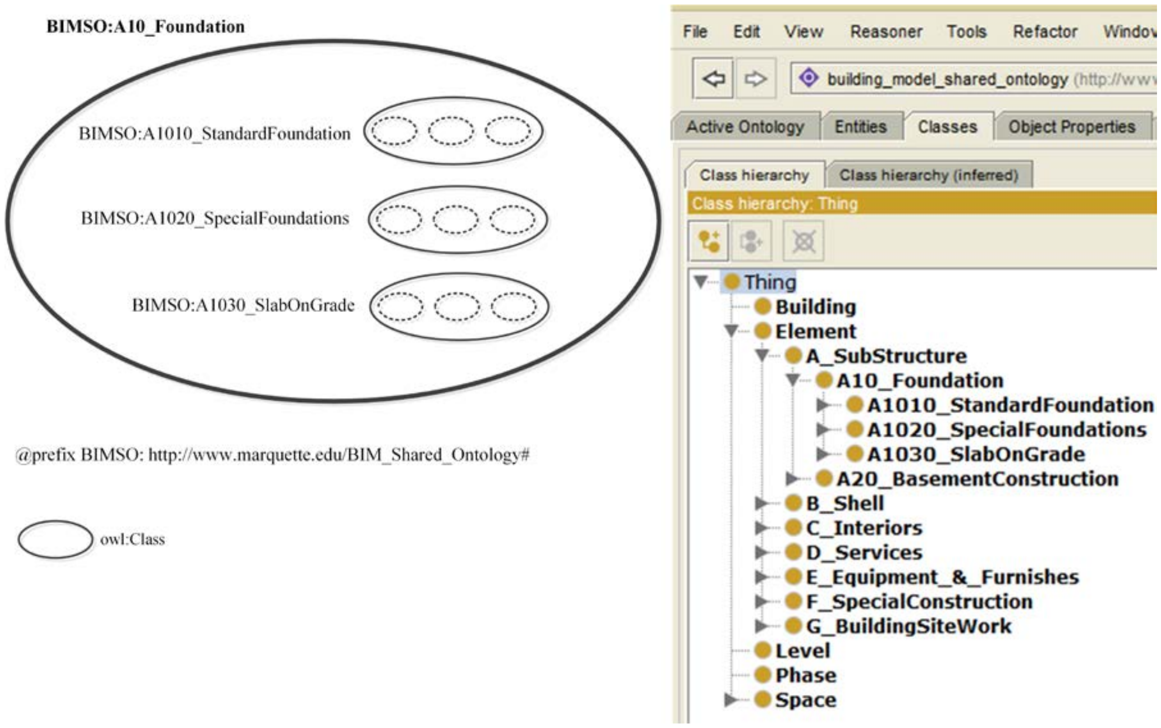
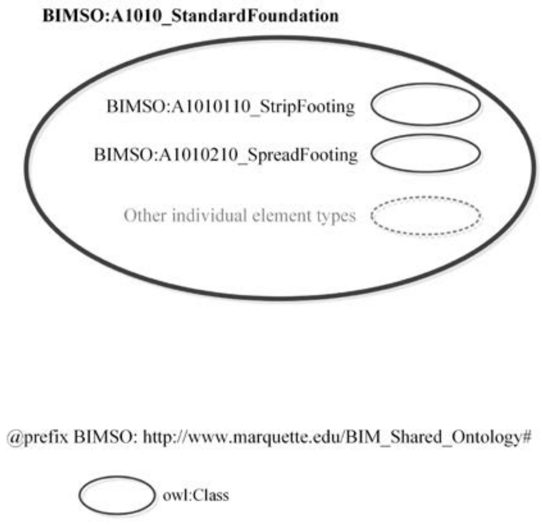
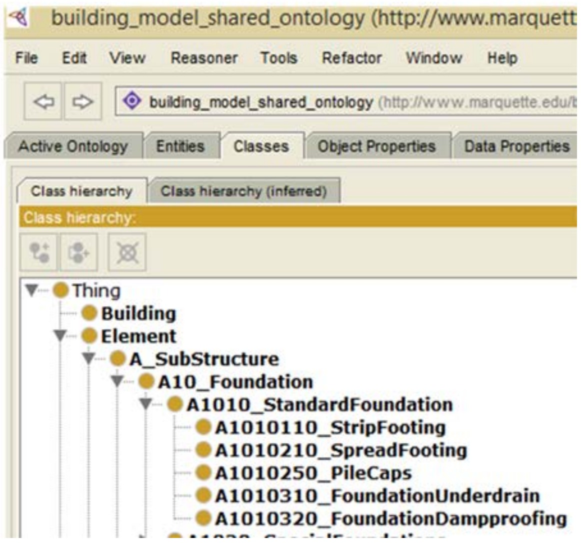


Figure 3-5: Sub-groups of A10-Foundation in BIMSO

In the level 4 of UNIFORMAT II, every sub-group is divided into Individual element types. For example, A1010-Standard Foundation is divided into types such as A1010110-Strip Footing and A1010210-Spread Footing. Figure 3-6 shows individual element types of A1010-Standard Foundation in BIMSO. In a similar method, other individual element types are created.


BIMSO:A1010_StandardFoundation
BIMSO:A1010110_StripFooting
BIMSO:A1010210_SpreadFooting
Other individual element types
@prefix BIMSO: http://www.marquette.edu/BIM_Shared_Ontology#
owl:Class



```
Thing
  Building
  Element
    A_SubStructure
      A10_Foundation
        A1010_StandardFoundation
          A1010110_StripFooting
          A1010210_SpreadFooting
          A1010250_PileCaps
          A1010310_FoundationUnderdrain
          A1010320_FoundationDampproofing
```

Figure 3-6: Individual element types of A1010-Standard Foundation in BIMSO

Figure 3-7 shows a general view of different levels of BIMSO:Element as presented in Figures 3-3 to 3-6. In Figure 3-7, different levels of UNIFORMAT II are shown on the left side. Footing-1 is an individual element (or an instance) of type BIMSO:A1010210. In a similar manner, other individual elements are created.

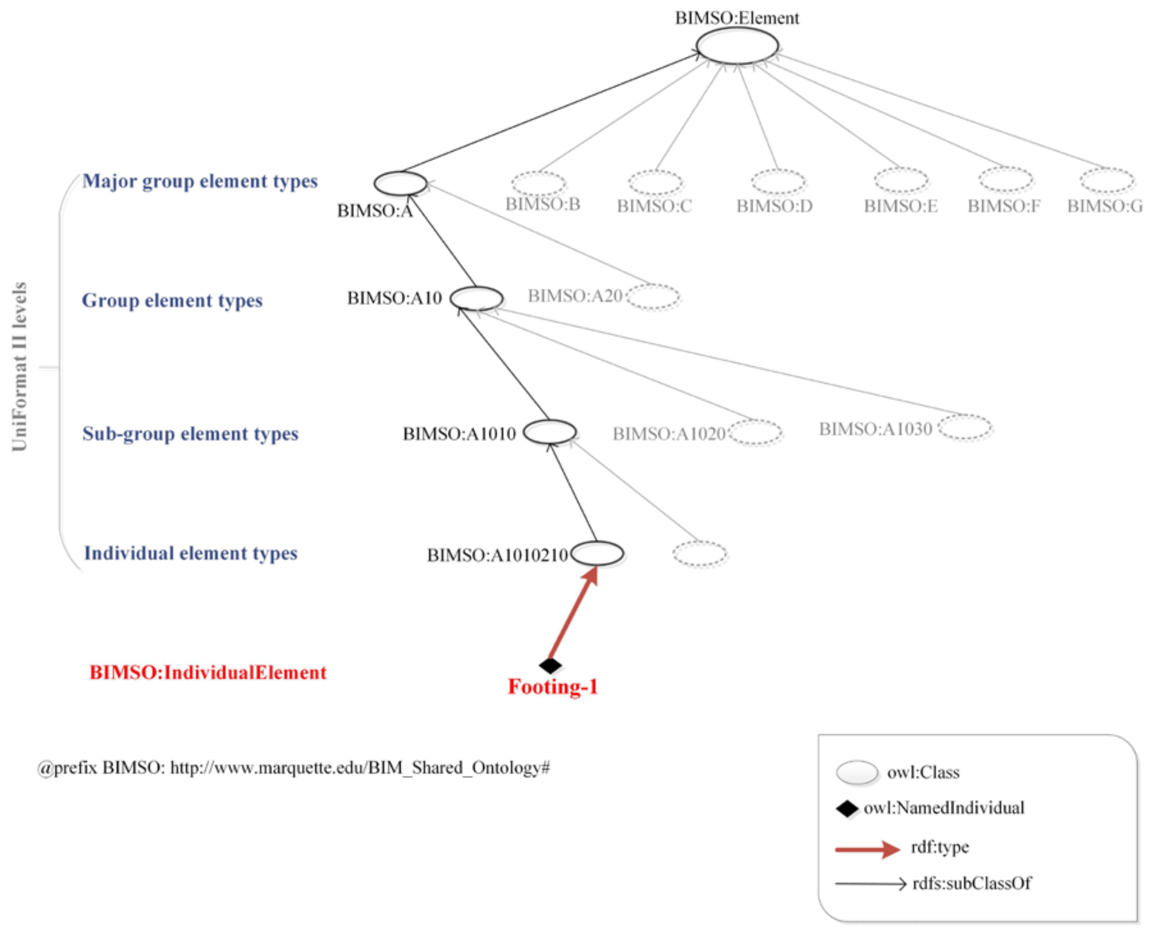


Figure 3-7: General view of different levels of element types in BIMSO

Every AEC-FM domain can define new element domain properties and relationships by extending BIMSO:Element. The following section explains how BIM design ontology is built by extending BIMSO ontology.

3.3 BIM Design Ontology (BIMDO)

The ontology requirements for BIMDO are defined as follow:

- **Purpose:** The purpose of BIMDO is to provide a conceptual model for expressing the design properties of BIM elements.

- **Scope:** The scope is limited to answer competency questions related to element identities, sizes, and material properties. BIMDO also responds to competency questions about element relationships such as hosts and intersects.
- **Implementation language:** The ontology is implemented in RDF/OWL.
- **Intended end-users:** Intended end-users are various AEC-FM domains.
- **Intended use:** Intended use is to create a BIM design knowledge base.

The prefix BIMDO is used for the BIM design ontology URI: http://www.marquette.edu/BIM_Design_Ontology#. The BIM design ontology developed in this study is shown in Figure 3-8. BIMSO:IndividualElement in Figure 3-8 represents all individual elements in a building project. Every individual element belongs to an element type in BIMSO (see Figure 3-7). BIM design ontology adds design properties to BIMSO:IndividualElement as shown in Figure 3-8.

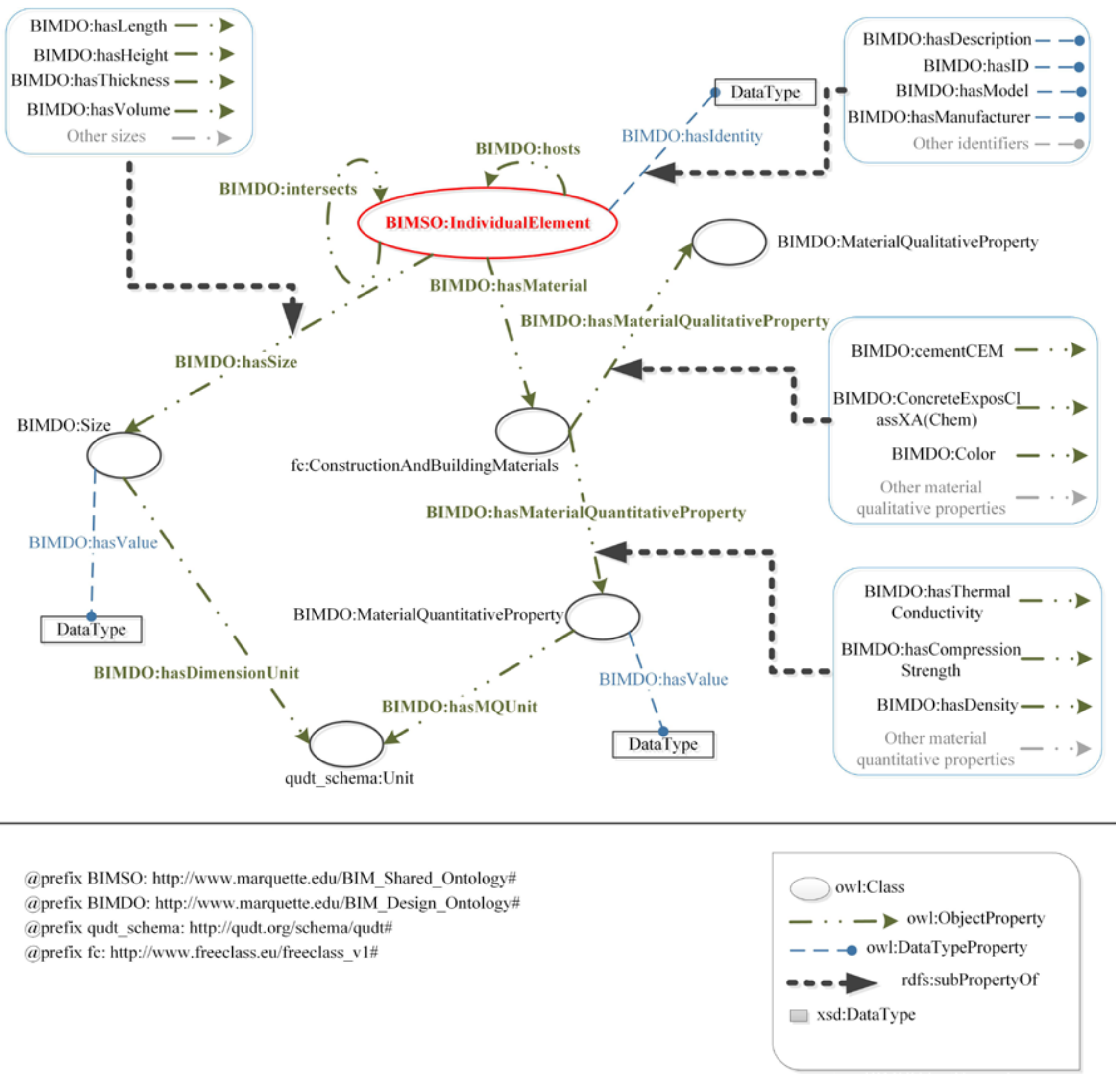
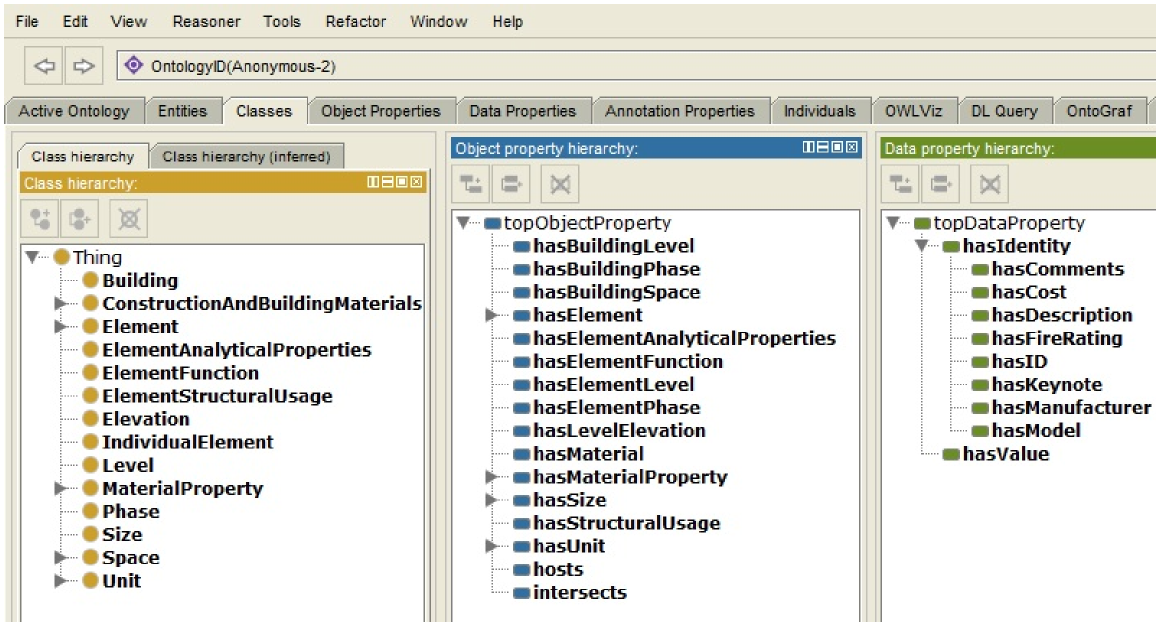

BIMDO:hasLength
BIMDO:hasHeight
BIMDO:hasThickness
BIMDO:hasVolume
Other sizes
DataType
BIMDO:hasIdentity
BIMDO:hasDescription
BIMDO:hasID
BIMDO:hasModel
BIMDO:hasManufacturer
Other identifiers
BIMDO:hosts
BIMDO:intersects
BIMSO:IndividualElement
BIMDO:MaterialQualitativeProperty
BIMDO:hasMaterial
BIMDO:hasMaterialQualitativeProperty
BIMDO:hasSize
BIMDO:cementCEM
BIMDO:ConcreteExposClassXA(Chem)
BIMDO:Color
Other material qualitative properties
BIMDO:Size
fc:ConstructionAndBuildingMaterials
BIMDO:hasMaterialQuantitativeProperty
BIMDO:hasValue
DataType
BIMDO:MaterialQuantitativeProperty
BIMDO:hasThermal Conductivity
BIMDO:hasCompression Strength
BIMDO:hasDensity
Other material quantitative properties
BIMDO:hasDimensionUnit
BIMDO:hasMQUnit
BIMDO:hasValue
DataType
qudt_schema:Unit
@prefix BIMSO: http://www.marquette.edu/BIM_Shared_Ontology#
@prefix BIMDO: http://www.marquette.edu/BIM_Design_Ontology#
@prefix qudt_schema: http://qudt.org/schema/qudt#
@prefix fc: http://www.freeclass.eu/freeclass_v1#
owl:Class
owl:ObjectProperty
owl:DataTypeProperty
rdfs:subPropertyOf
xsd:DataType

Figure 3-8: BIM design ontology (BIMDO)

In Figure 3-8, element identities are defined using the data type property BIMDO:hasIdentity and its sub-properties (e.g. BIMDO:hasDescription, BIMDO:hasID, and BIMDO:hasModel). Intersect and host relationships between elements are modeled using object properties BIMDO:intersects and BIMDO:hosts. The object property BIMDO:hasSize has sub-properties that define element sizes such as length, height, thickness, and volume. A unit of measurement and a value are defined for each size. The

QUDT (Hodgson et al. 2011) ontology is used to represent units of measurement. An element has one or more materials. The Free Class OWL ontology, FC, is used to classify building and construction materials. FC is developed by the European building and construction materials database and has over 88 million triples of real business data to describe construction materials (BauDataWeb 2015). Every material has a number of qualitative (e.g., cement type) and quantitative (e.g., compression strength) properties.

The implementation of BIMDO in Protégé is shown in Figure 3-9. The left panel in Figure 3-9 shows the concepts (classes), the middle panel shows the object properties, and the right panel shows the data type properties defined in BIMDO.



```
▼ Thing
    Building
  ▶ ConstructionAndBuildingMaterials
  ▶ Element
    ElementAnalyticalProperties
    ElementFunction
    ElementStructuralUsage
    Elevation
    IndividualElement
    Level
  ▶ MaterialProperty
    Phase
    Size
  ▶ Space
  ▶ Unit
```

```
▼ topObjectProperty
    hasBuildingLevel
    hasBuildingPhase
    hasBuildingSpace
  ▶ hasElement
    hasElementAnalyticalProperties
    hasElementFunction
    hasElementLevel
    hasElementPhase
    hasLevelElevation
    hasMaterial
  ▶ hasMaterialProperty
  ▶ hasSize
    hasStructuralUsage
  ▶ hasUnit
    hosts
    intersects
```

```
▼ topDataProperty
  ▼ hasIdentity
      hasComments
      hasCost
      hasDescription
      hasFireRating
      hasID
      hasKeynote
      hasManufacturer
      hasModel
    hasValue
```

Figure 3-9: BIMDO implemented in Protégé

3.4 BIM Knowledge Base

A knowledge base is an information repository created based on ontologies for collecting, organizing and sharing domain information (Noy and McGuinness 2001). So, a

BIM knowledge base is an information repository that is created based on BIM ontologies for managing information about a specific building project. Figure 3-10 shows the methodology used in this study for creating a BIM knowledge.

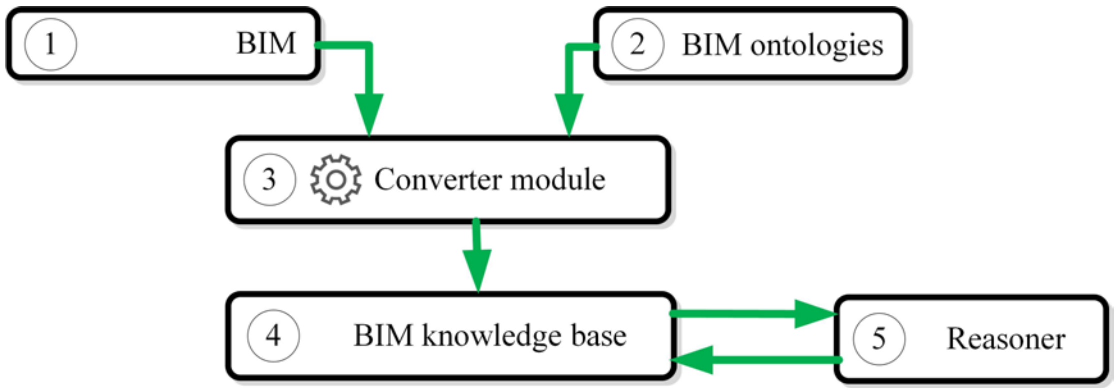


Figure 3-10: BIM knowledge base creation process

A brief description for different components shown in Figure 3-10 is provided below:

1. Building information model (BIM) is created by designers using a BIM platform (e.g. Autodesk Revit).
2. BIM ontologies provide the schema for converting BIM to RDF/OWL format before it can be saved in a BIM knowledge base. Several studies have investigated approaches based on IFC ontology for this purpose (Karan et al. 2015; Karan and Irizarry 2015; Pauwels and Terkaj 2014; De Farias et al. 2014; Pauwels et al. 2011b; Pauwels et al. 2011a; Demir et al. 2010; Beetz et al. 2009). In this study, the BIMSO and BIMDO ontologies are used for organizing BIM data in the RDF/OWL format.
3. A converter module is used to convert BIM data to the RDF/OWL format. Studies that use the IFC ontology utilize EXPRESS-TO-OWL approaches (Karan et al. 2015; Karan

and Irizarry 2015; Pauwels and Terkaj 2014; De Farias et al. 2014; Pauwels et al. 2011b; Pauwels et al. 2011a; Demir et al. 2010; Beetz et al. 2009). The converter module developed in this study extracts Revit model data using Revit application programming interface (API). Revit API allows access to the graphical and parameter data of a building information model.

4. A BIM knowledge base is a semantic representation of a building information model created according to a set of ontologies. As the knowledge about a BIM is extracted, it is stored in a repository. OpenRDF Sesame triplestore (Sesame 2015) is used for saving BIM knowledge bases in this study. A Sesame triplestore provides a SPARQL endpoint interface that allows local and remote (over the Internet) access to its data.
5. A Reasoner software adds logical inference capabilities to a knowledge base. The Apache Jena library (Apache Jena 2015) and the Pellet Reasoner (Pellet 2014) are used for this purpose.

The following section presents the organization of a knowledge base for an example project.

3.5 A Case Study

The building used in this case study is shown in Figure 3-11. The building is called Engineering Hall and is modeled in the Autodesk Revit BIM platform (Autodesk 2015b).

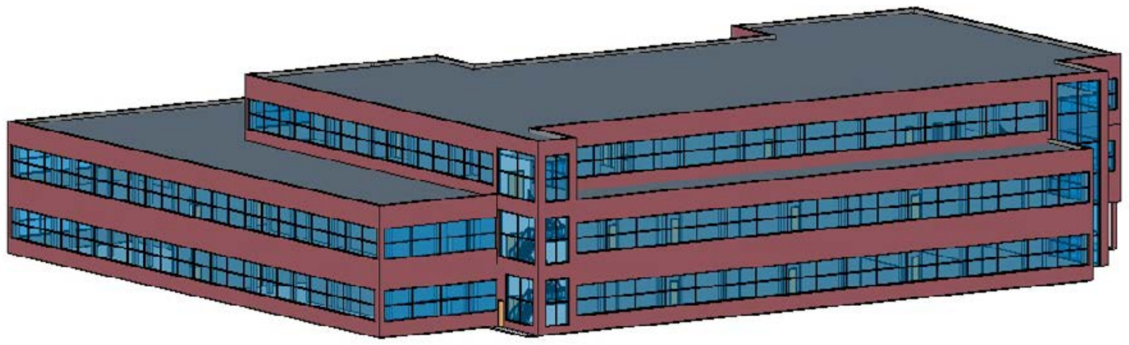
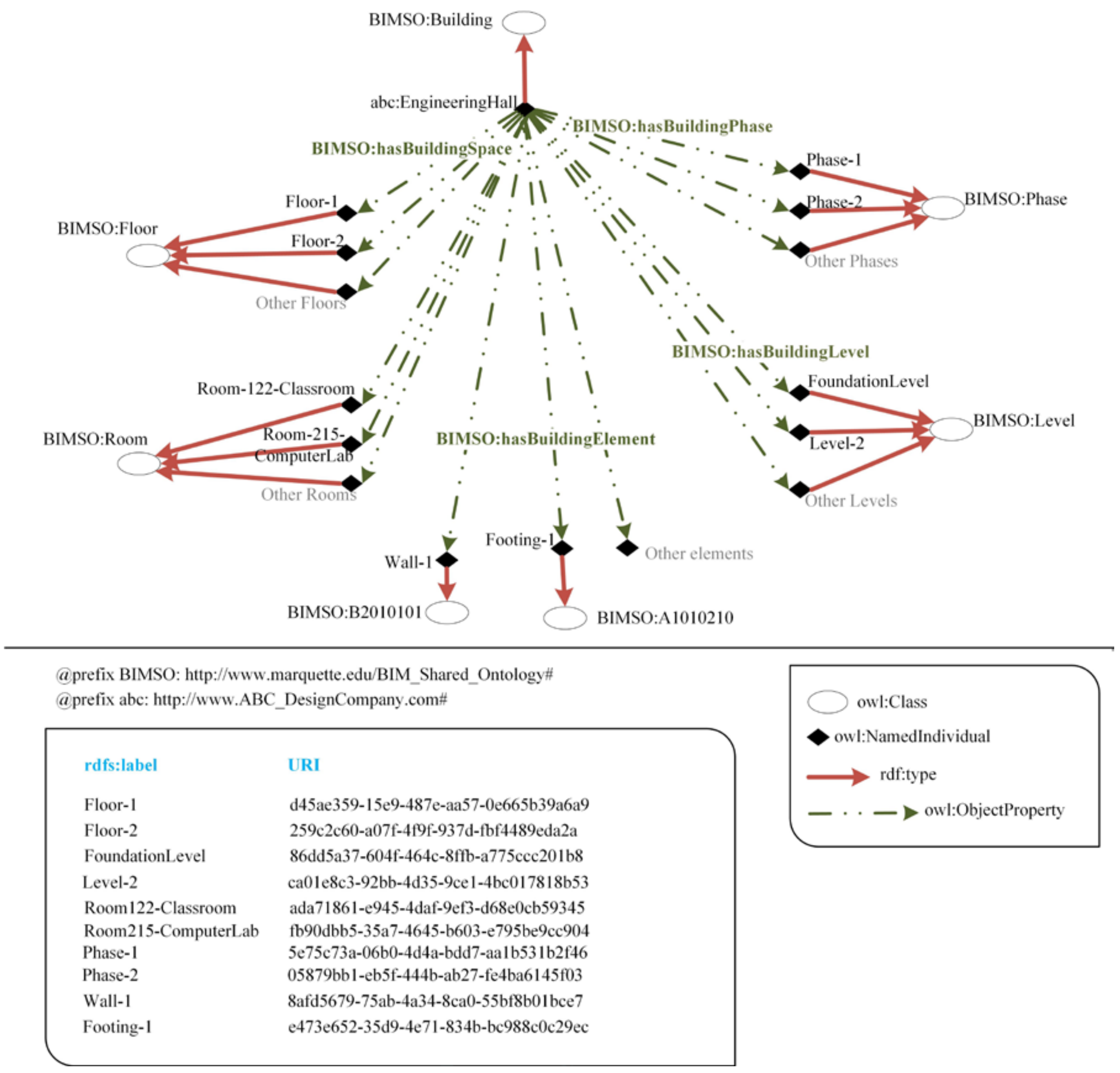


Figure 3-11: A 3D view of Engineering Hall

The first step for creating a BIM knowledge base for a building is to assign unique identifiers to the building and its elements. URIs (W3C 2005) are used as unique identifiers for Engineering Hall and its elements. Since every building has a designer, the building model URI is created by adding the name of the building to the URI of its designer. The URI http://www.ABC_DesignCompany.com# is used for the designer of Engineering Hall and the prefix abc is assigned to it. So, the URI of the Engineering Hall is abc:EngineeringHall which is equal to URI http://www.ABC_DesignCompany.com#EngineeringHall.

For BIM element URI, 128-bit global unique identifiers created by the BIM platform are used. For example, the 128-bit URI that Revit platform generated for a footing in Engineering Hall is e473e652-35d9-4e71-834b-bc988c0c29ec. In this study, to simplify references to the footing element, the label Footing-1 is assigned to the 128-bit unique identifier of the footing.

Figure 3-12 shows a schematic view of the Engineering Hall knowledge base that shows how instances of phases, floors, levels, rooms, and elements of Engineering Hall are defined using the BIMSO ontology. A list of the URIs and their corresponding labels is shown at the bottom of Figure 3-12. For example, e473e652-35d9-4e71-834b-bc988c0c29ec is the URI of a footing labeled as Footing-1. Every instance in Figure 3-12 has a type in BIMSO. For example, Footing-1 is of type BIMSO:A1010210 which is a spread footing (see Figure 3-6).

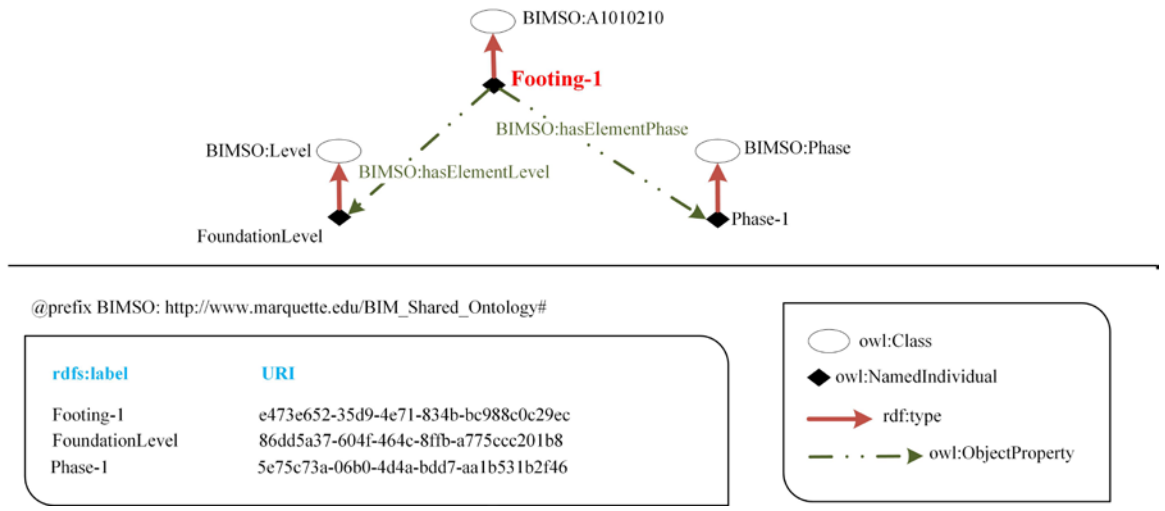


@prefix BIMSO: http://www.marquette.edu/BIM_Shared_Ontology#
@prefix abc: http://www.ABC_DesignCompany.com#

rdfs:label	URI
Floor-1	d45ae359-15e9-487e-aa57-0e665b39a6a9
Floor-2	259c2c60-a07f-4f9f-937d-fbf4489eda2a
FoundationLevel	86dd5a37-604f-464c-8ffb-a775ccc201b8
Level-2	ca01e8c3-92bb-4d35-9ce1-4bc017818b53
Room122-Classroom	ada71861-e945-4daf-9ef3-d68e0cb59345
Room215-ComputerLab	fb90dbb5-35a7-4645-b603-e795be9cc904
Phase-1	5e75c73a-06b0-4d4a-bdd7-aa1b531b2f46
Phase-2	05879bb1-eb5f-444b-ab27-fe4ba6145f03
Wall-1	8afd5679-75ab-4a34-8ca0-55bf8b01bce7
Footing-1	e473e652-35d9-4e71-834b-bc988c0c29ec

Figure 3-12: A schematic view of the Engineering Hall knowledge base

The rest of this section presents relations and properties of Footing-1. Figure 3-13 shows how Footing-1 is related to the construction phases and building levels defined for Engineering Hall. Footing-1 belongs to Phase-1 of the Engineering Hall project and located at foundation level. In a similar method, the Engineering Hall BIM knowledge base represents all element relations to the building phases and levels. For other elements that may belong to a room and a floor such as a wall element, relations of the element to floor and room are defined in a similar method. Examples of a wall element that include element floor and room and the host and intersect relations between elements are shown in Appendix A.

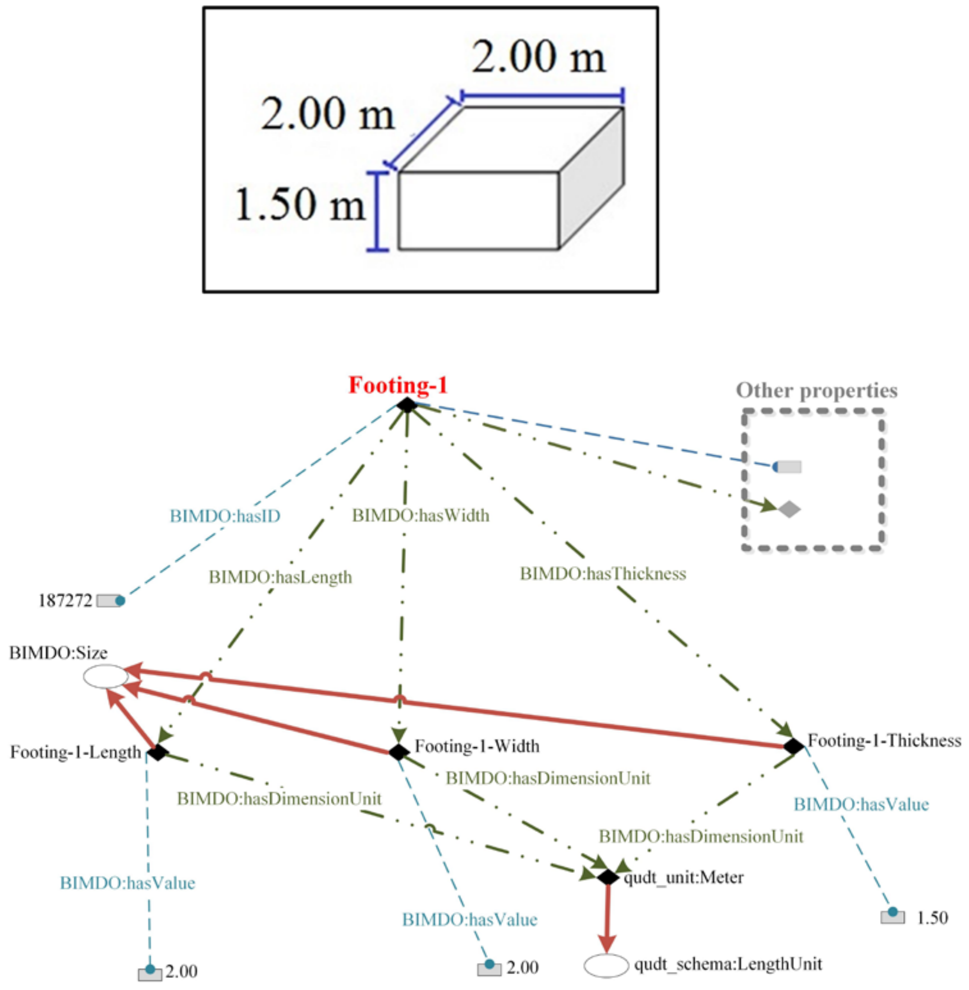


@prefix BIMSO: http://www.marquette.edu/BIM_Shared_Ontology#

rdfs:label	URI
Footing-1	e473e652-35d9-4e71-834b-bc988c0c29ec
FoundationLevel	86dd5a37-604f-464c-8ffb-a775ccc201b8
Phase-1	5e75c73a-06b0-4d4a-bdd7-aa1b531b2f46

Figure 3-13: Footing-1 relations with building phases and levels

In addition to the above-mentioned relations, the BIM knowledge base also includes building element design properties. Element design properties include element material and element sizes. The design properties for the Engineering Hall elements are defined using the BIMDO ontology. Figure 3-14 shows the length, width, and thickness properties of Footing-1. As Figure 3-14 shows, these properties are multi-valued and need a numerical value and a unit of measurement. For example, the length of Footing-1 has value 2.00 and unit Meter. The QUDT (Hodgson et al. 2011) ontology is used for representing units of measurement.

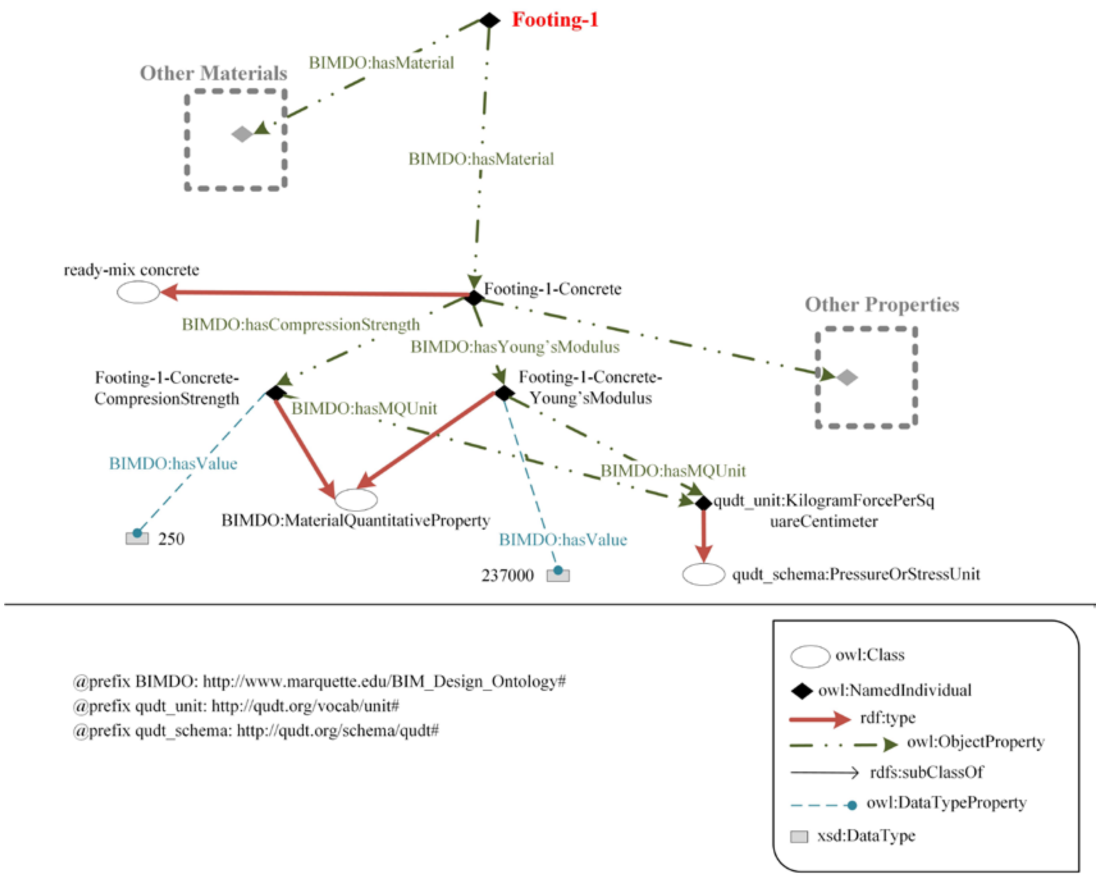


@prefix BIMDO: http://www.marquette.edu/BIM_Design_Ontology#
@prefix qudt_unit: http://qudt.org/vocab/unit#
@prefix qudt_schema: http://qudt.org/schema/qudt#

rdfs:label	URI
Footing-1	e473e652-35d9-4e71-834b-bc988c0c29ec
Footing-1-Length	e473e652-35d9-4e71-834b-bc988c0c29ec_Length
Footing-1-Width	e473e652-35d9-4e71-834b-bc988c0c29ec_Width
Footing-1-Thickness	e473e652-35d9-4e71-834b-bc988c0c29ec_Thickness

Figure 3-14: Footing-1 dimensions

The element material representation in a BIM knowledge base specifies the material type and properties. The BIMDO ontology is used for representing element material. Figure 3-15 shows the representation of concrete material used in Footing-1. As Figure 3-15 shows, the compression strength of the concrete used in Footing-1 is 250 and its unit is Kilogram Force per Square Centimeter. Other material properties are included in the knowledge base in a similar manner. As discussed above, the material types are modeled using FC ontology (BauDataWeb 2015). Appendix B shows RDF/XML representation of the footing example.



@prefix BIMDO: http://www.marquette.edu/BIM_Design_Ontology#
@prefix qudt_unit: http://qudt.org/vocab/unit#
@prefix qudt_schema: http://qudt.org/schema/qudt#

rdfs:label	URI
Footing-1	e473e652-35d9-4e71-834b-bc988c0c29ec
Footing-1-Concrete	e473e652-35d9-4e71-834b-bc988c0c29ec_Concrete
Footing-1-Concrete-CompresionStrength	e473e652-35d9-4e71-834b-bc988c0c29ec_Concrete_CompresionStrength
Footing-1-Concrete-Young'sModulus	e473e652-35d9-4e71-834b-bc988c0c29ec_Concrete_Young'sModulus
ready-mix concrete	http://www.freeclass.eu/freeclass_v1#C_A140-gen

Figure 3-15: Footing-1 material properties

CHAPTER 4
ASSEMBLY AND WORK ITEM KNOWLEDGE BASES FOR COST ESTIMATING

4.1 Assembly and Work Item Ontology

An estimating assembly represents the work items that must be completed for construction of a building element. For example, an estimating assembly for a footing element includes forming, reinforcing, and placing concrete work items. Figure 4-1 shows the estimating assembly ontology developed in this study. To uniquely identify concepts in the estimating assembly ontology, the URI http://www.marquette.edu/estimating_ontology# is defined with the prefix mueo. The purpose of the estimating ontology is to provide a semantic model for estimating assemblies. The assembly ontology includes assembly properties such as ID, title, special conditions (e.g., weather condition), assembly cost and a list of work items that are part of the assembly. The ontology allows adding a list of job conditions that can influence work item productivities. Currently, no ontological resources are available for estimating assemblies. In this study, the estimating assembly ontology is organized according to the UNIFORMAT II classification system; therefore, the ID and title of an estimating assembly are the same as those of its UNIFORMAT II counterpart. Figure 4-1 also shows a screenshot of the estimating assembly ontology implementation in Protégé software (Stanford University 2015).

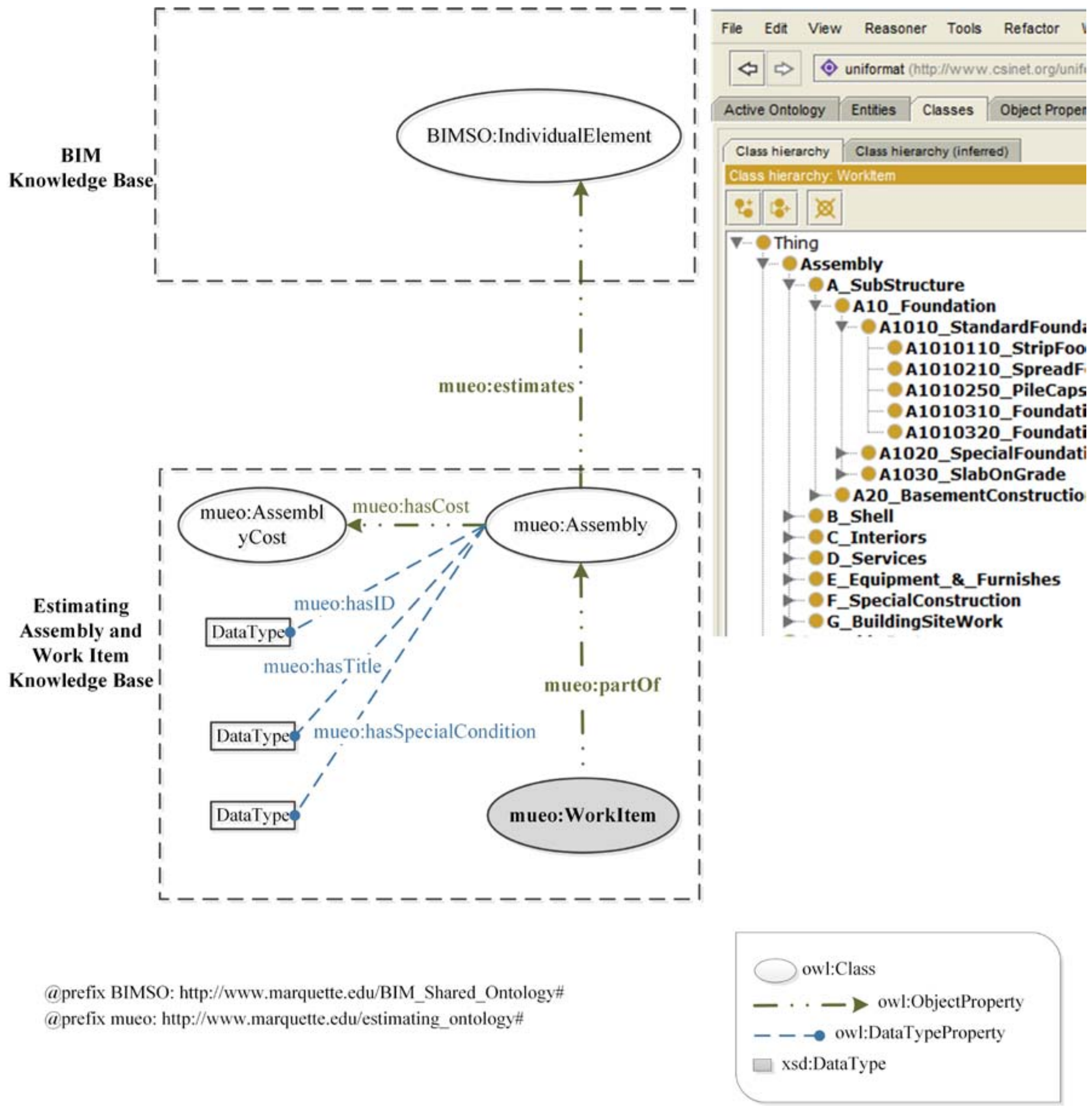


Figure 4-1: Estimating assembly ontology

The purpose of the work item ontology is providing a semantic model for construction work items. The work item ontology developed in this study is shown in Figure 4-2. The concepts in the work item ontology are extracted from non-ontological resources including estimating books and estimating references such as CSI MasterFormat (CSI 2015) and RSMeans reference books (RSMeans 2015).

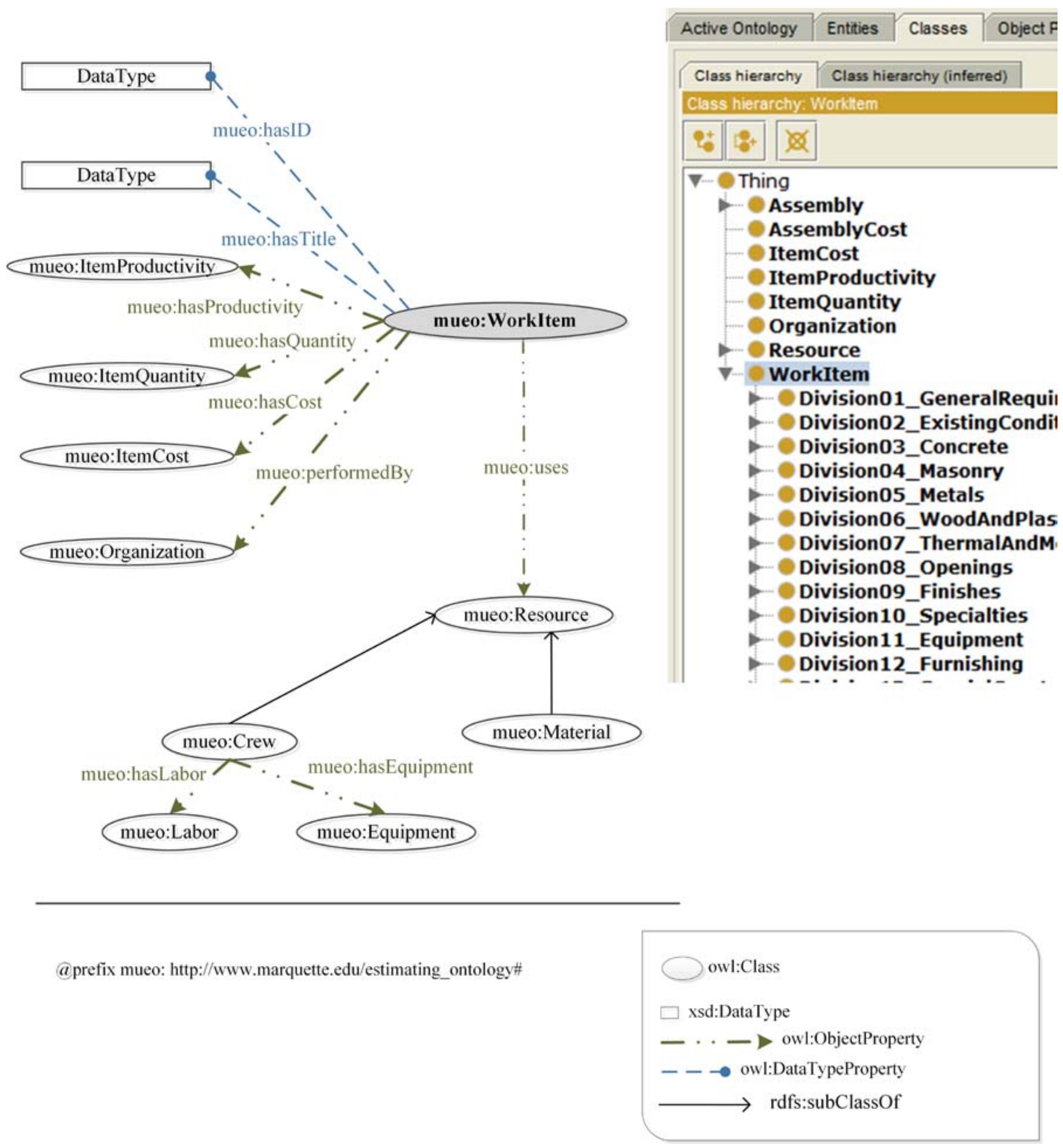


Figure 4-2: Work item ontology

The work item ontology shown in Figure 4-2 includes the following concepts:

A) **ID & Title:** Each work item has an ID and a title for identification. CSI MasterFormat (CSI 2015) work item IDs and titles are used to identify work items.

B) **ItemProductivity**: This concept specifies the crew productivity for the work item. The productivity is affected by labor, equipment and job conditions.

C) **ItemQuantity**: This concept represents the quantity of work involved in a work item. The work item's quantity is calculated by obtaining information from the BIM knowledge base.

D) **Resource**: This concept defines the type, unit cost, and quantity of resources used in a work item. It is divided to the following subclasses:

1. **Crew**: This concept represents the type, unit cost, and quantity of labor or equipment used in a work item. The quantities for labor and equipment are calculated based on the crew productivity and work item quantity.
2. **Materia**l: This concept defines the type, unit cost and quantity of the material used in a work item. The quantity of material for a work item is calculated directly from the work item quantity considering an appropriate waste factor. The Free Class OWL ontology (FC) (BauDataWeb 2015) (Ontology URI: http://www.freeclass.eu/freeclass_v1#) is used for specifying the type of material. FC is developed by the European Building and Construction Materials Database for describing construction materials.

E) **ItemCost**. This concept defines the estimated cost of a work item.

F) **Organization**: This concept defines the companies involved in a work item such as the contractor performing the work, material suppliers, and the inspecting organization. The organization ontology (Ontology URI: http://www.w3.org/ns/org#) developed by W3C is used in the work item ontology (W3C 2014).

The above mentioned ontologies are coded in Protégé software in this study. The Protégé user interface allows a user to open ontologies available on the web and reuse them when developing a new ontology. The Organization and Free Class OWL Ontologies are examples of existing ontologies that are reused in developing the assembly and work item ontologies.

4.2 Assembly and Work Item Knowledge Base

The above-mentioned ontologies are used to develop RDF and OWL knowledge bases for estimating assemblies and work items. Figure 4-3 shows a section of this knowledge base representing an estimating assembly instance for a spread footing element. In Figure 4-3, mueo represents the estimating assembly and work item ontologies shown in Figure 4-1 and 4-2. The prefix xyz is used to identify the example company that prepared the estimating assemblies and work items.

In Figure 4-3, xyz:A1010210_SpreadFooting_7261 is an instance of the mueo:A1010210 class in the estimating ontology which represents the spread footing class. The assembly instance has an ID and a title for identification. It also has a cold weather special condition which affects the productivity. The assembly shown in Figure 4-3 includes work items xyz:D03311370_PlacingConcrete_2457, xyz:D03111345_FormsInPlaceFooting_4152, and xyz:D03211160_ReinforcingInPlace_0151.

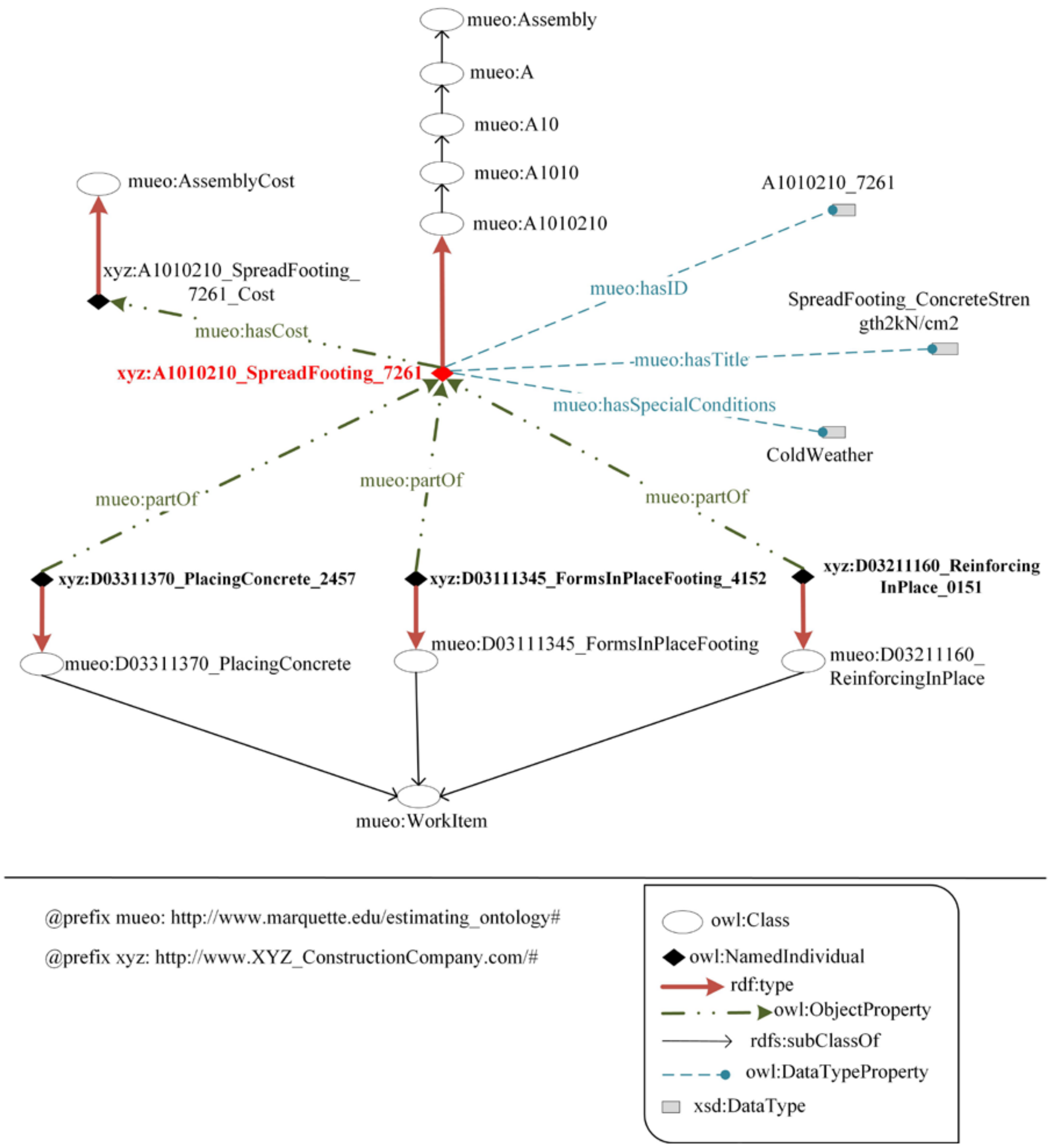

mueo:Assembly
mueo:A
mueo:A10
mueo:A1010
mueo:A1010210
mueo:AssemblyCost
xyz:A1010210_SpreadFooting_7261_Cost
mueo:hasCost
A1010210_7261
mueo:hasID
SpreadFooting_ConcreteStrength2kN/cm2
mueo:hasTitle
xyz:A1010210_SpreadFooting_7261
mueo:hasSpecialConditions
ColdWeather
mueo:partOf
mueo:partOf
mueo:partOf
xyz:D03311370_PlacingConcrete_2457
xyz:D03111345_FormsInPlaceFooting_4152
xyz:D03211160_ReinforcingInPlace_0151
mueo:D03311370_PlacingConcrete
mueo:D03111345_FormsInPlaceFooting
mueo:D03211160_ReinforcingInPlace
mueo:WorkItem
@prefix mueo: http://www.marquette.edu/estimating_ontology#
@prefix xyz: http://www.XYZ_ConstructionCompany.com/#
owl:Class
owl:NamedIndividual
rdf:type
owl:ObjectProperty
rdfs:subClassOf
owl:DataTypeProperty
xsd:DataType

Figure 4-3: A spread footing assembly in the knowledge base

The work items that are part of an estimating assembly are semantically defined according to the work item ontology shown in Figure 4-2. The semantic representation of xyz:D03311370_PlacingConcrete_2457 is shown in Figure 4-4. In the list of prefixes, rst is the prefix for URI of the material supplier for the work item, org is the organization ontology defined by W3C (W3C 2014), and fc represents the Free Class OWL ontology (FC) (BauDataWeb 2015). Multi-valued properties are used to model quantity, material, crew, productivity, and item cost. The rest of this chapter describes these properties.

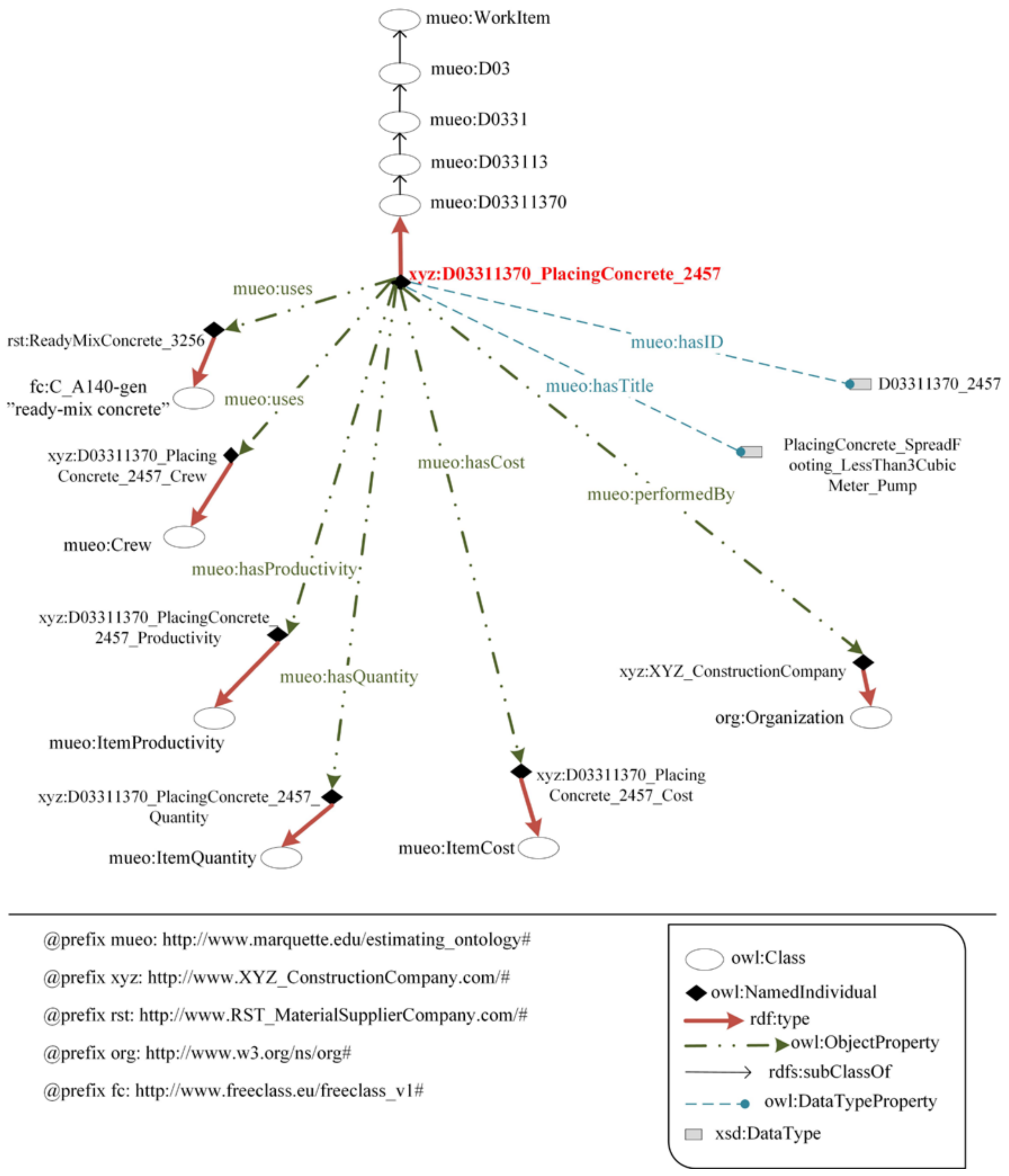

mueo:WorkItem
mueo:D03
mueo:D0331
mueo:D033113
mueo:D03311370
xyz:D03311370_PlacingConcrete_2457
mueo:uses
rst:ReadyMixConcrete_3256
fc:C_A140-gen "ready-mix concrete"
mueo:uses
xyz:D03311370_Placing Concrete_2457_Crew
mueo:Crew
mueo:hasProductivity
xyz:D03311370_PlacingConcrete_ 2457_Productivity
mueo:ItemProductivity
mueo:hasQuantity
xyz:D03311370_PlacingConcrete_2457_ Quantity
mueo:ItemQuantity
mueo:hasCost
xyz:D03311370_Placing Concrete_2457_Cost
mueo:ItemCost
mueo:hasID
D03311370_2457
mueo:hasTitle
PlacingConcrete_SpreadF ooting_LessThan3Cubic Meter_Pump
mueo:performedBy
xyz:XYZ_ConstructionCompany
org:Organization
@prefix mueo: http://www.marquette.edu/estimating_ontology#
@prefix xyz: http://www.XYZ_ConstructionCompany.com/#
@prefix rst: http://www.RST_MaterialSupplierCompany.com/#
@prefix org: http://www.w3.org/ns/org#
@prefix fc: http://www.freeclass.eu/freeclass_v1#
owl:Class
owl:NamedIndividual
rdf:type
owl:ObjectProperty
rdfs:subClassOf
owl:DataTypeProperty
xsd:DataType

Figure 4-4: A placing concrete work item

Figure 4-5 shows the semantic representation for the quantity of the placing concrete work item. A set of SWRL (Horrocks et al. 2004) rules are developed in this study to calculate work item quantities from the BIM element properties. SWRL is a language that expresses logic rules that are of the form of an implication between an antecedent (body) and consequent (head) which means that whenever the conditions specified in the antecedent hold, meaning, they are true, then the conditions specified in the consequent must also hold. This eliminates the need for manual mapping of BIM element properties to estimating assembly properties as is practiced in the current estimating applications.

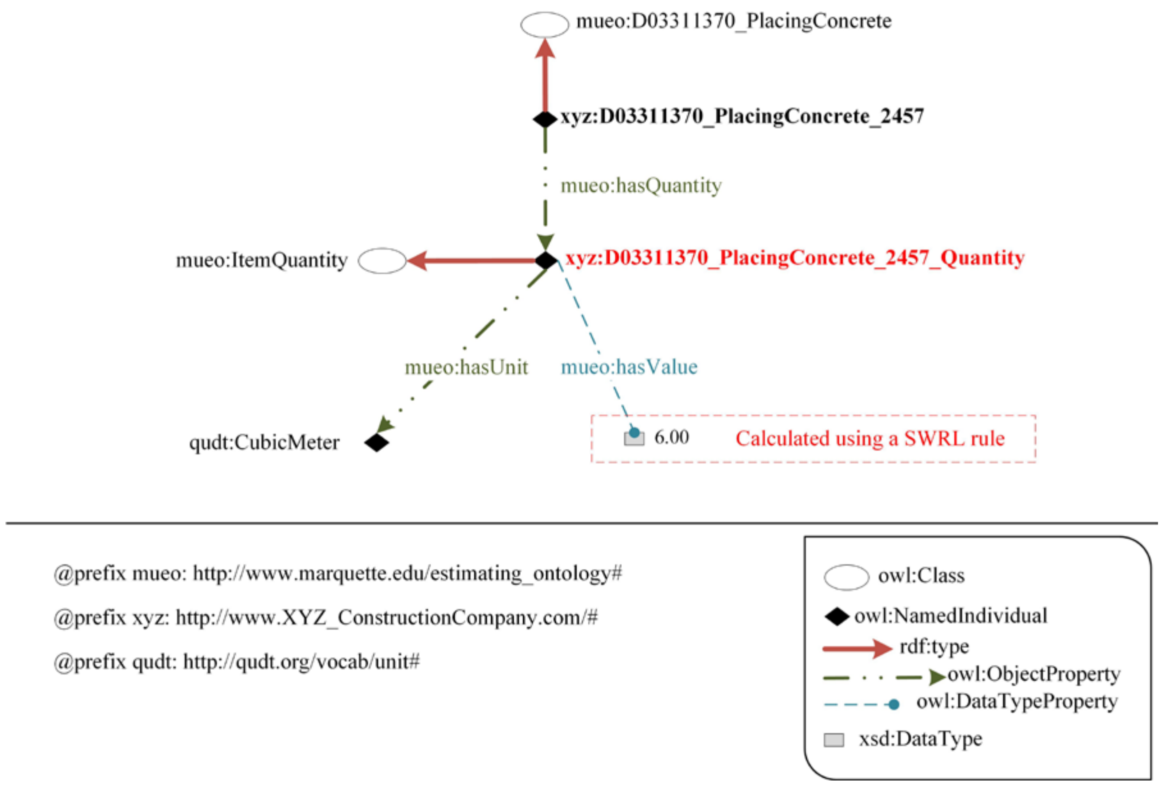

mueo:D03311370_PlacingConcrete
xyz:D03311370_PlacingConcrete_2457
mueo:hasQuantity
mueo:ItemQuantity
xyz:D03311370_PlacingConcrete_2457_Quantity
mueo:hasUnit
mueo:hasValue
qudt:CubicMeter
6.00
Calculated using a SWRL rule
@prefix mueo: http://www.marquette.edu/estimating_ontology#
@prefix xyz: http://www.XYZ_ConstructionCompany.com/#
@prefix qudt: http://qudt.org/vocab/unit#
owl:Class
owl:NamedIndividual
rdf:type
owl:ObjectProperty
owl:DataTypeProperty
xsd:DataType

Figure 4-5: Work item quantity representation

The SWRL rule for calculating the quantity of placing concrete work item is shown in Figure 4-6a; the graphical representation of the rule components is shown in Figure 4-6b. The following is a short description of the SWRL rule:

Lines 1 to 4:	Footing dimensions are accessed from the BIM knowledge base. In line 1, ?footing refers to an owl individual of type BIMSO:A1010210 which is the spread footing class. In lines 2 to 4, the values of the footing's length, width, and thickness are retrieved.
Lines 5 to 7:	A footing assembly (?footingAssembly) and its work item (?concreteWorkItem) are retrieved from the estimating knowledge base. In line 5, ?footingAssembly is assigned to the ?footing element that was defined in lines 1 to 4. In line 6, the work item type is specified as mueo:D03311370_PlacingConcrete. In line 7, the work item quantity that must be calculated is defined as ?concreteWorkItemQuantity.
Lines 8 and 9:	The mathematical operations to calculate the placing concrete work item quantity are defined. In line 8, the ?area is calculated by multiplying ?footingLengthValue and ?footingWidthValue. In line 9, ?concreteWorkItemQuantityValue is calculated by multiplying ?area and ?footingThicknessValue.
Line 10:	The calculated work item quantity value is assigned to ?concreteWorkItemQuantity.

Pellet Reasoner (Pellet 2014) is used to calculate work item quantities based on the SWRL rules developed. Pellet provides standard reasoning services for OWL knowledge bases.

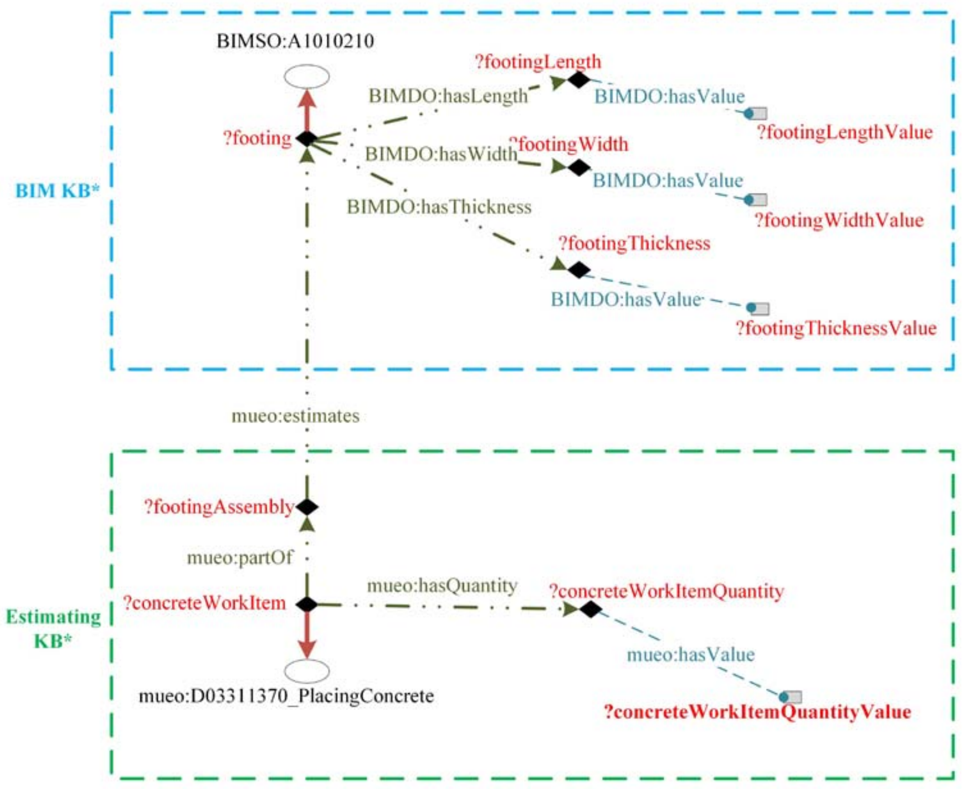
```
Line 1   BIMSO:A1010210(?footing),                                                                                      ⎫
Line 2   BIMDO:hasLength(?footing, ?footingLength), BIMDO:hasValue(?footingLength, ?footingLengthValue),                ⎬ BIM
Line 3   BIMDO:hasWidth(?footing, ?footingWidth), BIMDO:hasValue(?footingWidth, ?footingWidthValue),                    ⎪ KB*
Line 4   BIMDO:hasThickness(?footing, ?footingThickness),  BIMDO:hasValue(?footingThickness, ?footingThicknessValue),   ⎭

Line 5   mueo:estimates(?footingAssembly, ?footing),                                        ⎫
Line 6   mueo:D03311370_PlacingConcrete(?concreteWorkItem),                                 ⎬ Estimating
             mueo:partOf(?concreteWorkItem, ?footingAssembly),                              ⎪ KB*
Line 7   mueo:hasQuantity(?concreteWorkItem, ?concreteWorkItemQuantity),                    ⎭

Line 8   multiply(?area, ?footingLengthValue, ?footingWidthValue),                          ⎫ Calculations
Line 9   multiply(?concreteWorkItemQuantityValue, ?area, ?footingThicknessValue)            ⎭

Line 10  -> mueo:hasValue(?concreteWorkItemQuantity, ?concreteWorkItemQuantityValue)        } Rule result
```

(a) SWRL Rule



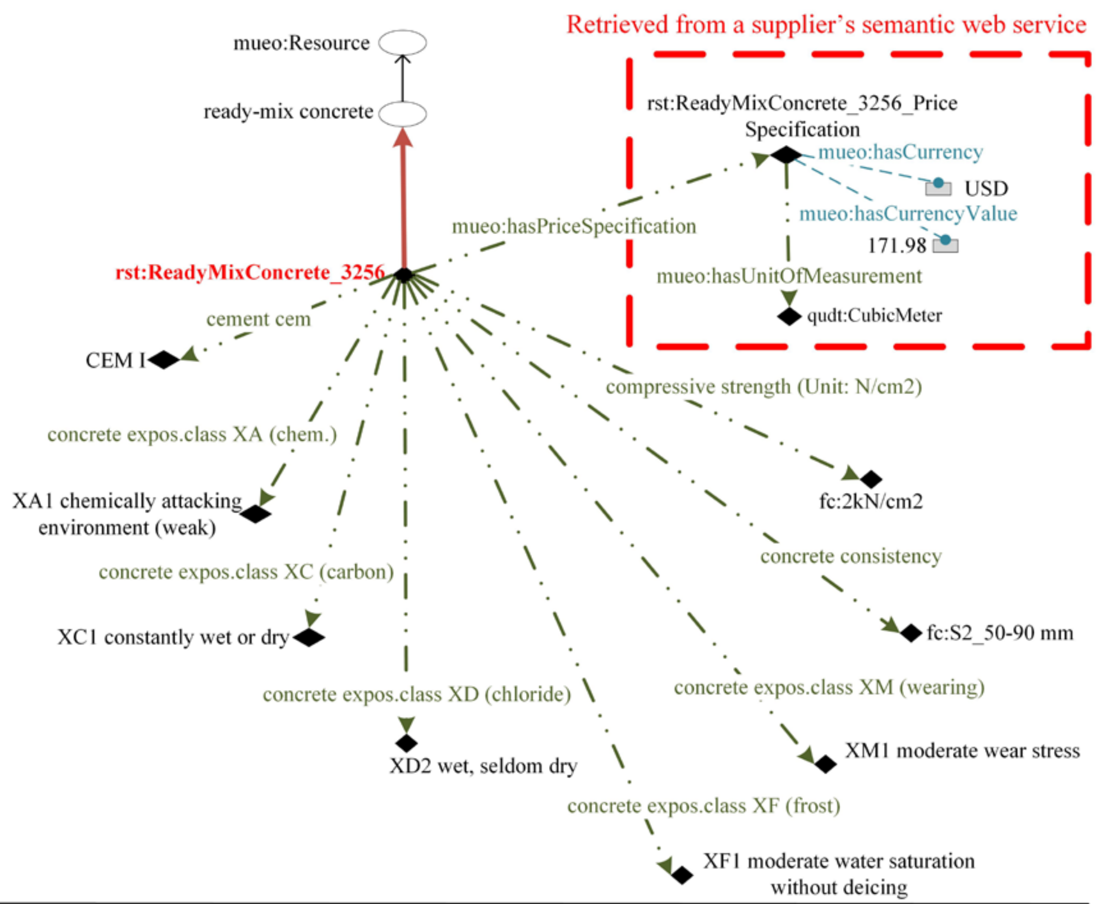
(b) Visual representation of the rule

KB* = Knowledge Base

Figure 4-6: SWRL rule for calculating placing concrete work item quantity

The work item shown in Figure 4-4 uses a material resource; the material URI is rst:ReadyMixConcrete_3256. A material is defined with properties for material specifications and unit cost as shown in Figure 4-7. FC ontology is used to represent material specifications. For example, cement type is specified using its FC URI http://www.freeclass.eu/freeclass_v1#P_E52 with the label “cement cem”. Unit cost property of material is a multi-value property that is specified using a currency, a currency value, and a unit of measurement.

The estimating application sends material specifications to material suppliers’ semantic web services and retrieves the latest material unit cost information. More details about the FC ontology will be provided in chapter 5.


Retrieved from a supplier's semantic web service
mueo:Resource
ready-mix concrete
rst:ReadyMixConcrete_3256_Price Specification
mueo:hasCurrency
USD
mueo:hasCurrencyValue
171.98
mueo:hasPriceSpecification
mueo:hasUnitOfMeasurement
qudt:CubicMeter
rst:ReadyMixConcrete_3256
cement cem
CEM I
compressive strength (Unit: N/cm2)
concrete expos.class XA (chem.)
fc:2kN/cm2
XA1 chemically attacking environment (weak)
concrete consistency
concrete expos.class XC (carbon)
fc:S2_50-90 mm
XC1 constantly wet or dry
concrete expos.class XD (chloride)
concrete expos.class XM (wearing)
XD2 wet, seldom dry
XM1 moderate wear stress
concrete expos.class XF (frost)
XF1 moderate water saturation without deicing

@prefix mueo: http://www.marquette.edu/estimating_ontology#
@prefix rst: http://www.RST_MaterialSupplierCompany.com/#
@prefix qudt: http://qudt.org/vocab/unit#
@prefix fc: http://www.freeclass.eu/freeclass_v1#

rdfs:label	URI
ready-mix concrete	http://www.freeclass.eu/freeclass_v1#C_A140-gen
cement cem	http://www.freeclass.eu/freeclass_v1#P_E52
CEM I	http://www.freeclass.eu/freeclass_v1#V_W167
concrete expos.class XA (chem.)	http://www.freeclass.eu/freeclass_v1#P_E27
XA1 chemically attacking environment (weak)	http://www.freeclass.eu/freeclass_v1#V_W87
concrete expos.class XC (carbon)	http://www.freeclass.eu/freeclass_v1#P_E31
XC1 constantly wet or dry	http://www.freeclass.eu/freeclass_v1#V_W101
concrete expos.class XD (chloride)	http://www.freeclass.eu/freeclass_v1#P_E32
XD2 wet, seldom dry	http://www.freeclass.eu/freeclass_v1#V_W106
concrete expos.class XF (frost)	http://www.freeclass.eu/freeclass_v1#P_E28
XF1 moderate water saturation without deicing	http://www.freeclass.eu/freeclass_v1#V_W91
concrete expos.class XM (wearing)	http://www.freeclass.eu/freeclass_v1#P_E29
XM1 moderate wear stress	http://www.freeclass.eu/freeclass_v1# V_W95
concrete consistency	http://www.freeclass.eu/freeclass_v1# P_E26
compressive strength (Unit: N/cm2)	http://www.freeclass.eu/freeclass_v1# P_16

Figure 4-7: A concrete material specification and unit cost

Figure 4-8 shows semantic descriptions of the crew resource used in the placing concrete work item shown in Figure 4-4. The work item crew consists of a cement finisher, two laborers, a concrete pump, and a concrete vibrator. Labor properties include labor classification, specialty, location, cost per hour, and quantity. Friend of a Friend ontology (foaf) (Brickley and Miller 2012) is used to describe persons, their activities, and their relations to other persons and objects. Equipment descriptions include classification, cost per hour and quantity.

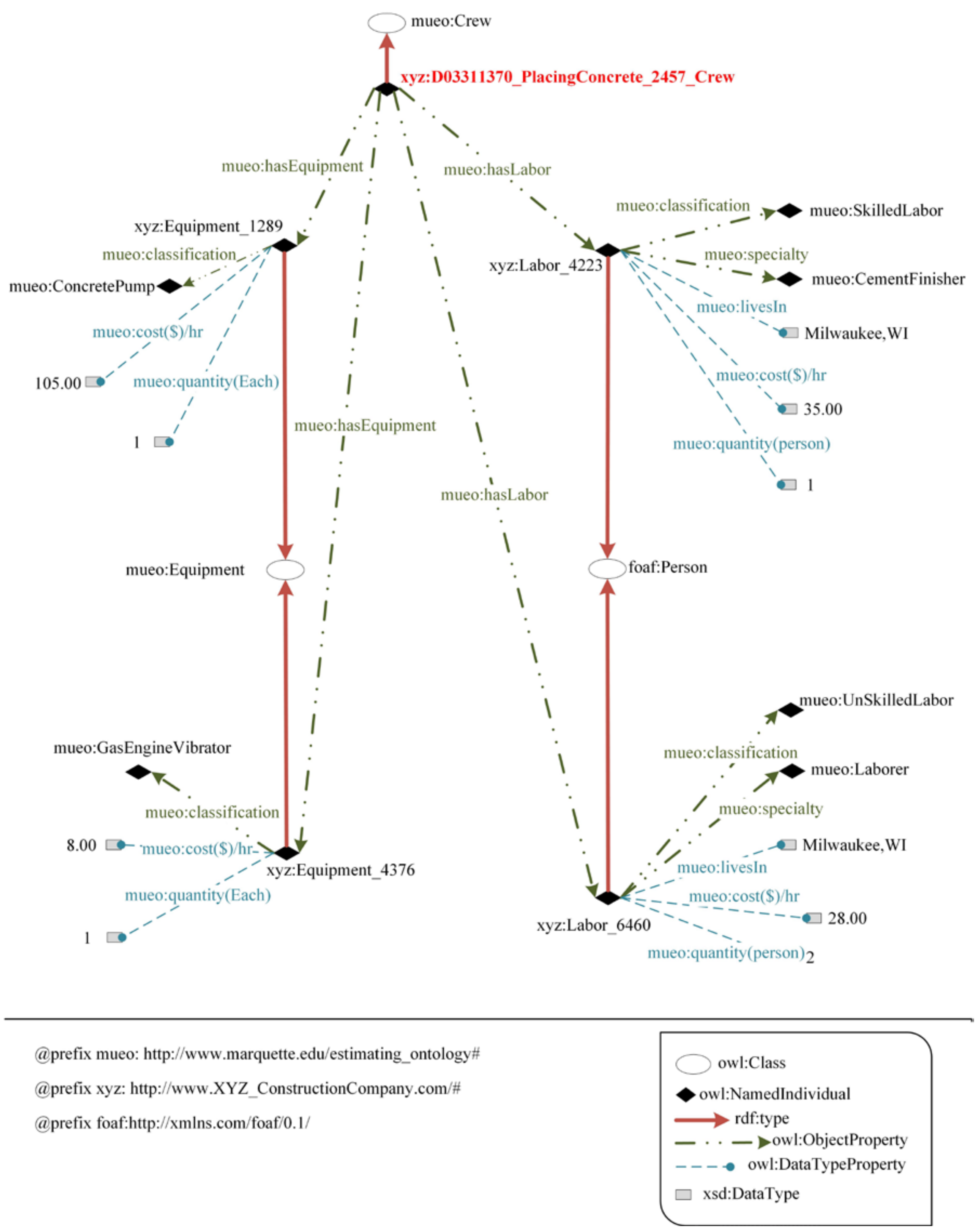


Figure 4-8: Crew representation

Figure 4-9 shows the multivalued productivity property of a work item; to define the productivity property a value and a unit of measurement for the value is needed.

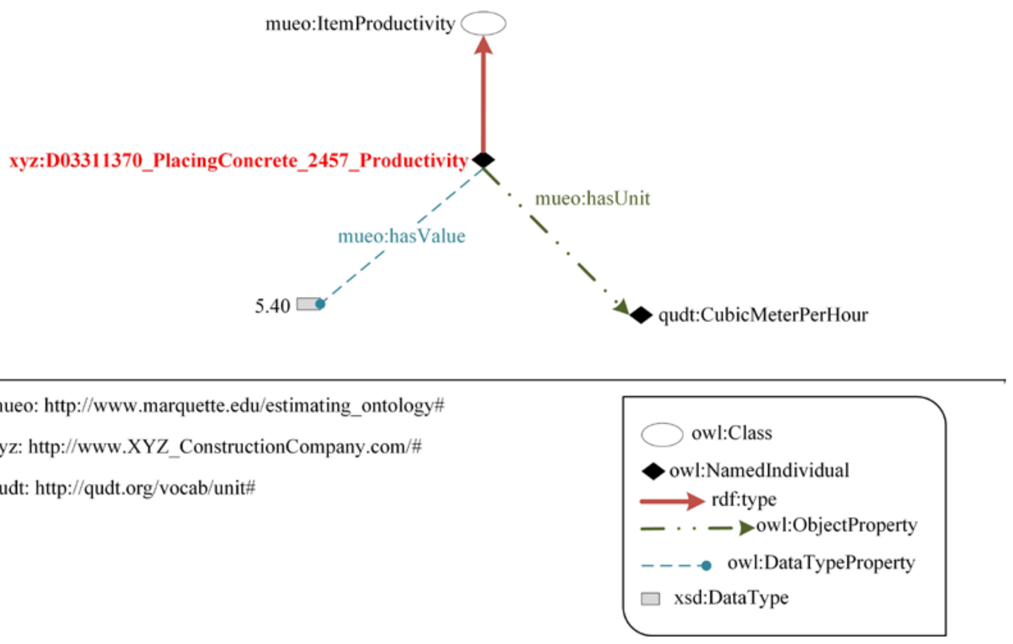

mueo:ItemProductivity
xyz:D03311370_PlacingConcrete_2457_Productivity
mueo:hasUnit
mueo:hasValue
5.40
qudt:CubicMeterPerHour
@prefix mueo: http://www.marquette.edu/estimating_ontology#
@prefix xyz: http://www.XYZ_ConstructionCompany.com/#
@prefix qudt: http://qudt.org/vocab/unit#
owl:Class
owl:NamedIndividual
rdf:type
owl:ObjectProperty
owl:DataTypeProperty
xsd:DataType

Figure 4-9: Productivity

Semantic representation of the multi-valued cost property of a placing concrete work item is shown in Figure 4-10.

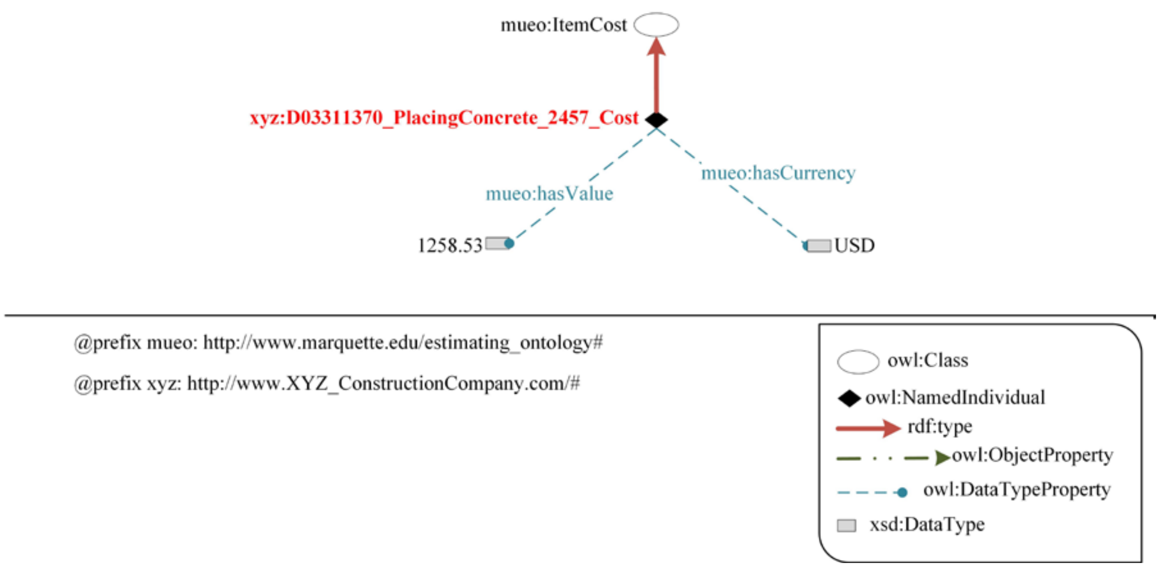

mueo:ItemCost
xyz:D03311370_PlacingConcrete_2457_Cost
mueo:hasValue
mueo:hasCurrency
1258.53
USD
@prefix mueo: http://www.marquette.edu/estimating_ontology#
@prefix xyz: http://www.XYZ_ConstructionCompany.com/#
owl:Class
owl:NamedIndividual
rdf:type
owl:ObjectProperty
owl:DataTypeProperty
xsd:DataType

Figure 4-10: Work item cost representation

The above approach is used to define different estimating assemblies and work items. RDF/XML representation of the assembly and work item examples given in this chapter are shown in Appendix C. The assembly and work item ontologies are developed in Protégé software. Java programming language and Appache Jena (Apache Jena 2015) library are used for creating assembly and work item instances; the estimating assembly knowledge base (including ontologies and instances) are saved to an OpenRDF Sesame triplestore (Sesame 2015). Jena is a Java framework for creating computer applications using the Semantic Web. Jena includes a collection of tools and Java libraries to support Semantic Web programming which includes reading, processing, writing, reasoning, storing, and querying RDF and OWL data. Jena supports SPARQL queries and rule-based inference engines such as Pellet Reasoner.

CHAPTER 5
MATERIAL SUPPLIERS' SEMANTIC WEB SERVICES

This chapter explains material suppliers' semantic web services as it was shown in Figure 2-3 in chapter 2. The purpose of this chapter is to provide material supplier products' information in a way that computer programs can access it over the Internet. This chapter presents web services technology and how those can be used for information exchange between material suppliers and an estimating application. The web services technology have been enhanced with Semantic Web technology which can be referred to as Semantic Web Services. This chapter presents how material suppliers' semantic web services are developed in this study.

5.1 Web Services

Web services provide standard means for interoperability between different software applications independent of the software platforms and/or frameworks (http://www.w3.org/TR/ws-arch/#introduction). Web services are designed to support interoperable machine-to-machine interaction over the Internet. The function of a web service is usually to supply information and/or exchange or sale of goods or obligations (Martin et al. 2007). Enterprise applications use service-oriented architecture to communicate with web services provided by different parties (Li et al. 2010).

Web Services Description Language (WSDL) (http://www.w3.org/TR/wsdl) is an XML format for describing web service interfaces. Other computer applications interact with a web service in a manner prescribed in its WSDL descriptions. The interaction between a computer application and a web service is performed in Simple Object Access

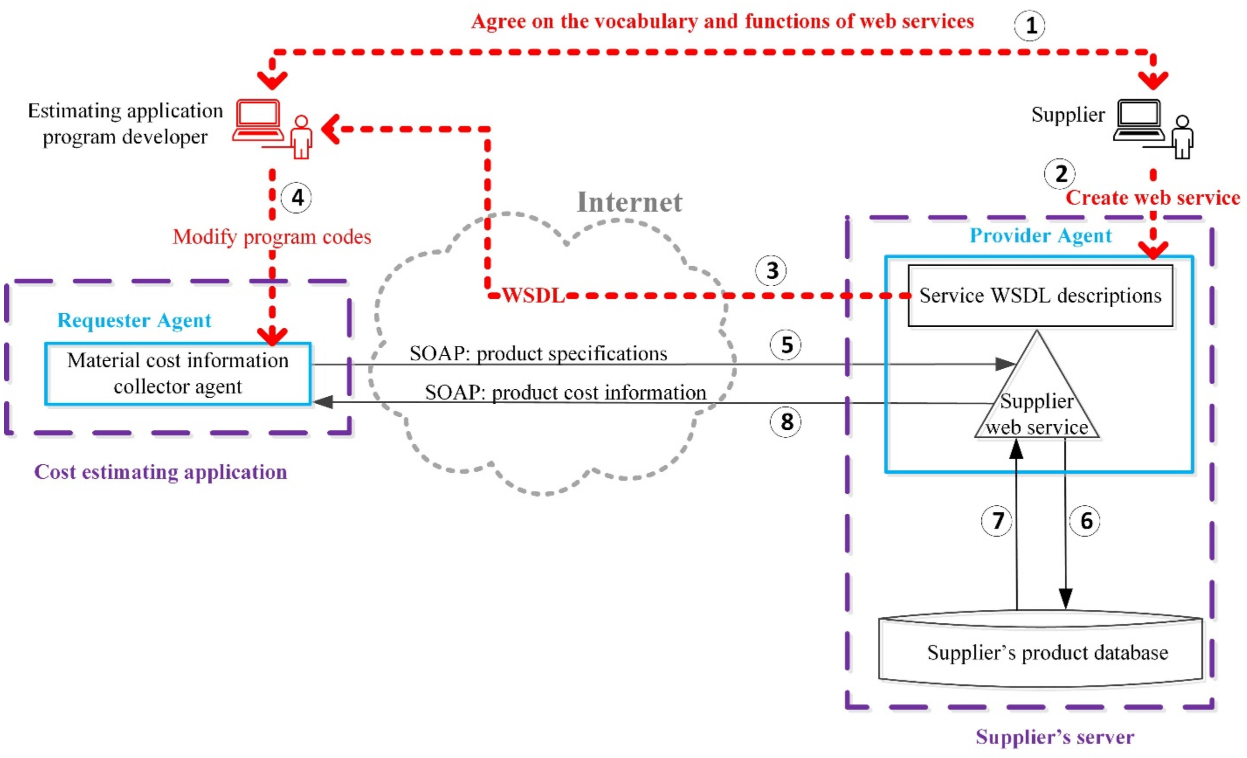
Protocol (SOAP) (http://www.w3schools.com/webservices/ws_soap_intro.asp) messages in XML serialization over HTTP. SOAP enables applications running on different operating systems in different programming languages to communicate information over the Internet.

A software agent is a program acting on behalf a person or an organization using web service technologies (http://www.w3.org/TR/ws-arch/#introduction). A provider agent provides a particular service and functionality on behalf of its owner. Whereas, a requester agent wishes to make use of a provider agent web service.

Web services can be used in the construction industry to support information exchange between a contractor and construction material resource suppliers. In the case of construction cost estimating, a material resource supplier can provide its product cost information as a web service. Then, an estimating application could send requests to the supplier web services to retrieve the latest supplier product cost information. In this study, the service oriented architecture for cost estimating was tested in a computer lab. Figure 5-1 shows the architecture of an estimating application that accesses a supplier web service. In Figure 5-1, the estimating application (on the left hand) has a requester agent software that communicates with a supplier provider agent (on the right hand) to retrieve the latest material resource cost data. This architecture requires the following steps:

1. A resource supplier and estimating application developer should agree on the vocabulary and functions based on which supplier should develop its web service.
2. A supplier should create its web service based on the agreement achieved in step 1.

3. The supplier sends WSDL descriptions of its service to the estimating application program developer.
4. Estimating application developer modifies the estimating application programming codes based on the WSDL descriptions to be able to communicate with the supplier web service.
5. After the estimating application programming codes are modified, the estimating application is able to send construction material resource specifications to the supplier web service in order to retrieve supplier product cost information.
6. Supplier web service queries the supplier database based on the product specifications retrieved from the estimating application to find the corresponding product in the supplier database.
7. Database returns product cost information to the supplier web service
8. Supplier web service returns product cost information to the cost estimating application.



Agree on the vocabulary and functions of web services
1
Estimating application program developer
Supplier
2
Create web service
4
Modify program codes
Internet
3
WSDL
Provider Agent
Service WSDL descriptions
Requester Agent
Material cost information collector agent
SOAP: product specifications
5
SOAP: product cost information
8
Supplier web service
Cost estimating application
7
6
Supplier's product database
Supplier's server

Figure 5-1: Service-oriented architecture for retrieving material cost data

Web services are designed to support interoperable machine-to-machine interaction; still, they need human actions as explained in steps 1 to 4 in the above architecture. The following section describes how Semantic Web Services can overcome some of the limitations of web services.

5.2 Semantic Web Services

Although service-oriented architecture can facilitate interoperability, still human involvement is needed in discovering and understanding functions of web services. Web service technologies operate at the syntactic level (Roman et al. 2005); but, they do not specify what happens when a service is executed. This means that computer applications that use a particular service must be revised if an existing service is modified or a new service is created. The large number of resource suppliers involved in construction projects can cause difficulties for estimating application developers. A cost estimating application would require special programming code for each material supplier web service.

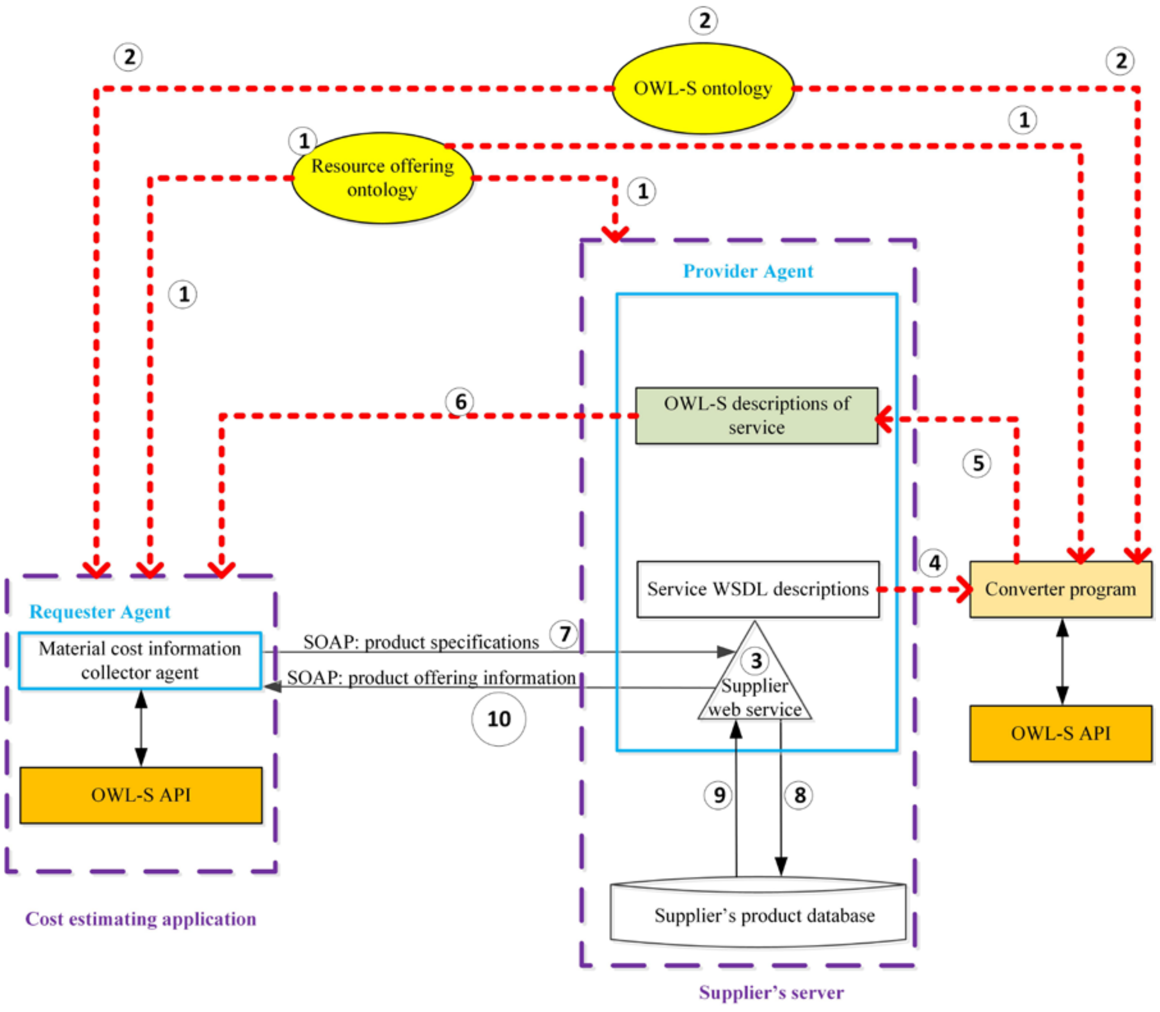
E-COGNOS (Lima et al. 2005) was a European project that developed web services to manage ontologies in the construction industry. E-COGNOS used other classifications such as IFC and ISO 12006-3 to develop ontologies. They used aspects of semantic web to document and update organizational information. E-COGNOS utilized a set of web service related technologies such as Simple Object Access Protocol (SOAP), Universal Discovery Description and Integration (UDDI), Web Services Description Language (WSDL), and XML. Their web services are intended to enable users to manage ontologies for specific tasks. E-COGNOS is not intended to enable computer applications find and execute web services provided by other organizations.

Ontologies can be used to semantically describe web services; which are known as Semantic Web Services (Grasic and Podgorelec 2010). Semantic descriptions of web services can automate service discovery and execution (Fensel et al. 2011; Niknam and Karshenas 2015b). In this study, ontologies are used to semantically define resource suppliers' web services and map them to estimating resource ontologies. This allows an estimating application to facilitate resource suppliers' web service discovery and execution.

Ready mixed concrete material resource suppliers' semantic web services were developed on a server in a computer lab to test the presented semantic web service approach. To retrieve a material resource cost data, the estimating application communicates with a supplier’s semantic web service. Figure 5-2 shows the architecture of accessing a supplier’s semantic web service. The estimating application developed in this study uses a requester agent to communicate with a supplier’s semantic web service. The function of this agent is to send product specifications to the supplier’s semantic web service and to receive supplier’s product offering information; this type of communication is in Simple Object Access Protocol (SOAP) format. This new architecture requires the following steps:

1. A resource offering ontology is needed to model supplier product offering information that includes product specifications and cost.
2. OWL-S ontology (http://www.w3.org/Submission/OWL-S/) of web services is needed to provide semantic descriptions of web services.
3. A resource supplier develops its web service in a way that its inputs and outputs are defined according the resource offering ontology.
4. A converter program reads service WSDL descriptions, resource offering ontology, and OWL-S ontology to create OWL-S descriptions of service.
5. The converter program sends OWL-S descriptions to the supplier service provider agent.
6. Supplier provider agent sends OWL-S descriptions of supplier web service to the estimating application.
7. Estimating application sends product specifications as a SOAP message to the supplier web service.
8. Supplier web service queries the supplier database to find the corresponding product information.

9. The supplier database sends the query results to the supplier web service.
10. The supplier web service sends product offering information to the estimating application that includes product cost information.



OWL-S ontology
Resource offering ontology
Provider Agent
OWL-S descriptions of service
Service WSDL descriptions
Converter program
OWL-S API
Requester Agent
Material cost information collector agent
OWL-S API
SOAP: product specifications
SOAP: product offering information
Supplier web service
Supplier's product database
Cost estimating application
Supplier's server
1
2
3
4
5
6
7
8
9
10

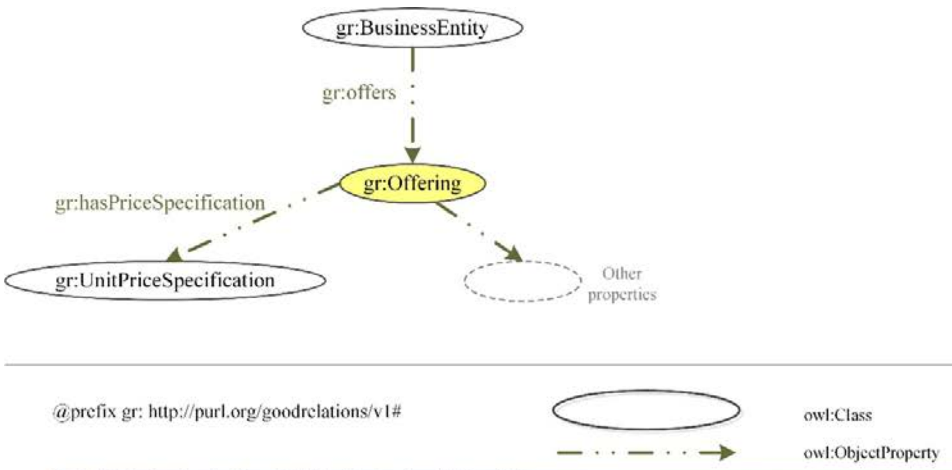
Figure 5-2: Accessing a supplier's semantic web service

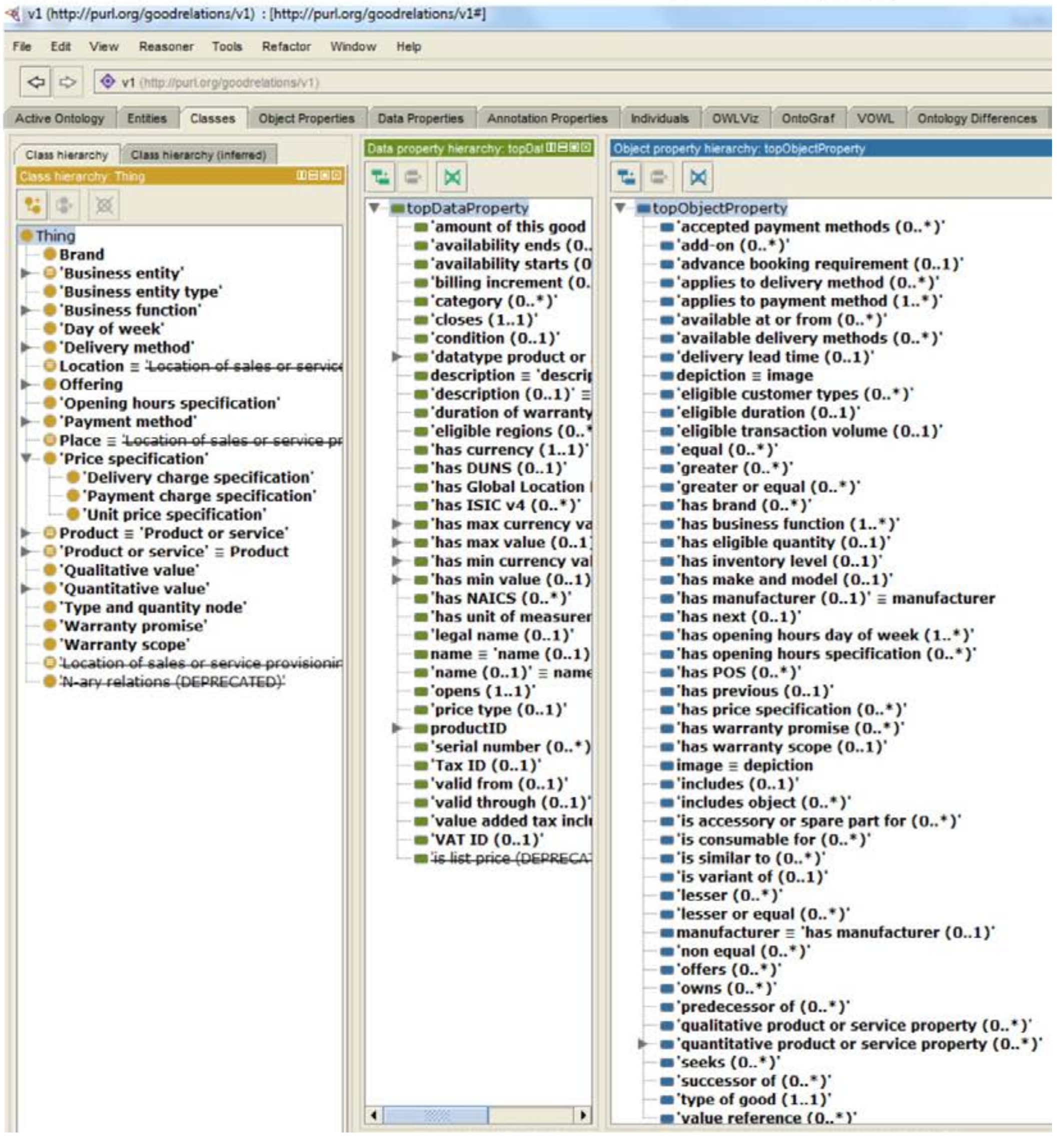
To create a SOAP message, the estimating application's requester agent accesses resource offering ontology and web service semantic descriptions (OWL-S descriptions) over the Internet. These ontologies are discussed below.

5.2.1 Resource Offering Ontology

The estimating application presented in this study retrieves material resource cost information from supplier's semantic web services. Construction material suppliers provide offerings that include material resource specifications and cost information. A resource offering ontology is required in order to semantically define construction material suppliers' offerings. Good Relations (GR 2008) ontology is used in this study for this purpose. GR allows businesses semantically define their product offerings and publish them on the Internet (Hepp 2008); it provides a conceptual model for general concepts such as company, product descriptions, store location, offer, price, payment, warranty, and shipment information (Hepp et al. 2009). GR has been adopted and widely used in web-based eCommerce (Ashraf et al. 2011; Fürber and Hepp 2010; Allemang and Hendler 2011). For example, Best Buy and overstock.com, two of the largest retail chains for consumer products, have used GR to semantically define their product data and publish them on the Internet. Search engines such as Google also use GR to enhance the information they provide in response to a search query about a product. In this research, GR is used to represent semantic descriptions of construction material suppliers' offerings.

Figure 5-3 shows the ontology for a business entity that offers an offering. In Figure 5-3, gr is the prefix of Good Relations ontology URI and equals to http://purl.org/goodrelations/v1#. For example, gr:Offering equals to http://purl.org/goodrelations/v1#Offering. In a similar manner the URIs for other concepts and properties are defined.




v1 (http://purl.org/goodrelations/v1) : [http://purl.org/goodrelations/v1#]

File Edit View Reasoner Tools Refactor Window Help

v1 (http://purl.org/goodrelations/v1)

Active Ontology | Entities | Classes | Object Properties | Data Properties | Annotation Properties | Individuals | OWLViz | OntoGraf | VOWL | Ontology Differences

Class hierarchy | Class hierarchy (inferred)

Class hierarchy: Thing

- Thing
  - Brand
  - 'Business entity'
  - 'Business entity type'
  - 'Business function'
  - 'Day of week'
  - 'Delivery method'
  - Location ≡ ~~'Location of sales or servic~~
  - Offering
  - 'Opening hours specification'
  - 'Payment method'
  - Place ≡ ~~'Location of sales or service pr~~
  - 'Price specification'
    - 'Delivery charge specification'
    - 'Payment charge specification'
    - 'Unit price specification'
  - Product ≡ 'Product or service'
  - 'Product or service' ≡ Product
  - 'Qualitative value'
  - 'Quantitative value'
  - 'Type and quantity node'
  - 'Warranty promise'
  - 'Warranty scope'
  - ~~'Location of sales or service provisionir~~
  - ~~'N-ary relations (DEPRECATED)'~~

Data property hierarchy: topDat

- topDataProperty
  - 'amount of this good
  - 'availability ends (0..
  - 'availability starts (0
  - 'billing increment (0.
  - 'category (0..*)'
  - 'closes (1..1)'
  - 'condition (0..1)'
  - 'datatype product or
  - description ≡ 'descri
  - 'description (0..1)' ≡
  - 'duration of warranty
  - 'eligible regions (0..*
  - 'has currency (1..1)'
  - 'has DUNS (0..1)'
  - 'has Global Location
  - 'has ISIC v4 (0..*)'
  - 'has max currency va
  - 'has max value (0..1
  - 'has min currency va
  - 'has min value (0..1)
  - 'has NAICS (0..*)'
  - 'has unit of measurer
  - 'legal name (0..1)'
  - name ≡ 'name (0..1)
  - 'name (0..1)' ≡ name
  - 'opens (1..1)'
  - 'price type (0..1)'
  - productID
  - 'serial number (0..*)
  - 'Tax ID (0..1)'
  - 'valid from (0..1)'
  - 'valid through (0..1)'
  - 'value added tax incl
  - 'VAT ID (0..1)'
  - ~~'is list price (DEPRECA~~

Object property hierarchy: topObjectProperty

- topObjectProperty
  - 'accepted payment methods (0..*)'
  - 'add-on (0..*)'
  - 'advance booking requirement (0..1)'
  - 'applies to delivery method (0..*)'
  - 'applies to payment method (1..*)'
  - 'available at or from (0..*)'
  - 'available delivery methods (0..*)'
  - 'delivery lead time (0..1)'
  - depiction ≡ image
  - 'eligible customer types (0..*)'
  - 'eligible duration (0..1)'
  - 'eligible transaction volume (0..1)'
  - 'equal (0..*)'
  - 'greater (0..*)'
  - 'greater or equal (0..*)'
  - 'has brand (0..*)'
  - 'has business function (1..*)'
  - 'has eligible quantity (0..1)'
  - 'has inventory level (0..1)'
  - 'has make and model (0..1)'
  - 'has manufacturer (0..1)' ≡ manufacturer
  - 'has next (0..1)'
  - 'has opening hours day of week (1..*)'
  - 'has opening hours specification (0..*)'
  - 'has POS (0..*)'
  - 'has previous (0..1)'
  - 'has price specification (0..*)'
  - 'has warranty promise (0..*)'
  - 'has warranty scope (0..1)'
  - image ≡ depiction
  - 'includes (0..1)'
  - 'includes object (0..*)'
  - 'is accessory or spare part for (0..*)'
  - 'is consumable for (0..*)'
  - 'is similar to (0..*)'
  - 'is variant of (0..1)'
  - 'lesser (0..*)'
  - 'lesser or equal (0..*)'
  - manufacturer ≡ 'has manufacturer (0..1)'
  - 'non equal (0..*)'
  - 'offers (0..*)'
  - 'owns (0..*)'
  - 'predecessor of (0..*)'
  - 'qualitative product or service property (0..*)'
  - 'quantitative product or service property (0..*)'
  - 'seeks (0..*)'
  - 'successor of (0..*)'
  - 'type of good (1..1)'
  - 'value reference (0..*)'

Figure 5-3: Partial view of Good Relations ontology

An offering has properties such as: gr:acceptedPaymentMethods, gr:advanceBookingRequirement, gr:availabilityStarts, gr:availabilityEnds, gr:availableAtOrFrom, gr:availableDeliveryMethods, gr:category, gr:condition, gr:deliveryLeadTime, gr:description, gr:eligibleCustomerTypes, gr:eligibleDuration, gr:eligibleRegions, gr:eligibleTransactionVolume, gr:hasBusinessFunction, gr:hasEligibleQuantity, gr:hasInventoryLevel, gr:hasPriceSpecification, gr:hasStockKeepingUnit, gr:hasWarrantyPromise, gr:includes, gr:includesObject, gr:name, gr:serialNumber, gr:validFrom, and gr:validThrough.

At top of Figure 5-3, only unit price specification of an offering is shown. A unit price specification describes the currency, currency value, and unit of measurement for an offer. Other properties of an offer are included in the ontology as shown at the bottom of Figure 5-3 in the Protégé software screenshot. In the protégé screenshot, the left panel shows classes, the middle panel shows data properties, and the right panel shows object properties in GR.

A business entity offers its offerings that include a product or services. Figure 5-4 shows the ontology where an offering includes a product or service. Each product or service is described with quantitative and qualitative values. In Figure 5-4, fc is the prefix of Free Class OWL ontology to describe construction and building materials as it will be explained in the following section. fc:ConstructionAndBuildingMaterials is subclass of gr:ProductOrService.

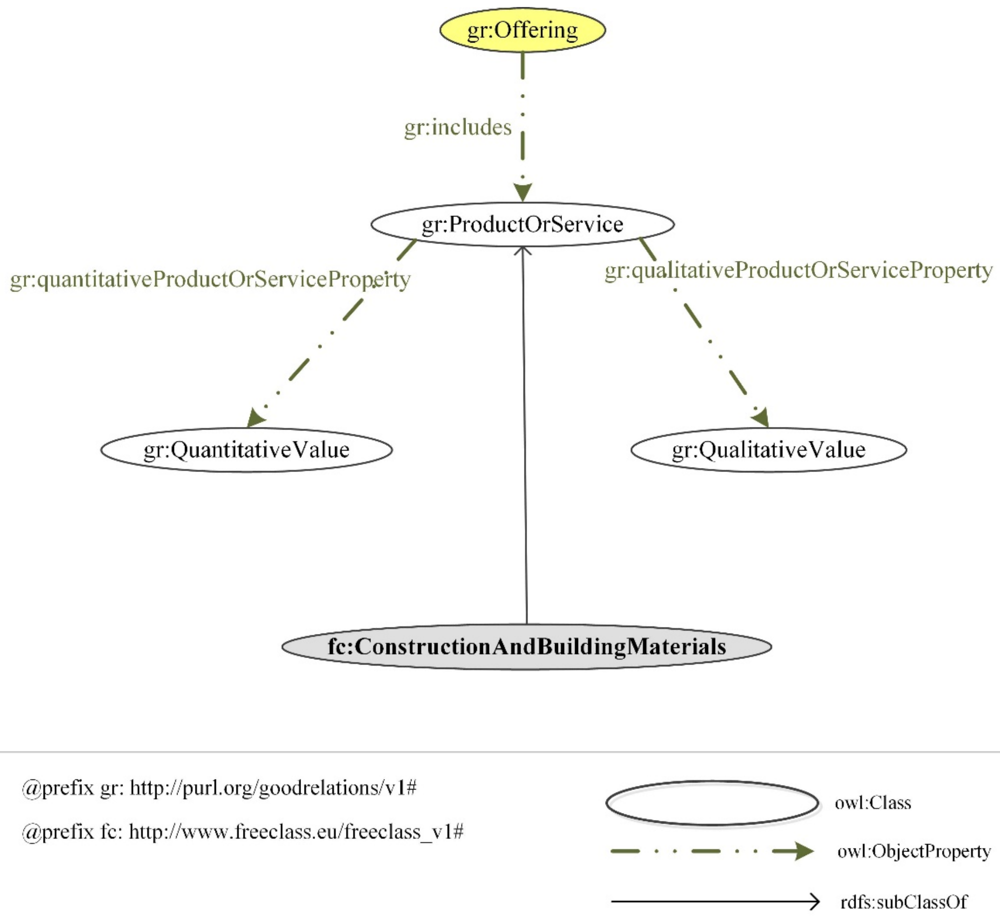
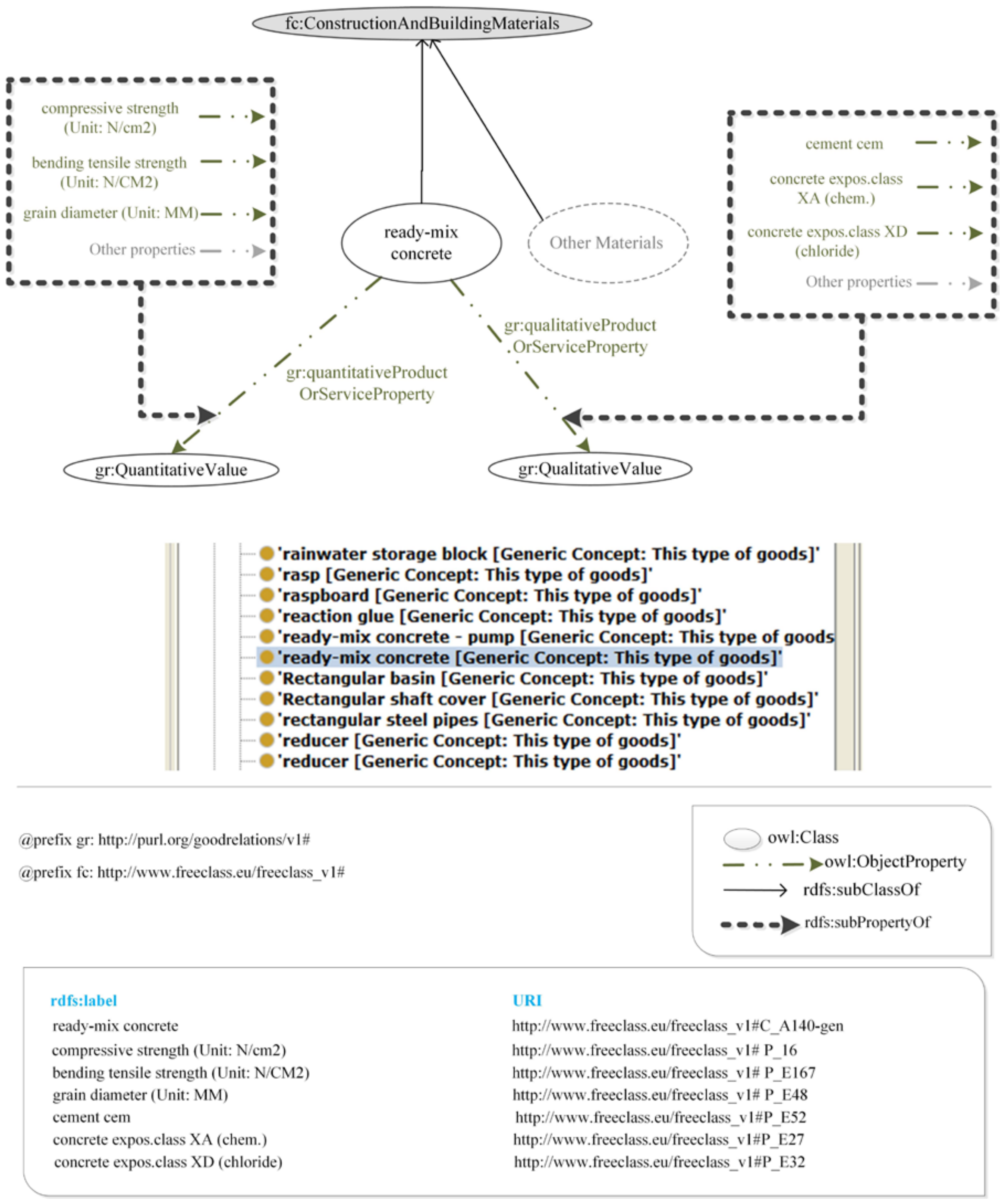

gr:Offering
gr:includes
gr:ProductOrService
gr:quantitativeProductOrServiceProperty
gr:qualitativeProductOrServiceProperty
gr:QuantitativeValue
gr:QualitativeValue
fc:ConstructionAndBuildingMaterials
@prefix gr: http://purl.org/goodrelations/v1#
@prefix fc: http://www.freeclass.eu/freeclass_v1#
owl:Class
owl:ObjectProperty
rdfs:subClassOf

Figure 5-4: A product or service ontology

5.2.2 Material Ontology

The European Building and Construction Materials Database developed Free Class OWL ontology (FC) to describe construction materials and services. FC has over 88 million triples of real business information to represent construction material and services (BauDataWeb 2015). FC is a GR and W3C compliant ontology. FC and GR vocabularies allow construction material suppliers to semantically define their construction product offerings. Figure 5-5 shows ready-mix concrete material class in FC ontology. In FC, every material is described with quantitative and qualitative properties. Several quantitative properties of concrete such as compressive strength, bending tensile strength, and grain diameter are shown on the left side of Figure 5-5; these FC properties are sub-properties of the gr:quantitativeProductOrServiceProperty. On the right hand of Figure 5-5, some of the qualitative properties defined in FC for concrete such as cemenet cem, and concrete exposure classes are presented; these properties are sub-properties of gr:qualitativeProductOrServiceProperty. Other FC quantitative and qualitative properties are defined in the ontology in a similar manner. FC assigns labels to different classes and properties URIs. The URI of the FC labels are shown at the bottom of Figure 5-5.


fc:ConstructionAndBuildingMaterials
compressive strength (Unit: N/cm2)
bending tensile strength (Unit: N/CM2)
grain diameter (Unit: MM)
Other properties
ready-mix concrete
Other Materials
cement cem
concrete expos.class XA (chem.)
concrete expos.class XD (chloride)
Other properties
gr:quantitativeProductOrServiceProperty
gr:qualitativeProductOrServiceProperty
gr:QuantitativeValue
gr:QualitativeValue
owl:Class
owl:ObjectProperty
rdfs:subClassOf
rdfs:subPropertyOf

'rainwater storage block [Generic Concept: This type of goods]'
'rasp [Generic Concept: This type of goods]'
'raspboard [Generic Concept: This type of goods]'
'reaction glue [Generic Concept: This type of goods]'
'ready-mix concrete - pump [Generic Concept: This type of goods
'ready-mix concrete [Generic Concept: This type of goods]'
'Rectangular basin [Generic Concept: This type of goods]'
'Rectangular shaft cover [Generic Concept: This type of goods]'
'rectangular steel pipes [Generic Concept: This type of goods]'
'reducer [Generic Concept: This type of goods]'
'reducer [Generic Concept: This type of goods]'

@prefix gr: http://purl.org/goodrelations/v1#

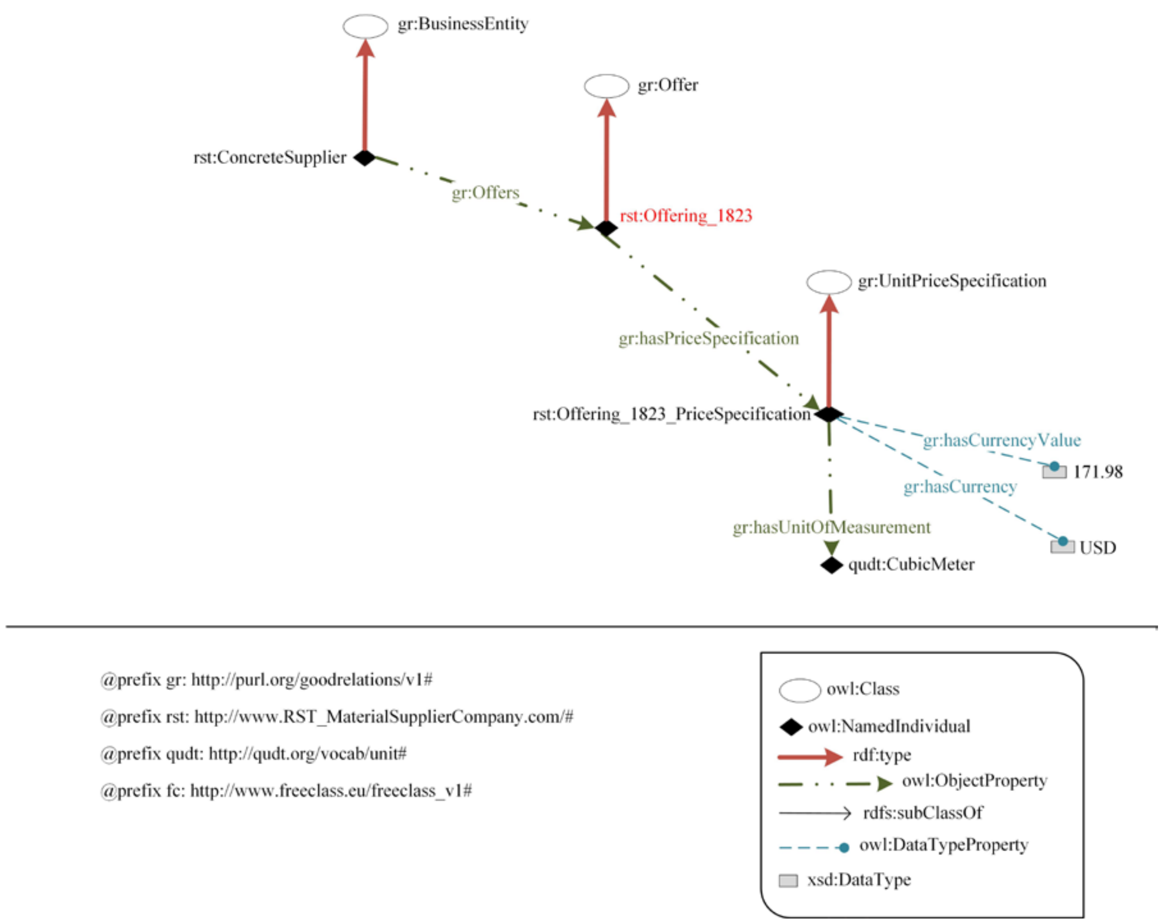
@prefix fc: http://www.freeclass.eu/freeclass_v1#

rdfs:label	URI
ready-mix concrete	http://www.freeclass.eu/freeclass_v1#C_A140-gen
compressive strength (Unit: N/cm2)	http://www.freeclass.eu/freeclass_v1# P_16
bending tensile strength (Unit: N/CM2)	http://www.freeclass.eu/freeclass_v1# P_E167
grain diameter (Unit: MM)	http://www.freeclass.eu/freeclass_v1# P_E48
cement cem	http://www.freeclass.eu/freeclass_v1#P_E52
concrete expos.class XA (chem.)	http://www.freeclass.eu/freeclass_v1#P_E27
concrete expos.class XD (chloride)	http://www.freeclass.eu/freeclass_v1#P_E32

Figure 5-5: Ready-mix concrete in FC ontology

In this study, GR and FC are used to develop suppliers' product offering semantic web services. Inputs and outputs of suppliers' semantic web services are defined according to GR and FC. The estimating application developed in this study uses GR and FC to send product specifications to suppliers' semantic web services and retrieve the latest material cost information.

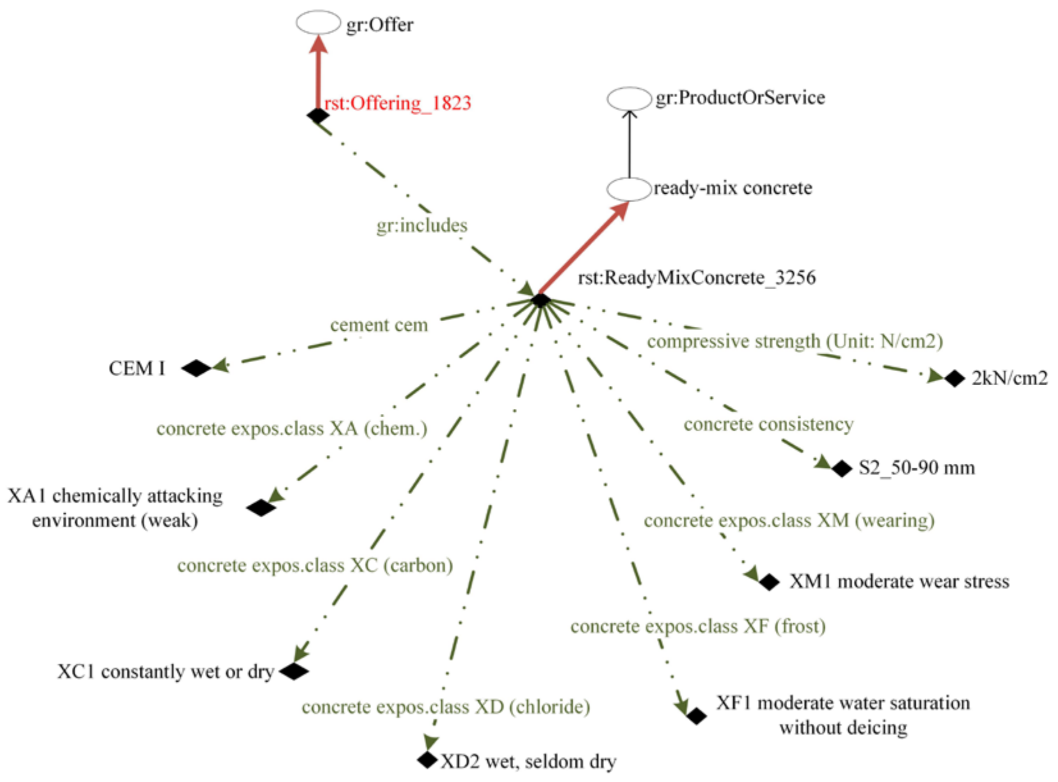
Figure 5-6 shows a section of a ready mixed concrete supplier knowledge base developed in this study. In the list of prefixes, "rst" is the prefix of a ready mix concrete material supplier URI that equals to http://www.RST_MaterialSupplierCompany.com/#. So, rst:Offering_1823 defines the unit cost and specifications for a ready mixed concrete product. GR vocabulary enables suppliers to describe their offerings with information such as price, accepted payment methods, store location, warranty, available delivery methods, and advanced booking requirements. Figure 5-6 shows price specification for an offering that is described with currency, currency value, and unit of measurement. Similarly other properties of each offer are included in the supplier knowledge base using the GR vocabulary.



```
@prefix gr: http://purl.org/goodrelations/v1#
@prefix rst: http://www.RST_MaterialSupplierCompany.com/#
@prefix qudt: http://qudt.org/vocab/unit#
@prefix fc: http://www.freeclass.eu/freeclass_v1#
```

Figure 5-6: Part of a material supplier knowledge base-Price Specification

Figure 5-7 shows that rst:Offering_1823 includes a ready mixed concrete product as rst:ReadyMixConcrete_3256. Figure 5-7 shows different properties of this ready-mixed concrete product such as cement CEM, consistency, compressive strength and concrete exposure to chemicals, carbon, chloride, frost and wearing. RDF/XML representation of the example provided in figures 5-6 and 5-7 is given in Appendix D.


gr:Offer
rst:Offering_1823
gr:ProductOrService
ready-mix concrete
gr:includes
rst:ReadyMixConcrete_3256
cement cem
CEM I
compressive strength (Unit: N/cm2)
2kN/cm2
concrete expos.class XA (chem.)
concrete consistency
S2_50-90 mm
XA1 chemically attacking environment (weak)
concrete expos.class XM (wearing)
concrete expos.class XC (carbon)
XM1 moderate wear stress
concrete expos.class XF (frost)
XC1 constantly wet or dry
XF1 moderate water saturation without deicing
concrete expos.class XD (chloride)
XD2 wet, seldom dry

@prefix gr: http://purl.org/goodrelations/v1#

@prefix rst: http://www.RST_MaterialSupplierCompany.com/#

@prefix qudt: http://qudt.org/vocab/unit#

@prefix fc: http://www.freeclass.eu/freeclass_v1#

○	owl:Class
◆	owl:NamedIndividual
➔	rdf:type
– · · –▶	owl:ObjectProperty
⟶	rdfs:subClassOf
– – – ●	owl:DataTypeProperty
▭	xsd:DataType

rdfs:label	URI
ready-mix concrete	http://www.freeclass.eu/freeclass_v1#C_A140-gen
cement cem	http://www.freeclass.eu/freeclass_v1#P_E52
CEM I	http://www.freeclass.eu/freeclass_v1#V_W167
concrete expos.class XA (chem.)	http://www.freeclass.eu/freeclass_v1#P_E27
XA1 chemically attacking environment (weak)	http://www.freeclass.eu/freeclass_v1#V_W87
concrete expos.class XC (carbon)	http://www.freeclass.eu/freeclass_v1#P_E31
XC1 constantly wet or dry	http://www.freeclass.eu/freeclass_v1#V_W101
concrete expos.class XD (chloride)	http://www.freeclass.eu/freeclass_v1#P_E32
XD2 wet, seldom dry	http://www.freeclass.eu/freeclass_v1#V_W106
concrete expos.class XF (frost)	http://www.freeclass.eu/freeclass_v1#P_E28
XF1 moderate water saturation without deicing	http://www.freeclass.eu/freeclass_v1#V_W91
concrete expos.class XM (wearing)	http://www.freeclass.eu/freeclass_v1#P_E29
XM1 moderate wear stress	http://www.freeclass.eu/freeclass_v1# V_W95
concrete consistency	http://www.freeclass.eu/freeclass_v1# P_E26
compressive strength (Unit: N/cm2)	http://www.freeclass.eu/freeclass_v1# P_16

Figure 5-7: Part of a material supplier knowledge base-Material Specification

Concrete supplier semantic web services are created according to GR and FC ontologies. The function of these web services is to receive concrete material specification and return supplier product offering information that includes unit price specifications. The following section describes OWL-S ontology in order to enable an estimating application communicate with a supplier semantic web service.

5.2.3 OWL-S Descriptions

Semantic descriptions of suppliers' web services were created to enable the estimating application to identify which services must be accessed, prepare input messages for those services, execute web services, and integrate data returned from the web services into the estimating knowledge bases. In this study, OWL-S (W3C 2004c) was used to provide semantic web service descriptions. OWL-S provides computer-interpretable descriptions of web services and the means by which they are accessed (Martin et al. 2004a). OWL-S is an ontology within the OWL framework of the Semantic Web technology for describing semantic web services. It enables users and software agents to discover, invoke, compose, and monitor web services (http://www.w3.org/Submission/OWL-S/). The estimating application developed in this study uses these descriptions to learn how to use services. OWL-S provides semantic descriptions for web services by three sub-ontologies: service profile ontology, process model ontology, and grounding ontology as shown in Figure 5-8. Below is a brief description for each of the OWL-S sub-ontologies:

1. **OWL-S service profile ontology** provides semantic description of what a service does along with the limitations, quality, and requirements of the service. The

estimating application developed in this study uses service profile descriptions to find an appropriate material supplier web service.

2. **OWL-S model ontology** describes how to use and request a service. It describes what happens when that service is requested. The model ontology is also used to compose and coordinate different web services.
3. **OWL-S service grounding ontology** specifies communication protocols, message formats, port numbers, and other details that describe how a web service can be accessed.

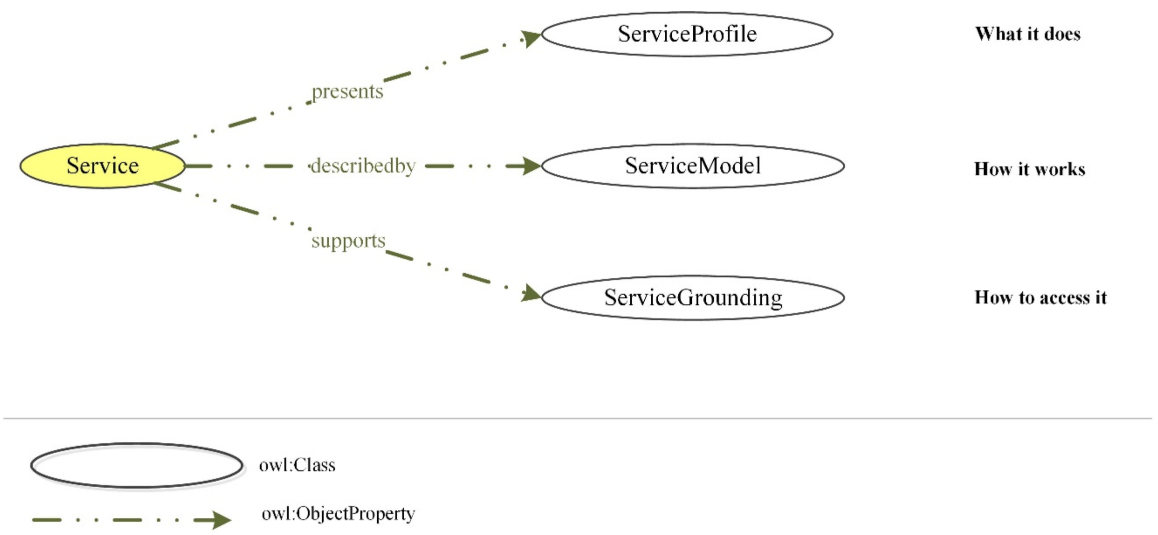
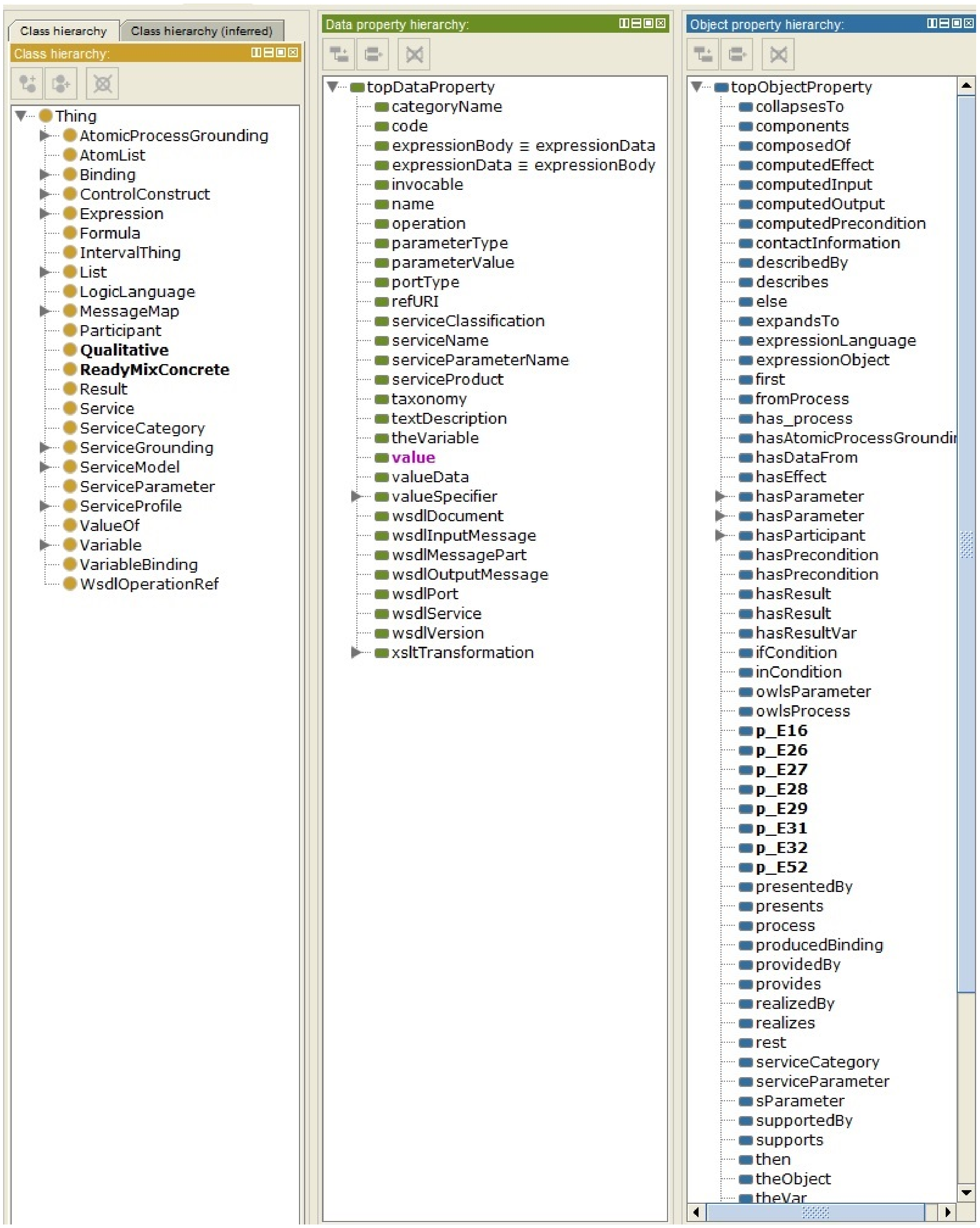


Figure 5-8: OWL-S ontology

Figure 5-9 shows a screenshot of Protégé software that includes OWL-S to describe construction material suppliers' semantic web services. RDF/XML descriptions for a concrete material supplier semantic web service is given Appendix E.



Thing
AtomicProcessGrounding
AtomList
Binding
ControlConstruct
Expression
Formula
IntervalThing
List
LogicLanguage
MessageMap
Participant
Qualitative
ReadyMixConcrete
Result
Service
ServiceCategory
ServiceGrounding
ServiceModel
ServiceParameter
ServiceProfile
ValueOf
Variable
VariableBinding
WsdlOperationRef

topDataProperty
categoryName
code
expressionBody ≡ expressionData
expressionData ≡ expressionBody
invocable
name
operation
parameterType
parameterValue
portType
refURI
serviceClassification
serviceName
serviceParameterName
serviceProduct
taxonomy
textDescription
theVariable
value
valueData
valueSpecifier
wsdlDocument
wsdlInputMessage
wsdlMessagePart
wsdlOutputMessage
wsdlPort
wsdlService
wsdlVersion
xsltTransformation

topObjectProperty
collapsesTo
components
composedOf
computedEffect
computedInput
computedOutput
computedPrecondition
contactInformation
describedBy
describes
else
expandsTo
expressionLanguage
expressionObject
first
fromProcess
has_process
hasAtomicProcessGroundi
hasDataFrom
hasEffect
hasParameter
hasParameter
hasParticipant
hasPrecondition
hasPrecondition
hasResult
hasResult
hasResultVar
ifCondition
inCondition
owlsParameter
owlsProcess
p_E16
p_E26
p_E27
p_E28
p_E29
p_E31
p_E32
p_E52
presentedBy
presents
process
producedBinding
providedBy
provides
realizedBy
realizes
rest
serviceCategory
serviceParameter
sParameter
supportedBy
supports
then
theObject
theVar

Figure 5-9: A screenshot of OWL-S ontology in Protégé

Supplier semantic web services prototypes for this study were developed using Java programming language. The inputs and outputs of these web services were defined according to GR and FC ontologies. OWL-S descriptions of the web services were created using the OWL-S API, which provides a Java API for programmatic access to create, read, write, and execute semantic web services.

Using the new estimating approach requires construction material suppliers to semantically define and publish their product specifications and cost information on the web as semantic web services. Currently, none of the construction material suppliers use ontologies for describing their products. However, several large consumer companies such as Best Buy and Overstock.com have started to use ontologies to semantically define their products and services for web commerce. These companies have successfully implemented product offering ontologies such as Good Relations. So, the technology is available if the construction industry decides to use it.

CHAPTER 6
IMPLEMENTATION AND VALIDATION

6.1 Implementation

This chapter describes a prototype estimating application developed in this study based on the estimating architecture presented in chapter 2. In this architecture, an estimating application accesses a BIM knowledge base, an estimating assembly and work item knowledge base, and material suppliers' semantic web services to estimate the cost of a building project. A cost estimating application is developed in this study using Java programming language to test the new architecture. OpenRDF Sesame triplestores are used to store the BIM and estimating assembly and work item knowledge bases. Jena framework is used to bring the capabilities of Pellet reasoner to the estimating application. A semantic web service communication module is developed as part of the estimating application using the OWL-S API.

Figure 6-1 shows the use cases developed in the prototype estimating application. The most basic functionality for an estimating application are provided in the prototype. As shown in Figure 6-1, an estimator selects a building model, prepares a list of material suppliers, defines estimating assemblies and work items, maps BIM elements to estimating assemblies, and performs an estimate. To perform these activities, estimating application needs access to a BIM knowledge base, an estimating assembly and work item knowledge base, and material suppliers' semantic web services.

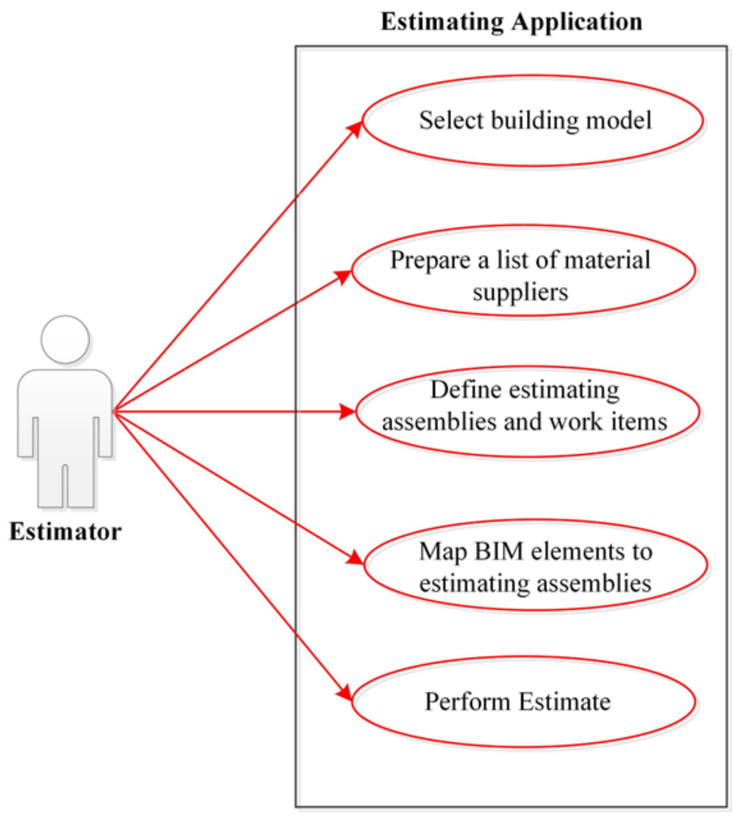

Estimating Application
Select building model
Prepare a list of material suppliers
Define estimating assemblies and work items
Map BIM elements to estimating assemblies
Perform Estimate
Estimator

Figure 6-1: Estimator use cases

A brief description for each use case is provided below:

A) **Select building model**. Using the estimating application user interface, a user can select a building model from a Sesame BIM triple-store server. The user enters the URI of the building to be estimated to select the building model. For example, the URI of the building project used in this study is abc:EngineeringHall as shown in chapter 3. The estimating application uses the building project URI entered by the user to stablish a connection with the BIM knowledge base of the building project stored in a Sesame server.

B) **Prepare a list of material suppliers**. The user creates a list of material suppliers for the project using the estimating application user interfaces. The material supplier list includes information about the suppliers such as supplier names and URIs.

C) **Define estimating assemblies and work items**. Predefined estimating assemblies and work items are required as discussed in chapter 4. The prototype estimating application has interfaces that allow a user to create work items and assemblies and store them in a Sesame knowledge base. The work item user interface allows a user to create a new work item and enter the properties discussed in chapter 4 for the work item. The user also selects the material supplier for the work item from the list of material suppliers. Estimating application assembly user interface enables the user to define a new assembly and assign a number of predefined work items to the assembly.

D) **Map BIM elements to estimating assemblies.** An estimating assembly is created to estimate the cost of a building element. A mapping user interface provides the user with two lists: the first list includes the names of predefined estimating assemblies from the estimating knowledge base and the second list contains building element types retrieved from the BIM knowledge base. The user maps each model element type to the appropriate estimating assembly.

E) **Perform estimate**. A command button is developed to perform estimate, which executes a number of program processes as follow:

- Iterate through BIM elements in the BIM knowledge base.
- Retrieve the corresponding assembly from the estimating knowledge base for each BIM element.
- Retrieve the work items that are part of the assembly.
- Calculate work item quantity for each work item.
- Submit material specifications to the selected material supplier and receive material unit cost for each work item.
- Calculate work item cost.
- Calculate assembly cost by adding assembly work item costs.
- Calculate project cost by adding all project assembly costs.
- Return project cost.

6.2 Validation

A building project includes a large number of elements and material resources. The validation of prototype approach is limited to structural concrete elements in a 3 story educational building project named Engineering Hall. The validation included a total of 40 elements including footings, columns, beams, and slabs. For each element, length, width, and depth of each BIM element were mapped to its corresponding estimating assembly dimensions. Validation included only concrete material and concrete pouring work item.

The prototype semantics-based estimating application was validated for accuracy of results and its impact on estimator efficiency in a computer lab at Marquette University. The cost estimates using the prototype estimating application for 40 elements were compared with those obtained using a commercial computer estimating application called WinEst (WinEst 2015). The prototype application produced the same cost estimating results as WinEst software. It is worth mentioning that since the prototype application always runs the same procedures it has the same accuracy every time it runs whereas in manual approaches human error is inevitable.

In addition to accuracy, the prototype was validated for estimating efficiency by measuring the time it takes to create a cost estimate using the semantics-based approach developed in this study compared with WinEst estimating software. WinEst is a popular commercial estimating software and was readily available for this study as it is used in a construction cost estimating course at Marquette University.

Currently, to prepare a BIM-based cost estimate, an estimator must complete the following steps:

1. Edit predefined assemblies of work items in the estimating application that represent BIM elements to be estimated.
2. Map BIM elements to their corresponding estimating assemblies. An example is mapping a BIM rectangular footing element to a spread footing assembly in the estimating software.
3. Map BIM element properties to the estimating assembly properties for calculating work item quantities. For example, mapping the dimensions of a BIM footing element to the corresponding estimating assembly dimensions.
4. Update material unit costs in the estimating application database.

Currently, the above mentioned steps are manually performed by estimators. The prototype semantics-based approach presented in this study modifies how steps 3 and 4 are performed. To calculate work item quantities in step 3, the prototype estimating application eliminates the need for manually mapping element properties to the corresponding assembly properties as those are semantically defined. To update the estimating application material resource unit cost database in step 4, the prototype estimating application retrieves material unit costs directly from suppliers' semantic web services as explained in chapter 5. By eliminating the manual activities from steps 3 and 4, the prototype improves estimator efficiency. The followings explain in more details how steps 3 and 4 were performed.

6.2.1 Mapping BIM Element Properties to Estimating Assembly Properties

The efficiency of the prototype semantics-based approach relative to WinEst's method was investigated for calculating concrete quantities for the above-mentioned 40 structural elements in Engineering Hall. 5 students in civil engineering and familiar with WinEst software were taught the interfaces to the prototype estimating software and how to use the software for estimating purposes. According to Nielsen (2000), 5 users provide adequate results for validation purposes. The 5 students were asked to do the following steps in a computer lab at Marquette University:

1. Prepare the necessary estimating assemblies for concrete footings, beams, columns, and slabs.
2. Map each BIM element to its corresponding estimating assembly.

3. Map the dimensions of each BIM element to its corresponding estimating assembly dimensions to calculate the concrete quantity for the BIM element.

The students were given the same guidelines and instructions on how to prepare an estimate step by step. The time was measured using a stopwatch. The time required to complete the first two steps are almost the same in both WinEst and the prototype developed in this study. To perform step 3 in WinEst, it requires an estimator to manually map element dimensions to their corresponding assembly dimensions. In WinEst, the mapping required in step 3 took an average of 53 seconds per concrete element. The same 5 students also used the prototype estimating application developed in this study to estimate the cost of the same building elements. Since in the prototype estimating application, BIM elements and estimating assemblies are semantically defined, the mapping required in step 3 is machine processable and does not require estimator involvement. The total time saved for preparing the cost estimate for 40 concrete elements was about 35 minutes and 20 seconds. It is worth mentioning that the prototype eliminates human error in step 3 which makes it more accurate than a manual method.

Considering the fact that a construction project includes a large number of building elements, eliminating step 3 substantially improves estimator efficiency. Future work can investigate the efficiency of the presented approach for estimating the cost of building elements not included in the validation performed in this study.

6.2.2 Updating Estimating Application's Material Resource Unit Cost Database

In a computer lab, two web servers were developed representing two ready-mixed concrete material suppliers. Each web server published concrete material specifications and unit costs in two formats: (1) in table formats for human access and (2) as semantic web services for computer access. In a test, the same 5 students were asked to update a material unit cost database using the tabulated data published on suppliers' web sites. The students were asked to obtain the minimum unit costs for the concrete mixes needed for Engineering Hall from the supplier web sites and update material databases in WinEst. It took on the average 1 minute and 58 seconds for each student to update the unit cost for each of the 4 concrete mix specifications used in the 40 building elements investigated. The prototype estimating application developed in this study directly accessed suppliers' semantic web services, submitted the specifications for the required concrete mixes, obtained suppliers' material unit costs, and used the minimum material cost to update the estimating software's material database. The total time saved for updating unit costs of 4 concrete specifications was about 7 minutes and 52 seconds. In addition to saving time, the prototype always creates accurate results whereas in manual methods human error cannot be eliminated when retrieving and entering unit costs into a cost database. In a construction project, a large number of material unit costs must be updated; the results of this test shows the potential for reducing the time required for updating material cost databases. Future work can investigate the results for material types not used in this study.

CHAPTER 7
SUMMARY, CONCLUSION, AND RECOMMENDATIONS FOR FUTURE WORK

7.1 Summary and Conclusion

Preparing a construction cost estimate requires access to several sources of information. These sources include building information model (BIM) created by designers, estimating assemblies and work items created and maintained by construction companies, and material resource cost information provided by material suppliers. In this study, the application of Semantic Web and Semantic Web Service technologies to facilitate finding, accessing, and combining the information necessary for construction cost estimating was investigated. One of the objectives of this study was to represent the estimating domain information in a semantics-based format. Ontologies were used to create (1) a BIM knowledge base, (2) an estimating assembly and work item knowledge base, and (3) material supplier semantic web services. A prototype cost estimating application was developed that can access these knowledge bases and semantic web services to perform cost estimating.

To validate the prototype estimating application, its performance was compared with a commercial estimating program called WinEst. In a test, 5 students were asked to perform quantity takeoffs for 40 structural concrete elements in a 3 story building. The BIM elements investigated included concrete footings, columns, beams, and slabs. In WinEst estimating software, an estimator is required to manually map element dimensions to their corresponding estimating assembly dimensions. Each student spent on the average 53 seconds to map a concrete element's length, width and depth properties to its corresponding WinEst assembly properties. In the semantics-based prototype, BIM elements and estimating assemblies are semantically defined which makes the mapping machine processable and eliminates the need for estimator involvement in the element property mapping process. The total time saved for mapping 40 concrete elements' dimensions was about 35 minutes and 20 seconds. Future studies can investigate the efficiency of the semantic estimating approach for estimating a large building.

Updating material unit cost databases is another time consuming task for estimators, which is currently performed manually. A new approach was presented in this study that requires material specifications and costs available as semantic web services. In this approach, a semantics-based cost estimating application can submit material resource specifications to suppliers' semantic web services and retrieve material unit cost data for updating material cost databases. In a lab test, the presented approach for updating estimating material cost databases was compared with the current manual methods. It was observed that on the average it takes 1 minute and 58 seconds less time to update a concrete mix unit cost in a database if material suppliers' product data are available as semantic web services. The total time saved for updating unit costs for 4 concrete specifications used in 40 structural concrete elements was about 7 minutes and 52 seconds. In addition to time savings, the prototype creates accurate results whereas in a manual method human error cannot be always eliminated. Considering the fact that estimating databases include a large number of material resources and estimating material cost databases must be updated before a new estimate, future work can investigate the results when a large number of material types are involved.

The purpose of this study was to show the potential of the Semantic Web technology in construction cost estimating. Although semantic web technology is being used in web commerce, its application to construction cost estimating requires that the industry further defines and standardizes the ontologies, knowledge bases, and semantic web services discussed in this study.

7.2 Recommendations for Future Work

The estimating approach discussed in this study requires a number of knowledge bases. The required knowledge bases must be developed based on standard ontologies that are not available yet. The prototype knowledge bases used in this study were created based on ontologies that were developed in this study.

The AEC industry must develop a standard ontology that can be used for creating BIM knowledge bases. There have been some efforts to use EXPRESS-to-OWL conversion procedures for developing an ifcOWL ontology (Beetz et al. 2009; Karan et al. 2015; Karan and Irizarry 2015; Pauwels and Terkaj 2014). However, none of the ifcOWL ontologies have become a standard yet. The ontologies and knowledge bases created in this study were based on level of detail (LOD) currently available in BIM platforms. As BIM platform LOD improves over time, the ontologies and knowledge bases may require updates. In addition, this study mainly focused on building projects. Future work can investigate the presented approach on other types of civil and construction projects.

Standard ontologies are also needed for creating estimating knowledge bases that represent estimating assemblies and work items. The estimating ontologies presented in this study can serve as the starting point for further research towards development of industry standard ontologies.

As discussed in the dissertation, to make the process of updating the material cost databases machine processable, construction material suppliers must provide material information as knowledge bases that can be accessed over the Internet. The Good Relations ontology discussed in this study has been used by a number of businesses such as Best Buy, Overstock.com, Yahoo!, and Google (Allemang and Hendler 2011) for providing product data or for product search purposes. Future development of a standard material ontology would allow construction material suppliers to develop knowledge bases that can be searched and queried by remote computers.

Semantically defined construction knowledge can lead to other innovations in the AEC industry. Niknam and Karshenas (2014) have used semantically defined BIM and construction knowledge bases to create a project social networking website which improves communication among project participants and allows adding new knowledge to project knowledge bases. Incorporating social networking and construction cost estimating knowledge can inspire new and innovative approaches to construction cost estimating.

REFERENCES

K. Alexander. (2013). "The Difference Between a Triplestore and a Relational Database." http://krisalexander.com/innovation/2013/07/16/the-difference-between-a-triplestore-and-a-relational-database/ 2015).

Allemang, D., and Hendler, J. (2011). *Semantic web for the working ontologist: effective modeling in RDFS and OWL.* Elsevier, .

Apache Jena. (2015). "A free and open source Java framework for building Semantic Web and Linked Data applications." https://jena.apache.org/ (06/03, 2015).

Ashraf, J., Cyganiak, R., O'Riain, S., and Hadzic, M. (2011). "Open ebusiness ontology usage: Investigating community implementation of goodrelations." *Linked Data on the Web Workshop (LDOW 2011) at WWW,* .

ASTM standard. (2015). "Standard Classification for Building Elements and Related Sitework-UNIFORMAT II." http://www.astm.org/Standards/E1557.htm (06/03, 2015).

Autodesk. (2015a). "Revit." http://www.autodesk.com/products/revit-family/overview 2015).

Autodesk. (2015b). "Revit Building design and construction software." http://www.autodesk.com/products/revit-family/overview (06/03, 2015).

Azhar, S. (2011). "Building information modeling (BIM): Trends, benefits, risks, and challenges for the AEC industry." *Leadership and Management in Engineering,* 11(3), 241-252.

Baker, C. J., and Cheung, K. (2007). *Semantic web: Revolutionizing knowledge discovery in the life sciences.* Springer Science & Business Media, .

BauDataWeb. (2015). "The European Building and Construction Materials Database for the Semantic Web." http://semantic.eurobau.com/ (06/03, 2015).

Beetz, J., Van Leeuwen, J., and De Vries, B. (2009). "IfcOWL: A case of transforming EXPRESS schemas into ontologies." *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* 23(01), 89-101.

Bergman, M. (2009). "Advantages and Myths of RDF." *AI3, April,* .

T. Berners-Lee. (1998). "Why RDF model is different from the XML model." http://www.w3.org/DesignIssues/RDF-XML (06/02, 2015).

Brickley, D., and Miller, L. (2012). "FOAF vocabulary specification 0.98." *Namespace Document,* 9.

Bussler, C. (2002). "Modeling and executing semantic B2B integration." *Research Issues in Data Engineering: Engineering E-Commerce/E-Business Systems, 2002. RIDE-2EC 2002. Proceedings. Twelfth International Workshop on,* IEEE, 69-74.

Cambridge. (2015). "RDF vs. XML." https://www.cambridgesemantics.com/semantic-university/rdf-vs-xml (06/02, 2015).

Cerovsek, T. (2011). "A review and outlook for a 'Building Information Model'(BIM): A multi-standpoint framework for technological development." *Advanced Engineering Informatics,* 25(2), 224-244.

CSI. (2015). "MasterFormat." http://www.csinet.org/masterformat 2015).

Cunningham & Cunningham. (2014). "Object database." http://c2.com/cgi/wiki?ObjectOrientedDatabase 2015).

R. Cyganiak, and Jentzsch, A. (2010). "Introduction to: Linked Data." http://www.dataversity.net/linked-data-an-introduction/ (06/02, 2015).

De Farias, T. M., Roxin, A., and Nicolle, C. (2014). "A Rule Based System for Semantical Enrichment of Building Information Exchange." *RuleML 2014,* 2.

Demir, S., Garrett Jr, J. H., Akinci, B., Akin, O., and Palmer, M. (2010). "A semantic web-based approach for representing and reasoning with vocabulary for computer based standards processing." *The Proceedings of the 2010 International Conference on Computing in Civil and Building Engineering (ICCCBE'10), Nottingham, UK,* .

Fensel, D., Facca, F. M., Simperl, E., and Toma, I. (2011). "Web Service Modeling Ontology." *Semantic Web Services,* 107-129.

Fernández-López, M., Gómez-Pérez, A., and Juristo, N. (1997). "Methontology: from ontological art towards ontological engineering." .

Fürber, C., and Hepp, M. (2010). "Using SPARQL and SPIN for Data Quality Management on the Semantic Web." *Business Information Systems,* Springer, 35-46.

Grasic, B., and Podgorelec, V. (2010). "Automating ontology based information integration using service orientation." *WSEAS Transactions on Computers,* 9(6), 547-556.

Gruber, T. R. (1995). "Toward principles for the design of ontologies used for knowledge sharing?" *International Journal of Human-Computer Studies,* 43(5), 907-928.

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

Gu, N., and London, K. (2010). "Understanding and facilitating BIM adoption in the AEC industry." *Autom.Constr.,* 19(8), 988-999.

Hebeler, J., Fisher, M., Blace, R., and Perez-Lopez, A. (2011a). *Semantic web programming.* John Wiley & Sons, .

Hebeler, J., Fisher, M., Blace, R., and Perez-Lopez, A. (2011b). *Semantic web programming.* John Wiley & Sons, .

M. Hepp. (2015). "GoodRelations." http://www.heppnetz.de/projects/goodrelations/ 2015).

Hepp, M. (2008). "Goodrelations: An ontology for describing products and services offers on the web." *Knowledge Engineering: Practice and Patterns,* Springer, 329-346.

Hepp, M., Radinger, A., Wechselberger, A., Stolz, A., Bingel, D., Irmscher, T., Mattern, M., and Ostheim, T. (2009). "GoodRelations Tools and Applications." *Poster and Demo Proceedings of the 8th International Semantic Web Conference, ISWC,* Citeseer, .

Hitzler, P., Krotzsch, M., and Rudolph, S. (2011). *Foundations of semantic web technologies.* Chapman and Hall/CRC, .

Hobbs, J. R., and Pan, F. (2006). "Time ontology in OWL." *W3C Working Draft,* 27 133.

Hodgson, R., Keller, P. J., Hodges, J., and Spivak, J. (2011). "QUDT–Quantities, Units, Dimensions and Data Types Ontologies. 2013." *URL Http://Qudt.Org,* .

Hore, A. V., and West, R. (2007). "CITAX: Defining XML Standards for Data Exchange in the Construction Industry Supply Chain." .

Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., and Dean, M. (2004). "SWRL: A semantic web rule language combining OWL and RuleML." *W3C Member Submission,* 21 79.

Karan, E., Irizarry, J., and Haymaker, J. (2015). "Generating IFC Models from heterogeneous data using semantic web." *Construction Innovation,* 15(2),.

Karan, E. P., and Irizarry, J. (2015). "Extending BIM interoperability to preconstruction operations using geospatial analyses and semantic web services." *Autom.Constr.,* 53(0), 1-12.

Karshenas, S., and Niknam, M. (2013). "Ontology-based building information modeling." *Computing in Civil Engineering (2013),* 476-483.

Knublauch, H., Oberle, D., Tetlow, P., Wallace, E., Pan, J., and Uschold, M. (2006). "A semantic web primer for object-oriented software developers." *W3c Working Group Note, W3C,* .

Li, G., Muthusamy, V., and Jacobsen, H. A. (2010). "A distributed service-oriented architecture for business process execution." *ACM Transactions on the Web (TWEB),* 4(1), 2.

Lima, C., El-Diraby, T., and Stephens, J. (2005). "Ontology-based optimization of knowledge management in e-construction." *Journal of IT in Construction,* 10 305-327.

A. Malhotra. (2007). "Integrating Data from Relational Databases Using RDF." http://www.w3.org/2007/03/RdfRDB/papers/malhotra.pdf 2015).

Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., and Payne, T. (2004a). "OWL-S: Semantic markup for web services." *W3C Member Submission,* 22 2007-2004.

Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., and Payne, T. (2004b). "OWL-S: Semantic markup for web services." *W3C Member Submission,* 22 2007-2004.

Martin, D., Burstein, M., Mcdermott, D., Mcilraith, S., Paolucci, M., Sycara, K., Mcguinness, D. L., Sirin, E., and Srinivasan, N. (2007). "Bringing semantics to web services with OWL-S." *World Wide Web,* 10(3), 243-277.

Mascardi, V., Locoro, A., and Rosso, P. (2010). "Automatic ontology matching via upper ontologies: A systematic evaluation." *Knowledge and Data Engineering, IEEE Transactions On,* 22(5), 609-623.

J. Nielsen. (2000). "Why You Only Need to Test with 5 Users." http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/ 2015).

Niknam, M., and Karshenas, S. (2015a). "Integrating distributed sources of information for construction cost estimating using Semantic Web and Semantic Web Service technologies." *Automation in Construction,* 57 222-238.

Niknam, M., and Karshenas, S. (2015b). "Sustainable Design of Buildings using Semantic BIM and Semantic Web Services." *Procedia Engineering,* 118 909-917.

Niknam, M., and Karshenas, S. (2014). "A Social Networking Website for AEC Projects." *Computing in Civil and Building Engineering (2014),* ASCE, 2208-2215.

Niknam, M., and Karshenas, S. (2013). "A Semantic Web Service Approach to Construction Cost Estimating." *Computing in Civil Engineering (2013),* 484-491.

Noy, N. F., and McGuinness, D. L. (2001). "Ontology development 101: A guide to creating your first ontology." .

M. Obitko. (2007). "Ontologies and Semantic Web." http://obitko.com/tutorials/ontologies-semantic-web/ 2015).

O'Leary, D. E. (1997). "Impediments in the use of explicit ontologies for KBS development." *International Journal of Human-Computer Studies,* 46(2), 327-337.

P. Pauwels, and Terkaj, W. (2014). "ifcOWL ontology <www.w3.org/community/lbd/ifcOWL>." 2015).

Pauwels, P., Van Deursen, D., De Roo, J., Van Ackere, T., De Meyer, R., Van de Walle, R., and Van Campenhout, J. (2011a). "Three-dimensional information exchange over the semantic web for the domain of architecture, engineering, and construction." *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* 25(04), 317-332.

Pauwels, P., Van Deursen, D., Verstraeten, R., De Roo, J., De Meyer, R., Van de Walle, R., and Van Campenhout, J. (2011b). "A semantic rule checking environment for building performance checking." *Autom.Constr.,* 20(5), 506-518.

Pellet. (2014). https://github.com/complexible/pellet (06/03, 2015).

Pinto, H. S., Staab, S., and Tempich, C. (2004). "DILIGENT: Towards a fine-grained methodology for DIstributed, Loosely-controlled and evolvInG Engineering of oNTologies." *ECAI,* Citeseer, 393.

Prud'Hommeaux, E., and Seaborne, A. (2008). "SPARQL query language for RDF." *W3C Recommendation,* 15.

Roman, D., Keller, U., Lausen, H., de Bruijn, J., Lara, R., Stollberg, M., Polleres, A., Feier, C., Bussler, C., and Fensel, D. (2005). "Web service modeling ontology." *Applied Ontology,* 1(1), 77-106.

RSMeans. (2015). "Construction Cost Data." http://www.rsmeans.com/ 2015).

Segaran, T., Evans, C., and Taylor, J. (2009a). *Programming the semantic web.* " O'Reilly Media, Inc.", .

Segaran, T., Evans, C., and Taylor, J. (2009b). *Programming the semantic web.* " O'Reilly Media, Inc.", .

J. Sequeda. (2012). "Introduction to: RDF vs XML." http://www.dataversity.net/introduction-to-rdf-vs-xml/ (06/02, 2015).

Sesame. (2015). "Sesame." http://rdf4j.org/ (06/03, 2015).

Stanford University. (2015). "Protege." http://protege.stanford.edu/ (06/02, 2015).

Suárez-Figueroa, M. C. (2012). "*NeOn Methodology for building ontology networks: specification, scheduling and reuse.* Dissertations in Artificial Intelligence. Vol. 338. IOS Press. ISBN 978-1-61499-115-1
Available at: http://oa.upm.es/3879/". PhD. Informatica, .

Suárez-Figueroa, M. C., García-Castro, R., Villazón-Terrazas, B., and Gómez-Pérez, A. (2011). "Essentials in ontology engineering: Methodologies, languages, and tools." .

Suárez-Figueroa, M. C., Gómez-Pérez, A., Motta, E., and Gangemi, A. (2012). *Ontology engineering in a networked world.* Springer Science & Business Media, .

Sure, Y., Staab, S., and Studer, R. (2004). "On-to-knowledge methodology (OTKM)." *Handbook on ontologies,* Springer, 117-132.

SWBPD. (2006). "A Semantic Web Primer for Object-Oriented Software Developers." http://www.w3.org/TR/sw-oosd-primer/ 2015).

Teicholz, P., Sacks, R., and Liston, K. (2011). *BIM handbook: a guide to building information modeling for owners, managers, designers, engineers, and contractors.* Wiley, .

Uschold, M., and King, M. (1995). *Towards a methodology for building ontologies.* Citeseer, .

W3C. (2014). "The organization ontology." http://www.w3.org/TR/vocab-org/ (06/04, 2015).

W3C. (2005). "Naming and Addressing: URIs, URLs." http://www.w3.org/Addressing/ (06/02, 2015).

W3C. (2004). "RDF/XML Syntax Specification." http://www.w3.org/TR/REC-rdf-syntax/ (06/02, 2015).

W3C RDF Working Group. (2014). "Resource Description Framework (RDF)." http://www.w3.org/RDF/ (06/02, 2015).

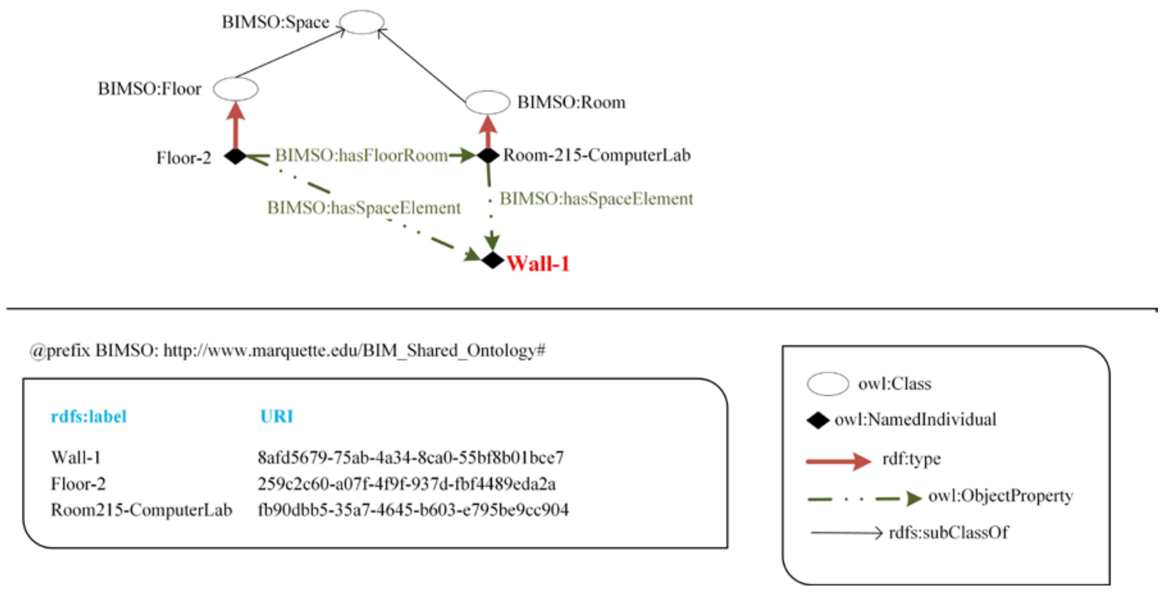
W3C Standard. (2015a). "Ontology." http://www.w3.org/standards/semanticweb/ontology 2015).

W3C Standard. (2015b). "Semantic Web." http://www.w3.org/standards/semanticweb/ 2015).

WinEst. (2015). http://www.meridiansystems.com/products/winest/overview/ 2015).

APPENDIX A
AN EXAMPLE OF ELEMENT RELATIONS TO FLOOR, ROOM, AND OTHER ELEMENTS

A BIM knowledge base must represent the relations of an element with the building floors, rooms, phases, and levels. An example of a footing element relations to a level and a phase was given in chapter 3. Figure A-1 shows how Wall-1 is related to spaces (floors, and rooms) defined for Engineering Hall. Wall-1 is located on Floor-2 and is a boundary of Room215-ComputerLab.

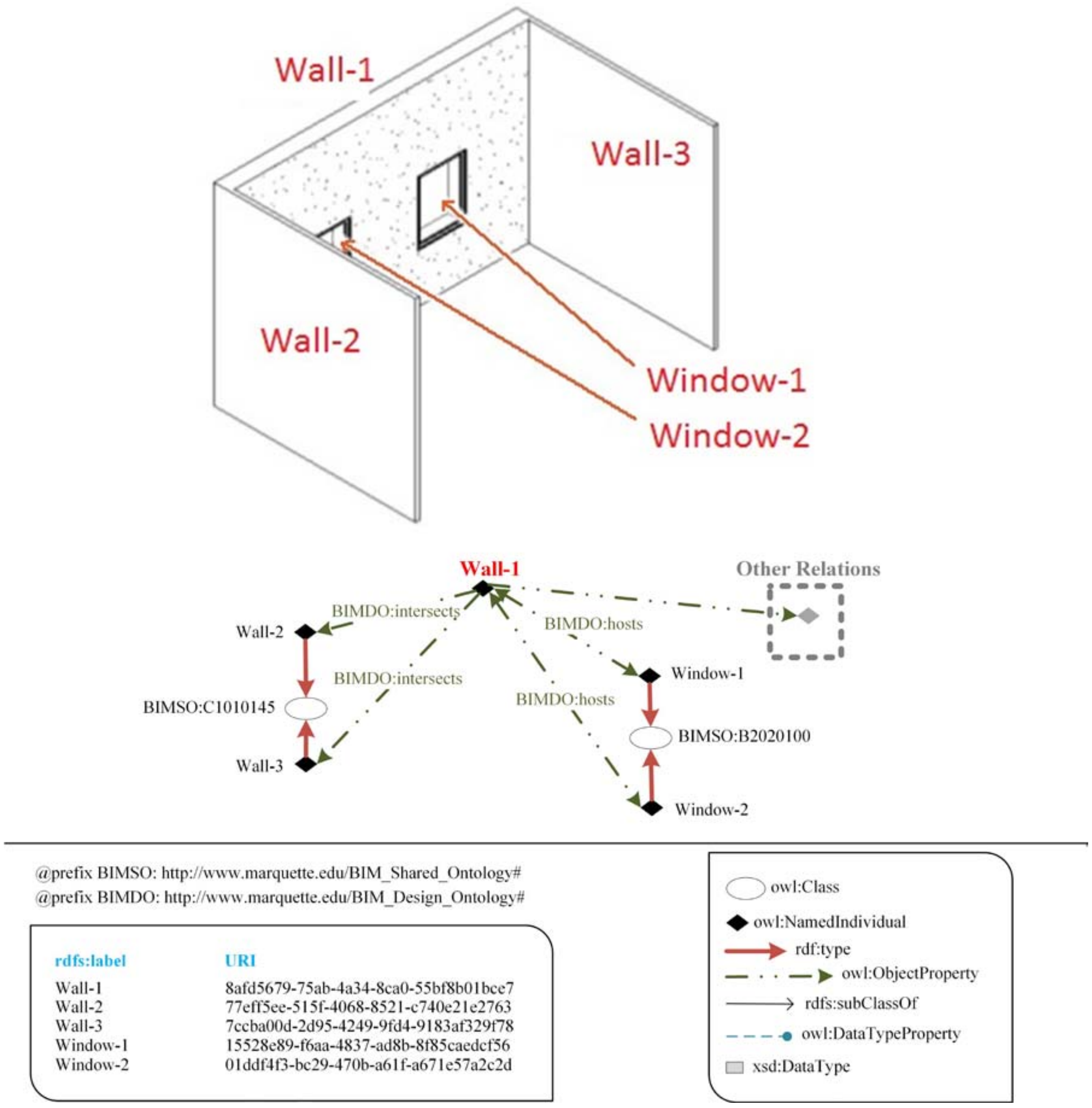


@prefix BIMSO: http://www.marquette.edu/BIM_Shared_Ontology#

rdfs:label	URI
Wall-1	8afd5679-75ab-4a34-8ca0-55bf8b01bce7
Floor-2	259c2c60-a07f-4f9f-937d-fbf4489eda2a
Room215-ComputerLab	fb90dbb5-35a7-4645-b603-e795be9cc904

Figure A-1: Wall-1 relations to floors and rooms

A BIM knowledge base also includes host and intersect relations among building elements. Figure A-2 shows that Wall-1 hosts Window-1 and Window-2 and intersects Wall-2 and Wall-3.



@prefix BIMSO: http://www.marquette.edu/BIM_Shared_Ontology#
@prefix BIMDO: http://www.marquette.edu/BIM_Design_Ontology#

rdfs:label	URI
Wall-1	8afd5679-75ab-4a34-8ca0-55bf8b01bce7
Wall-2	77eff5ee-515f-4068-8521-c740e21e2763
Wall-3	7ccba00d-2d95-4249-9fd4-9183af329f78
Window-1	15528e89-f6aa-4837-ad8b-8f85caedcf56
Window-2	01ddf4f3-bc29-470b-a61f-a671e57a2c2d

Figure A-2: Wall-1 relations with other building elements

APPENDIX B
RDF/XML REPRESENTATION OF THE FOOTING EXAMPLE GIVEN IN CHAPTER 3

```
<?xml version="1.0"?>


<!DOCTYPE rdf:RDF [
   <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
   <!ENTITY abc "http://www.ABC_DesignCompany.com#" >
   <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
   <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
   <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
   <!ENTITY BIMSO "http://www.marquette.edu/BIM_Shared_Ontology#" >
   <!ENTITY BIMDO "http://www.marquette.edu/BIM_Design_Ontology#" >
   <!ENTITY FE
"http://www.semanticweb.org/0521niknamm/ontologies/2015/7/FootingExample" >
]>



<rdf:RDF xmlns="&FE;#"
    xml:base="http://www.semanticweb.org/0521niknamm/ontologies/2015/7/FootingExample"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:abc="http://www.ABC_DesignCompany.com#"
    xmlns:BIMDO="http://www.marquette.edu/BIM_Design_Ontology#"
    xmlns:FE="http://www.semanticweb.org/0521niknamm/ontologies/2015/7/FootingExample"
    xmlns:BIMSO="http://www.marquette.edu/BIM_Shared_Ontology#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
   <owl:Ontology
rdf:about="http://www.semanticweb.org/0521niknamm/ontologies/2015/7/FootingExample"/>
```

```
<!--
///////////////////////////////////////////////////////////////////////////////////////
//
// Object Properties
//
///////////////////////////////////////////////////////////////////////////////////////
 -->


<!-- http://www.marquette.edu/BIM_Design_Ontology#hasCompressionStrength -->

<owl:ObjectProperty rdf:about="&BIMDO;hasCompressionStrength"/>



<!-- http://www.marquette.edu/BIM_Design_Ontology#hasDimensionUnit -->

<owl:ObjectProperty rdf:about="&BIMDO;hasDimensionUnit"/>



<!-- http://www.marquette.edu/BIM_Design_Ontology#hasMQUnit -->

<owl:ObjectProperty rdf:about="&BIMDO;hasMQUnit"/>



<!-- http://www.marquette.edu/BIM_Design_Ontology#hasMaterial -->

<owl:ObjectProperty rdf:about="&BIMDO;hasMaterial"/>



<!-- http://www.marquette.edu/BIM_Design_Ontology#hasYoung'sModulus -->
```

```
<owl:ObjectProperty rdf:about="&BIMDO;hasYoung’sModulus"/>



<!-- http://www.marquette.edu/BIM_Shared_Ontology#hasBuildingElement -->

<owl:ObjectProperty rdf:about="&BIMSO;hasBuildingElement"/>



<!-- http://www.marquette.edu/BIM_Shared_Ontology#hasBuildingLevel -->

<owl:ObjectProperty rdf:about="&BIMSO;hasBuildingLevel"/>



<!-- http://www.marquette.edu/BIM_Shared_Ontology#hasBuildingPhase -->

<owl:ObjectProperty rdf:about="&BIMSO;hasBuildingPhase"/>



<!-- http://www.marquette.edu/BIM_Shared_Ontology#hasElementLevel -->

<owl:ObjectProperty rdf:about="&BIMSO;hasElementLevel"/>



<!-- http://www.marquette.edu/BIM_Shared_Ontology#hasElementPhase -->

<owl:ObjectProperty rdf:about="&BIMSO;hasElementPhase"/>



<!--
///////////////////////////////////////////////////////////////////////////////////////
```

```
//
// Data properties
//
///////////////////////////////////////////////////////////////////////////////////////
 -->


<!-- http://www.marquette.edu/BIM_Design_Ontology#hasID -->

<owl:DatatypeProperty rdf:about="&BIMDO;hasID"/>



<!-- http://www.marquette.edu/BIM_Design_Ontology#hasValue -->

<owl:DatatypeProperty rdf:about="&BIMDO;hasValue"/>



<!--
///////////////////////////////////////////////////////////////////////////////////////
//
// Classes
//
///////////////////////////////////////////////////////////////////////////////////////
 -->



<!-- http://qudt.org/schema/qudt#LengthUnit -->

<owl:Class rdf:about="http://qudt.org/schema/qudt#LengthUnit"/>
```

```
<!-- http://qudt.org/schema/qudt#PressureOrStressUnit -->

<owl:Class rdf:about="http://qudt.org/schema/qudt#PressureOrStressUnit"/>



<!-- http://www.freeclass.eu/freeclass_v1#C_A140-gen -->

<owl:Class rdf:about="http://www.freeclass.eu/freeclass_v1#C_A140-gen"/>



<!-- http://www.marquette.edu/BIM_Design_Ontology#MaterialQuantitativeProperty -->

<owl:Class rdf:about="&BIMDO;MaterialQuantitativeProperty"/>



<!-- http://www.marquette.edu/BIM_Design_Ontology#Size -->

<owl:Class rdf:about="&BIMDO;Size"/>



<!-- http://www.marquette.edu/BIM_Shared_Ontology#A1010210_SpreadFooting -->

<owl:Class rdf:about="&BIMSO;A1010210_SpreadFooting"/>



<!-- http://www.marquette.edu/BIM_Shared_Ontology#Building -->

<owl:Class rdf:about="&BIMSO;Building"/>
```

```
<!-- http://www.marquette.edu/BIM_Shared_Ontology#Level -->

<owl:Class rdf:about="&BIMSO;Level"/>



<!-- http://www.marquette.edu/BIM_Shared_Ontology#Phase -->

<owl:Class rdf:about="&BIMSO;Phase"/>

<!--
///////////////////////////////////////////////////////////////////////////////////////
//
// Individuals
//
///////////////////////////////////////////////////////////////////////////////////////
 -->

<!-- http://qudt.org/vocab/unit#KilogramForcePerSquareCentimeter -->

<owl:NamedIndividual
rdf:about="http://qudt.org/vocab/unit#KilogramForcePerSquareCentimeter">
    <rdf:type rdf:resource="http://qudt.org/schema/qudt#PressureOrStressUnit"/>
</owl:NamedIndividual>



<!-- http://qudt.org/vocab/unit#Meter -->

<owl:NamedIndividual rdf:about="http://qudt.org/vocab/unit#Meter">
    <rdf:type rdf:resource="http://qudt.org/schema/qudt#LengthUnit"/>
</owl:NamedIndividual>
```

```
    <!-- http://www.ABC_DesignCompany.com#EngineeringHall -->

    <owl:NamedIndividual rdf:about="&abc;EngineeringHall">
        <rdf:type rdf:resource="&BIMSO;Building"/>
        <rdfs:label>EngineeringHall</rdfs:label>
        <BIMSO:hasBuildingPhase rdf:resource="&owl;5e75c73a-06b0-4d4a-bdd7-
aa1b531b2f46"/>
        <BIMSO:hasBuildingLevel rdf:resource="&owl;86dd5a37-604f-464c-8ffb-
a775ccc201b8"/>
        <BIMSO:hasBuildingElement rdf:resource="&owl;e473e652-35d9-4e71-834b-
bc988c0c29ec"/>
    </owl:NamedIndividual>



    <!-- http://www.w3.org/2002/07/owl#5e75c73a-06b0-4d4a-bdd7-aa1b531b2f46 -->

    <owl:NamedIndividual rdf:about="&owl;5e75c73a-06b0-4d4a-bdd7-aa1b531b2f46">
        <rdf:type rdf:resource="&BIMSO;Phase"/>
        <rdfs:label>Phase-1</rdfs:label>
    </owl:NamedIndividual>



    <!-- http://www.w3.org/2002/07/owl#86dd5a37-604f-464c-8ffb-a775ccc201b8 -->

    <owl:NamedIndividual rdf:about="&owl;86dd5a37-604f-464c-8ffb-a775ccc201b8">
        <rdf:type rdf:resource="&BIMSO;Level"/>
        <rdfs:label>FoundationLevel</rdfs:label>
    </owl:NamedIndividual>
    <!-- http://www.w3.org/2002/07/owl#e473e652-35d9-4e71-834b-bc988c0c29ec -->

    <owl:NamedIndividual rdf:about="&owl;e473e652-35d9-4e71-834b-bc988c0c29ec">
        <rdf:type rdf:resource="&BIMSO;A1010210_SpreadFooting"/>
        <rdfs:label>Footing-1</rdfs:label>
        <BIMDO:hasID>187272</BIMDO:hasID>
        <BIMSO:hasElementPhase rdf:resource="&owl;5e75c73a-06b0-4d4a-bdd7-
aa1b531b2f46"/>
        <BIMSO:hasElementLevel rdf:resource="&owl;86dd5a37-604f-464c-8ffb-
a775ccc201b8"/>
        <BIMDO:hasMaterial rdf:resource="&owl;e473e652-35d9-4e71-834b-
bc988c0c29ec_Concrete"/>
    </owl:NamedIndividual>



    <!-- http://www.w3.org/2002/07/owl#e473e652-35d9-4e71-834b-bc988c0c29ec_Concrete -->

    <owl:NamedIndividual rdf:about="&owl;e473e652-35d9-4e71-834b-
bc988c0c29ec_Concrete">
        <rdf:type rdf:resource="http://www.freeclass.eu/freeclass_v1#C_A140-gen"/>
        <rdfs:label>Footing-1-Concrete</rdfs:label>
        <BIMDO:hasCompressionStrength rdf:resource="&owl;e473e652-35d9-4e71-834b-
bc988c0c29ec_Concrete_CompresionStrength"/>
        <hasYoung:sModulus rdf:resource="&owl;e473e652-35d9-4e71-834b-
bc988c0c29ec_Concrete_Young’sModulus"/>
    </owl:NamedIndividual>



    <!-- http://www.w3.org/2002/07/owl#e473e652-35d9-4e71-834b-
bc988c0c29ec_Concrete_CompresionStrength -->
```

```
    <owl:NamedIndividual rdf:about="&owl;e473e652-35d9-4e71-834b-
bc988c0c29ec_Concrete_CompresionStrength">
        <rdf:type rdf:resource="&BIMDO;MaterialQuantitativeProperty"/>
        <rdfs:label>Footing-1-Concrete-CompresionStrength</rdfs:label>
        <BIMDO:hasValue rdf:datatype="&xsd;double">250.0</BIMDO:hasValue>
        <BIMDO:hasMQUnit
rdf:resource="http://qudt.org/vocab/unit#KilogramForcePerSquareCentimeter"/>
    </owl:NamedIndividual>



    <!-- http://www.w3.org/2002/07/owl#e473e652-35d9-4e71-834b-
bc988c0c29ec_Concrete_Young'sModulus -->


    <owl:NamedIndividual rdf:about="&owl;e473e652-35d9-4e71-834b-
bc988c0c29ec_Concrete_Young'sModulus">
        <rdf:type rdf:resource="&BIMDO;MaterialQuantitativeProperty"/>
        <rdfs:label>Footing-1-Concrete-Young'sModulus</rdfs:label>
        <rdfs:label>ready-mix concrete</rdfs:label>
        <BIMDO:hasValue rdf:datatype="&xsd;double">237000.0</BIMDO:hasValue>
        <BIMDO:hasMQUnit
rdf:resource="http://qudt.org/vocab/unit#KilogramForcePerSquareCentimeter"/>
    </owl:NamedIndividual>



    <!-- http://www.w3.org/2002/07/owl#e473e652-35d9-4e71-834b-bc988c0c29ec_Length -->


    <owl:NamedIndividual rdf:about="&owl;e473e652-35d9-4e71-834b-bc988c0c29ec_Length">
        <rdf:type rdf:resource="&BIMDO;Size"/>
        <rdfs:label>Footing-1-Length</rdfs:label>
        <BIMDO:hasValue rdf:datatype="&xsd;double">2.0</BIMDO:hasValue>
        <BIMDO:hasDimensionUnit rdf:resource="http://qudt.org/vocab/unit#Meter"/>
    </owl:NamedIndividual>
    <!-- http://www.w3.org/2002/07/owl#e473e652-35d9-4e71-834b-bc988c0c29ec_Thickness -->

    <owl:NamedIndividual rdf:about="&owl;e473e652-35d9-4e71-834b-
bc988c0c29ec_Thickness">
        <rdf:type rdf:resource="&BIMDO;Size"/>
        <rdfs:label>Footing-1-Thickness</rdfs:label>
        <BIMDO:hasValue rdf:datatype="&xsd;double">1.5</BIMDO:hasValue>
        <BIMDO:hasDimensionUnit rdf:resource="http://qudt.org/vocab/unit#Meter"/>
    </owl:NamedIndividual>



    <!-- http://www.w3.org/2002/07/owl#e473e652-35d9-4e71-834b-bc988c0c29ec_Width -->

    <owl:NamedIndividual rdf:about="&owl;e473e652-35d9-4e71-834b-bc988c0c29ec_Width">
        <rdf:type rdf:resource="&BIMDO;Size"/>
        <rdfs:label>Footing-1-Width</rdfs:label>
        <BIMDO:hasValue rdf:datatype="&xsd;double">2.0</BIMDO:hasValue>
        <BIMDO:hasDimensionUnit rdf:resource="http://qudt.org/vocab/unit#Meter"/>
    </owl:NamedIndividual>
</rdf:RDF>
```

APPENDIX C
RDF/XML REPRESENTATION OF THE ASSEMBLY AND WORK ITEM EXAMPLES GIVEN IN CHAPTER 4

```
<?xml version="1.0"?>


<!DOCTYPE rdf:RDF [
    <!ENTITY org "http://www.w3.org/ns/org#" >
    <!ENTITY foaf "http://xmlns.com/foaf/0.1/" >
    <!ENTITY qudt "http://qudt.org/vocab/unit#" >
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY gr "http://purl.org/goodrelations/v1#" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY fc "http://www.freeclass.eu/freeclass_v1#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY xyz "http://www.XYZ_ConstructionCompany.com/#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
    <!ENTITY rst "http://www.RST_MaterialSupplierCompany.com/#" >
    <!ENTITY mueo "http://www.marquette.edu/estimating_ontology#" >
]>


<rdf:RDF xmlns="http://www.semanticweb.org/0521niknamm/ontologies/2015/7/untitled-
ontology-47#"
     xml:base="http://www.semanticweb.org/0521niknamm/ontologies/2015/7/untitled-ontology-
47"
     xmlns:foaf="http://xmlns.com/foaf/0.1/"
     xmlns:org="http://www.w3.org/ns/org#"
     xmlns:qudt="http://qudt.org/vocab/unit#"
     xmlns:fc="http://www.freeclass.eu/freeclass_v1#"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
```

```
 xmlns:mueo="http://www.marquette.edu/estimating_ontology#"

 xmlns:xsd="http://www.w3.org/2001/XMLSchema#"

 xmlns:owl="http://www.w3.org/2002/07/owl#"

 xmlns:rst="http://www.RST_MaterialSupplierCompany.com/#"

 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

 xmlns:gr="http://purl.org/goodrelations/v1#"

 xmlns:xyz="http://www.XYZ_ConstructionCompany.com/#">

<owl:Ontology
rdf:about="http://www.semanticweb.org/0521niknamm/ontologies/2015/7/untitled-ontology-
47"/>



<!--

///////////////////////////////////////////////////////////////////////////////////////

//

// Object Properties

//

///////////////////////////////////////////////////////////////////////////////////////

 -->


<!-- http://purl.org/goodrelations/v1#qualitativeProductOrServiceProperty -->


<owl:ObjectProperty rdf:about="&gr;qualitativeProductOrServiceProperty"/>



<!-- http://www.freeclass.eu/freeclass_v1#P_16 -->


<owl:ObjectProperty rdf:about="&fc;P_16"/>



<!-- http://www.freeclass.eu/freeclass_v1#P_E26 -->
```

```
<owl:ObjectProperty rdf:about="&fc;P_E26"/>



<!-- http://www.freeclass.eu/freeclass_v1#P_E27 -->

<owl:ObjectProperty rdf:about="&fc;P_E27"/>



<!-- http://www.freeclass.eu/freeclass_v1#P_E28 -->

<owl:ObjectProperty rdf:about="&fc;P_E28"/>



<!-- http://www.freeclass.eu/freeclass_v1#P_E29 -->

<owl:ObjectProperty rdf:about="&fc;P_E29"/>



<!-- http://www.freeclass.eu/freeclass_v1#P_E31 -->

<owl:ObjectProperty rdf:about="&fc;P_E31"/>



<!-- http://www.freeclass.eu/freeclass_v1#P_E32 -->

<owl:ObjectProperty rdf:about="&fc;P_E32"/>



<!-- http://www.freeclass.eu/freeclass_v1#P_E52 -->
```

<owl:ObjectProperty rdf:about="&fc;P_E52"/>

<!-- http://www.marquette.edu/estimating_ontology#classification -->

<owl:ObjectProperty rdf:about="&mueo;classification"/>

<!-- http://www.marquette.edu/estimating_ontology#hasCost -->

<owl:ObjectProperty rdf:about="&mueo;hasCost"/>

<!-- http://www.marquette.edu/estimating_ontology#hasLabor -->

<owl:ObjectProperty rdf:about="&mueo;hasLabor"/>

<!-- http://www.marquette.edu/estimating_ontology#hasProductivity -->

<owl:ObjectProperty rdf:about="&mueo;hasProductivity"/>

<!-- http://www.marquette.edu/estimating_ontology#hasQuantity -->

<owl:ObjectProperty rdf:about="&mueo;hasQuantity"/>

<!-- http://www.marquette.edu/estimating_ontology#hasUnit -->

```
<owl:ObjectProperty rdf:about="&mueo;hasUnit"/>



<!-- http://www.marquette.edu/estimating_ontology#partOf -->

<owl:ObjectProperty rdf:about="&mueo;partOf"/>



<!-- http://www.marquette.edu/estimating_ontology#performedBy -->

<owl:ObjectProperty rdf:about="&mueo;performedBy"/>



<!-- http://www.marquette.edu/estimating_ontology#specialty -->

<owl:ObjectProperty rdf:about="&mueo;specialty"/>



<!-- http://www.marquette.edu/estimating_ontology#uses -->

<owl:ObjectProperty rdf:about="&mueo;uses"/>



<!--
///////////////////////////////////////////////////////////////////////////////////////
//
// Data properties
//
///////////////////////////////////////////////////////////////////////////////////////
```

```
-->


<!-- http://www.marquette.edu/estimating_ontology#cost($)/hr -->

<owl:DatatypeProperty rdf:about="&mueo;cost($)/hr"/>



<!-- http://www.marquette.edu/estimating_ontology#hasCurrency -->

<owl:DatatypeProperty rdf:about="&mueo;hasCurrency"/>



<!-- http://www.marquette.edu/estimating_ontology#hasID -->

<owl:DatatypeProperty rdf:about="&mueo;hasID"/>



<!-- http://www.marquette.edu/estimating_ontology#hasSpecialCondition -->

<owl:DatatypeProperty rdf:about="&mueo;hasSpecialCondition"/>



<!-- http://www.marquette.edu/estimating_ontology#hasTitle -->

<owl:DatatypeProperty rdf:about="&mueo;hasTitle"/>



<!-- http://www.marquette.edu/estimating_ontology#hasValue -->

<owl:DatatypeProperty rdf:about="&mueo;hasValue"/>
```

```
<!-- http://www.marquette.edu/estimating_ontology#livesIn -->

<owl:DatatypeProperty rdf:about="&mueo;livesIn"/>



<!-- http://www.marquette.edu/estimating_ontology#quantity -->

<owl:DatatypeProperty rdf:about="&mueo;quantity"/>



<!--
///////////////////////////////////////////////////////////////////////////////////////
//
// Classes
//
///////////////////////////////////////////////////////////////////////////////////////
 -->

<!-- http://purl.org/goodrelations/v1#QualitativeValue -->

<owl:Class rdf:about="&gr;QualitativeValue"/>



<!-- http://purl.org/goodrelations/v1#QuantitativeValueFloat -->

<owl:Class rdf:about="&gr;QuantitativeValueFloat"/>
```

```
<!-- http://www.freeclass.eu/freeclass_v1#C_A140-gen -->

<owl:Class rdf:about="&fc;C_A140-gen"/>



<!-- http://www.marquette.edu/estimating_ontology#A1010210_SpreadFooting -->

<owl:Class rdf:about="&mueo;A1010210_SpreadFooting"/>



<!-- http://www.marquette.edu/estimating_ontology#AssemblyCost -->

<owl:Class rdf:about="&mueo;AssemblyCost"/>



<!-- http://www.marquette.edu/estimating_ontology#Crew -->

<owl:Class rdf:about="&mueo;Crew"/>



<!-- http://www.marquette.edu/estimating_ontology#D03111345 -->

<owl:Class rdf:about="&mueo;D03111345"/>



<!-- http://www.marquette.edu/estimating_ontology#D03211160 -->

<owl:Class rdf:about="&mueo;D03211160"/>
```

```
<!-- http://www.marquette.edu/estimating_ontology#D03311370 -->

<owl:Class rdf:about="&mueo;D03311370"/>



<!-- http://www.marquette.edu/estimating_ontology#Equipment -->

<owl:Class rdf:about="&mueo;Equipment"/>



<!-- http://www.marquette.edu/estimating_ontology#ItemCost -->

<owl:Class rdf:about="&mueo;ItemCost"/>



<!-- http://www.marquette.edu/estimating_ontology#ItemProductivity -->

<owl:Class rdf:about="&mueo;ItemProductivity"/>



<!-- http://www.marquette.edu/estimating_ontology#ItemQuantity -->

<owl:Class rdf:about="&mueo;ItemQuantity"/>



<!-- http://www.w3.org/ns/org#Organization -->

<owl:Class rdf:about="&org;Organization"/>
```

```
<!-- http://xmlns.com/foaf/0.1/Person -->

<owl:Class rdf:about="&foaf;Person"/>



<!--
///////////////////////////////////////////////////////////////////////////////////////
//
// Individuals
//
///////////////////////////////////////////////////////////////////////////////////////
 -->

<!-- http://qudt.org/vocab/unit#CubicMeter -->

<owl:NamedIndividual rdf:about="&qudt;CubicMeter"/>



<!-- http://qudt.org/vocab/unit#CubicMeterPerHour -->

<owl:NamedIndividual rdf:about="&qudt;CubicMeterPerHour"/>



<!-- http://www.RST_MaterialSupplierCompany.com/#ReadyMixConcrete_3256 -->

<owl:NamedIndividual rdf:about="&rst;ReadyMixConcrete_3256">
    <rdf:type rdf:resource="&fc;C_A140-gen"/>
    <fc:P_16 rdf:resource="&fc;2kN/cm2"/>
    <fc:P_E26 rdf:resource="&fc;S2_50-90_mm"/>
```

```
        <fc:P_E31 rdf:resource="&fc;V_W101"/>
        <fc:P_E32 rdf:resource="&fc;V_W106"/>
        <fc:P_E52 rdf:resource="&fc;V_W167"/>
        <fc:P_E27 rdf:resource="&fc;V_W87"/>
        <fc:P_E28 rdf:resource="&fc;V_W91"/>
        <fc:P_E29 rdf:resource="&fc;V_W95"/>
    </owl:NamedIndividual>



    <!-- http://www.XYZ_ConstructionCompany.com/#A1010210_SpreadFooting_7261 -->

    <owl:NamedIndividual rdf:about="&xyz;A1010210_SpreadFooting_7261">
        <rdf:type rdf:resource="&mueo;A1010210_SpreadFooting"/>
        <mueo:hasTitle>SpreadFooting_ConcreteStrength2kN/cm2</mueo:hasTitle>
        <mueo:hasID>A1010210_7261</mueo:hasID>
        <mueo:hasSpecialCondition>ColdWeather</mueo:hasSpecialCondition>
    </owl:NamedIndividual>



    <!-- http://www.XYZ_ConstructionCompany.com/#A1010210_SpreadFooting_7261_Cost -->

    <owl:NamedIndividual rdf:about="&xyz;A1010210_SpreadFooting_7261_Cost">
        <rdf:type rdf:resource="&mueo;AssemblyCost"/>
    </owl:NamedIndividual>



    <!-- http://www.XYZ_ConstructionCompany.com/#D03111345_FormsInPlaceFooting_4152 -
->

    <owl:NamedIndividual rdf:about="&xyz;D03111345_FormsInPlaceFooting_4152">
```

```
        <rdf:type rdf:resource="&mueo;D03111345"/>
        <mueo:partOf rdf:resource="&xyz;A1010210_SpreadFooting_7261"/>
    </owl:NamedIndividual>



    <!-- http://www.XYZ_ConstructionCompany.com/#D03211160_ReinforcingInPlace_0151 -->

    <owl:NamedIndividual rdf:about="&xyz;D03211160_ReinforcingInPlace_0151">
        <rdf:type rdf:resource="&mueo;D03211160"/>
        <mueo:partOf rdf:resource="&xyz;A1010210_SpreadFooting_7261"/>
    </owl:NamedIndividual>



    <!-- http://www.XYZ_ConstructionCompany.com/#D03311370_PlacingConcrete_2457 -->

    <owl:NamedIndividual rdf:about="&xyz;D03311370_PlacingConcrete_2457">
        <rdf:type rdf:resource="&mueo;D03311370"/>
        <mueo:hasID>D03311370_2457</mueo:hasID>

<mueo:hasTitle>PlacingConcrete_SpreadFooting_LessThan3CubicMeter_Pump</mueo:hasTitle
>
        <mueo:uses rdf:resource="&rst;ReadyMixConcrete_3256"/>
        <mueo:partOf rdf:resource="&xyz;A1010210_SpreadFooting_7261"/>
        <mueo:hasCost rdf:resource="&xyz;D03311370_PlacingConcrete_2457_Cost"/>
        <mueo:uses rdf:resource="&xyz;D03311370_PlacingConcrete_2457_Crew"/>
        <mueo:hasProductivity
rdf:resource="&xyz;D03311370_PlacingConcrete_2457_Productivity"/>
        <mueo:hasQuantity rdf:resource="&xyz;D03311370_PlacingConcrete_2457_Quantity"/>
        <mueo:performedBy rdf:resource="&xyz;XYZ_ConstructionCompany"/>
    </owl:NamedIndividual>
    <!-- http://www.XYZ_ConstructionCompany.com/#D03311370_PlacingConcrete_2457_Cost -
->

    <owl:NamedIndividual rdf:about="&xyz;D03311370_PlacingConcrete_2457_Cost">
        <rdf:type rdf:resource="&mueo;ItemCost"/>
        <mueo:hasValue rdf:datatype="&xsd;double">1258.5</mueo:hasValue>
        <mueo:hasCurrency rdf:datatype="&xsd;double">USD</mueo:hasCurrency>
    </owl:NamedIndividual>



    <!-- http://www.XYZ_ConstructionCompany.com/#D03311370_PlacingConcrete_2457_Crew
-->

    <owl:NamedIndividual rdf:about="&xyz;D03311370_PlacingConcrete_2457_Crew">
        <rdf:type rdf:resource="&mueo;Crew"/>
        <gr:qualitativeProductOrServiceProperty rdf:resource="&xyz;Equipment_1289"/>
        <gr:qualitativeProductOrServiceProperty rdf:resource="&xyz;Equipment_4376"/>
        <mueo:hasLabor rdf:resource="&xyz;Labor_4223"/>
        <mueo:hasLabor rdf:resource="&xyz;Labor_6460"/>
    </owl:NamedIndividual>



    <!--
http://www.XYZ_ConstructionCompany.com/#D03311370_PlacingConcrete_2457_Productivity
-->

    <owl:NamedIndividual rdf:about="&xyz;D03311370_PlacingConcrete_2457_Productivity">
        <rdf:type rdf:resource="&mueo;ItemProductivity"/>
        <mueo:hasValue rdf:datatype="&xsd;double">5.4</mueo:hasValue>
        <mueo:hasUnit rdf:resource="&qudt;CubicMeterPerHour"/>
    </owl:NamedIndividual>



    <!--
http://www.XYZ_ConstructionCompany.com/#D03311370_PlacingConcrete_2457_Quantity -->

    <owl:NamedIndividual rdf:about="&xyz;D03311370_PlacingConcrete_2457_Quantity">
        <rdf:type rdf:resource="&mueo;ItemQuantity"/>
        <mueo:hasValue rdf:datatype="&xsd;double">6.0</mueo:hasValue>
        <mueo:hasUnit rdf:resource="&qudt;CubicMeter"/>
    </owl:NamedIndividual>



    <!-- http://www.XYZ_ConstructionCompany.com/#Equipment_1289 -->

    <owl:NamedIndividual rdf:about="&xyz;Equipment_1289">
        <rdf:type rdf:resource="&mueo;Equipment"/>
        <mueo:quantity rdf:datatype="&xsd;double">1.0</mueo:quantity>
        <cost:hr rdf:datatype="&xsd;double">105.0</cost:hr>
        <mueo:classification rdf:resource="&mueo;ConcretePump"/>
    </owl:NamedIndividual>



    <!-- http://www.XYZ_ConstructionCompany.com/#Equipment_4376 -->

    <owl:NamedIndividual rdf:about="&xyz;Equipment_4376">
        <rdf:type rdf:resource="&mueo;Equipment"/>
        <mueo:quantity rdf:datatype="&xsd;double">1.0</mueo:quantity>
        <cost:hr rdf:datatype="&xsd;double">8.0</cost:hr>
        <mueo:classification rdf:resource="&mueo;GasEngineVibrator"/>
```

```
</owl:NamedIndividual>



<!-- http://www.XYZ_ConstructionCompany.com/#Labor_4223 -->

<owl:NamedIndividual rdf:about="&xyz;Labor_4223">
   <rdf:type rdf:resource="&foaf;Person"/>
   <mueo:livesIn rdf:datatype="&xsd;double">1.0</mueo:livesIn>
   <mueo:quantity rdf:datatype="&xsd;double">1.0</mueo:quantity>
   <mueo:livesIn rdf:datatype="&xsd;double">35.0</mueo:livesIn>
   <mueo:livesIn rdf:datatype="&xsd;double">Milwaukee,WI</mueo:livesIn>
   <mueo:specialty rdf:resource="&mueo;CementFinisher"/>
   <mueo:classification rdf:resource="&mueo;SkilledLabor"/>
</owl:NamedIndividual>



<!-- http://www.XYZ_ConstructionCompany.com/#Labor_6460 -->

<owl:NamedIndividual rdf:about="&xyz;Labor_6460">
   <rdf:type rdf:resource="&foaf;Person"/>
   <mueo:quantity rdf:datatype="&xsd;double">2.0</mueo:quantity>
   <cost:hr rdf:datatype="&xsd;double">28.0</cost:hr>
   <mueo:livesIn rdf:datatype="&xsd;double">Milwaukee,WI</mueo:livesIn>
   <mueo:specialty rdf:resource="&mueo;Laborer"/>
   <mueo:classification rdf:resource="&mueo;UnSkilledLabor"/>
</owl:NamedIndividual>



<!-- http://www.XYZ_ConstructionCompany.com/#XYZ_ConstructionCompany -->
```

```
<owl:NamedIndividual rdf:about="&xyz;XYZ_ConstructionCompany">
    <rdf:type rdf:resource="&org;Organization"/>
</owl:NamedIndividual>



<!-- http://www.freeclass.eu/freeclass_v1#2kN/cm2 -->

<owl:NamedIndividual rdf:about="&fc;2kN/cm2">
    <rdf:type rdf:resource="&gr;QuantitativeValueFloat"/>
</owl:NamedIndividual>



<!-- http://www.freeclass.eu/freeclass_v1#S2_50-90_mm -->

<owl:NamedIndividual rdf:about="&fc;S2_50-90_mm">
    <rdf:type rdf:resource="&gr;QuantitativeValueFloat"/>
</owl:NamedIndividual>



<!-- http://www.freeclass.eu/freeclass_v1#V_W101 -->

<owl:NamedIndividual rdf:about="&fc;V_W101">
    <rdf:type rdf:resource="&gr;QualitativeValue"/>
    <rdfs:label xml:lang="en">XC1 constantly wet or dry</rdfs:label>
</owl:NamedIndividual>



<!-- http://www.freeclass.eu/freeclass_v1#V_W106 -->

<owl:NamedIndividual rdf:about="&fc;V_W106">
```

```
    <rdf:type rdf:resource="&gr;QualitativeValue"/>
    <rdfs:label xml:lang="en">XD2 wet, seldom dry</rdfs:label>
</owl:NamedIndividual>



<!-- http://www.freeclass.eu/freeclass_v1#V_W167 -->

<owl:NamedIndividual rdf:about="&fc;V_W167">
    <rdf:type rdf:resource="&gr;QualitativeValue"/>
    <rdfs:label xml:lang="en">CEM I</rdfs:label>
</owl:NamedIndividual>



<!-- http://www.freeclass.eu/freeclass_v1#V_W87 -->

<owl:NamedIndividual rdf:about="&fc;V_W87">
    <rdf:type rdf:resource="&gr;QualitativeValue"/>
    <rdfs:label xml:lang="en">XA1 chemically attacking environment (weak)</rdfs:label>
</owl:NamedIndividual>



<!-- http://www.freeclass.eu/freeclass_v1#V_W91 -->

<owl:NamedIndividual rdf:about="&fc;V_W91">
    <rdf:type rdf:resource="&gr;QualitativeValue"/>
    <rdfs:label xml:lang="en">XF1 moderate water saturation without deicing</rdfs:label>
</owl:NamedIndividual>
```

```
<!-- http://www.freeclass.eu/freeclass_v1#V_W95 -->

<owl:NamedIndividual rdf:about="&fc;V_W95">
    <rdf:type rdf:resource="&gr;QualitativeValue"/>
    <rdfs:label xml:lang="en">XM1 moderate wear stress</rdfs:label>
</owl:NamedIndividual>



<!-- http://www.marquette.edu/estimating_ontology#CementFinisher -->

<owl:NamedIndividual rdf:about="&mueo;CementFinisher"/>



<!-- http://www.marquette.edu/estimating_ontology#ConcretePump -->

<owl:NamedIndividual rdf:about="&mueo;ConcretePump"/>



<!-- http://www.marquette.edu/estimating_ontology#GasEngineVibrator -->

<owl:NamedIndividual rdf:about="&mueo;GasEngineVibrator"/>



<!-- http://www.marquette.edu/estimating_ontology#Laborer -->

<owl:NamedIndividual rdf:about="&mueo;Laborer"/>



<!-- http://www.marquette.edu/estimating_ontology#SkilledLabor -->
```

```
    <owl:NamedIndividual rdf:about="&mueo;SkilledLabor"/>



    <!-- http://www.marquette.edu/estimating_ontology#UnSkilledLabor -->

    <owl:NamedIndividual rdf:about="&mueo;UnSkilledLabor"/>
</rdf:RDF>
```

APPENDIX D
RDF/XML REPRESENTATION OF THE MATERIAL SUPPLIER EXAMPLE GIVEN IN CHAPTER 5

```
<?xml version="1.0"?>


<!DOCTYPE rdf:RDF [
    <!ENTITY qudt "http://qudt.org/vocab/unit#" >
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY gr "http://purl.org/goodrelations/v1#" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY fc "http://www.freeclass.eu/freeclass_v1#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
    <!ENTITY rst "http://www.RST_MaterialSupplierCompany.com/#" >
]>


<rdf:RDF
xmlns="http://www.semanticweb.org/0521niknamm/ontologies/2015/7/untitled-
ontology-51#"
     xml:base="http://www.semanticweb.org/0521niknamm/ontologies/2015/7/untitled-
ontology-51"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:rst="http://www.RST_MaterialSupplierCompany.com/#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
```

```
 xmlns:gr="http://purl.org/goodrelations/v1#"

 xmlns:qudt="http://qudt.org/vocab/unit#"

 xmlns:fc="http://www.freeclass.eu/freeclass_v1#">

<owl:Ontology
rdf:about="http://www.semanticweb.org/0521niknamm/ontologies/2015/7/untitled-
ontology-51"/>



<!--

///////////////////////////////////////////////////////////////////////////////////////

//

// Object Properties

//

///////////////////////////////////////////////////////////////////////////////////////

 -->



 <!-- http://purl.org/goodrelations/v1#hasPriceSpecification -->


<owl:ObjectProperty rdf:about="&gr;hasPriceSpecification">

    <rdfs:label xml:lang="en">has price specification (0..*)</rdfs:label>

    <rdfs:range rdf:resource="&gr;PriceSpecification"/>

    <rdfs:domain>

        <owl:Class>

            <owl:unionOf rdf:parseType="Collection">

                <rdf:Description rdf:about="&gr;Offering"/>

                <rdf:Description rdf:about="http://schema.org/Offer"/>

            </owl:unionOf>

        </owl:Class>

    </rdfs:domain>
```

```
</owl:ObjectProperty>



<!-- http://purl.org/goodrelations/v1#hasUnitOfMeasurement -->

<owl:ObjectProperty rdf:about="&gr;hasUnitOfMeasurement"/>



<!-- http://purl.org/goodrelations/v1#includes -->

<owl:ObjectProperty rdf:about="&gr;includes">
    <rdfs:label xml:lang="en">includes (0..1)</rdfs:label>
    <rdfs:domain rdf:resource="&gr;Offering"/>
    <rdfs:range rdf:resource="&gr;ProductOrService"/>
</owl:ObjectProperty>



<!-- http://purl.org/goodrelations/v1#offers -->

<owl:ObjectProperty rdf:about="&gr;offers">
    <rdfs:label xml:lang="en">offers (0..*)</rdfs:label>
    <rdfs:range rdf:resource="&gr;Offering"/>
    <rdfs:domain>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <rdf:Description rdf:about="&gr;BusinessEntity"/>
                <rdf:Description rdf:about="http://schema.org/Organization"/>
            </owl:unionOf>
        </owl:Class>
    </rdfs:domain>
</owl:ObjectProperty>



<!-- http://www.freeclass.eu/freeclass_v1#P_16 -->

<owl:ObjectProperty rdf:about="&fc;P_16"/>

<!-- http://www.freeclass.eu/freeclass_v1#P_E26 -->

<owl:ObjectProperty rdf:about="&fc;P_E26"/>



<!-- http://www.freeclass.eu/freeclass_v1#P_E27 -->

<owl:ObjectProperty rdf:about="&fc;P_E27"/>



<!-- http://www.freeclass.eu/freeclass_v1#P_E28 -->

<owl:ObjectProperty rdf:about="&fc;P_E28"/>



<!-- http://www.freeclass.eu/freeclass_v1#P_E29 -->

<owl:ObjectProperty rdf:about="&fc;P_E29"/>
```

```
<!-- http://www.freeclass.eu/freeclass_v1#P_E31 -->

<owl:ObjectProperty rdf:about="&fc;P_E31"/>



<!-- http://www.freeclass.eu/freeclass_v1#P_E32 -->

<owl:ObjectProperty rdf:about="&fc;P_E32"/>

<!-- http://www.freeclass.eu/freeclass_v1#P_E52 -->

<owl:ObjectProperty rdf:about="&fc;P_E52"/>



<!--
///////////////////////////////////////////////////////////////////////////////////////
//
// Data properties
//
///////////////////////////////////////////////////////////////////////////////////////
 -->

<!-- http://purl.org/goodrelations/v1#hasCurrency -->

<owl:DatatypeProperty rdf:about="&gr;hasCurrency"/>
```

```
<!-- http://purl.org/goodrelations/v1#hasCurrencyValue -->

<owl:DatatypeProperty rdf:about="&gr;hasCurrencyValue"/>

<!--
///////////////////////////////////////////////////////////////////////////////////////
//
// Classes
//
///////////////////////////////////////////////////////////////////////////////////////
 -->

<!-- http://purl.org/goodrelations/v1#Brand -->

<owl:Class rdf:about="&gr;Brand">
    <owl:disjointWith rdf:resource="&gr;BusinessEntity"/>
    <owl:disjointWith rdf:resource="&gr;Offering"/>
    <owl:disjointWith rdf:resource="&gr;PriceSpecification"/>
</owl:Class>



<!-- http://purl.org/goodrelations/v1#BusinessEntity -->

<owl:Class rdf:about="&gr;BusinessEntity">
    <rdfs:label xml:lang="en">Business entity</rdfs:label>
    <owl:disjointWith rdf:resource="&gr;Offering"/>
    <owl:disjointWith rdf:resource="&gr;PriceSpecification"/>
```

```
<rdfs:isDefinedBy rdf:resource="http://purl.org/goodrelations/v1"/>
    </owl:Class>



    <!-- http://purl.org/goodrelations/v1#Offering -->

    <owl:Class rdf:about="&gr;Offering">
        <rdfs:label xml:lang="en">Offering</rdfs:label>
        <owl:disjointWith rdf:resource="&gr;PriceSpecification"/>
        <owl:disjointWith rdf:resource="&gr;ProductOrService"/>
<rdfs:isDefinedBy rdf:resource="http://purl.org/goodrelations/v1"/>
    </owl:Class>



    <!-- http://purl.org/goodrelations/v1#PriceSpecification -->

    <owl:Class rdf:about="&gr;PriceSpecification">
        <rdfs:label xml:lang="en">Price specification</rdfs:label>
        <owl:disjointWith rdf:resource="&gr;ProductOrService"/>
        <rdfs:comment xml:lang="en">The superclass of all price
specifications.</rdfs:comment>
        <rdfs:isDefinedBy rdf:resource="http://purl.org/goodrelations/v1"/>
    </owl:Class>



    <!-- http://purl.org/goodrelations/v1#ProductOrService -->

    <owl:Class rdf:about="&gr;ProductOrService"/>
    <!-- http://purl.org/goodrelations/v1#UnitPriceSpecification -->

    <owl:Class rdf:about="&gr;UnitPriceSpecification">
        <rdfs:label xml:lang="en">Unit price specification</rdfs:label>
        <rdfs:subClassOf rdf:resource="&gr;PriceSpecification"/>
<rdfs:isDefinedBy rdf:resource="http://purl.org/goodrelations/v1"/>
    </owl:Class>



    <!-- http://schema.org/Offer -->

    <owl:Class rdf:about="http://schema.org/Offer">
        <rdfs:subClassOf rdf:resource="&gr;Offering"/>
    </owl:Class>



    <!-- http://schema.org/Organization -->

    <owl:Class rdf:about="http://schema.org/Organization">
        <rdfs:subClassOf rdf:resource="&gr;BusinessEntity"/>
    </owl:Class>



    <!-- http://www.freeclass.eu/freeclass_v1#C_A140-gen -->

    <owl:Class rdf:about="&fc;C_A140-gen">
        <rdfs:label xml:lang="en">ready-mix concrete [Generic Concept: This type of
goods]</rdfs:label>
```

```
        <rdfs:subClassOf rdf:resource="&gr;ProductOrService"/>

        <rdfs:subClassOf rdf:resource="&fc;C_A140-tax"/>

        <rdfs:comment xml:lang="en">This class subsumes all actual instances of the
following type of goods and true specializations: ready-mix concrete.</rdfs:comment>

        <rdfs:isDefinedBy rdf:resource="http://www.freeclass.eu/freeclass_v1"/>

        <rdfs:seeAlso
rdf:resource="http://www.freeclass.eu/?r=078$eclf$$noframe$$$$$$$$$$$$12050505$$
A140$$find$$~/./$$eclf@@"/>

    </owl:Class>



    <!-- http://www.freeclass.eu/freeclass_v1#C_A140-tax -->


    <owl:Class rdf:about="&fc;C_A140-tax">

        <rdfs:subClassOf rdf:resource="&gr;ProductOrService"/>

    </owl:Class>



    <!--

    ///////////////////////////////////////////////////////////////////////////////////////

    //

    // Individuals

    //

    ///////////////////////////////////////////////////////////////////////////////////////

     -->


    <!-- http://qudt.org/vocab/unit#CubicMeter -->
```

```
<owl:NamedIndividual rdf:about="&qudt;CubicMeter"/>



<!-- http://www.RST_MaterialSupplierCompany.com/#ConcreteSupplier -->

<owl:NamedIndividual rdf:about="&rst;ConcreteSupplier">
    <rdf:type rdf:resource="&gr;BusinessEntity"/>
    <rdf:type rdf:resource="http://schema.org/Organization"/>
    <gr:offers rdf:resource="&rst;Offering_1823"/>
</owl:NamedIndividual>



<!-- http://www.RST_MaterialSupplierCompany.com/#Offering_1823 -->

<owl:NamedIndividual rdf:about="&rst;Offering_1823">
    <rdf:type rdf:resource="&gr;Offering"/>
    <rdf:type rdf:resource="http://schema.org/Offer"/>
    <gr:hasPriceSpecification rdf:resource="&rst;Offering_1823_PriceSpecification"/>
    <gr:includes rdf:resource="&rst;ReadyMixConcrete_3256"/>
</owl:NamedIndividual>



<!--
http://www.RST_MaterialSupplierCompany.com/#Offering_1823_PriceSpecification -->

<owl:NamedIndividual rdf:about="&rst;Offering_1823_PriceSpecification">
    <rdf:type rdf:resource="&gr;UnitPriceSpecification"/>
    <gr:hasCurrency>USD</gr:hasCurrency>
    <gr:hasCurrencyValue>171.98</gr:hasCurrencyValue>
```

```
        <gr:hasUnitOfMeasurement rdf:resource="&qudt;CubicMeter"/>
    </owl:NamedIndividual>



    <!-- http://www.RST_MaterialSupplierCompany.com/#ReadyMixConcrete_3256 -->

    <owl:NamedIndividual rdf:about="&rst;ReadyMixConcrete_3256">
        <rdf:type rdf:resource="&fc;C_A140-gen"/>
        <fc:P_16 rdf:resource="&fc;2kN/cm2"/>
        <fc:P_E26 rdf:resource="&fc;S2_50-90_mm"/>
        <fc:P_E31 rdf:resource="&fc;V_W101"/>
        <fc:P_E32 rdf:resource="&fc;V_W106"/>
        <fc:P_E52 rdf:resource="&fc;V_W167"/>
        <fc:P_E27 rdf:resource="&fc;V_W87"/>
        <fc:P_E28 rdf:resource="&fc;V_W91"/>
        <fc:P_E29 rdf:resource="&fc;V_W95"/>
    </owl:NamedIndividual>



    <!-- http://www.XYZ_ConstructionCompany.com/#XYZ_ConstructionCompany -->

    <owl:NamedIndividual
rdf:about="http://www.XYZ_ConstructionCompany.com/#XYZ_ConstructionCompany"
/>



    <!-- http://www.freeclass.eu/freeclass_v1#2kN/cm2 -->

    <owl:NamedIndividual rdf:about="&fc;2kN/cm2"/>
```

```
<!-- http://www.freeclass.eu/freeclass_v1#S2_50-90_mm -->

<owl:NamedIndividual rdf:about="&fc;S2_50-90_mm"/>



<!-- http://www.freeclass.eu/freeclass_v1#V_W101 -->

<owl:NamedIndividual rdf:about="&fc;V_W101">
    <rdfs:label xml:lang="en">XC1 constantly wet or dry</rdfs:label>
</owl:NamedIndividual>



<!-- http://www.freeclass.eu/freeclass_v1#V_W106 -->

<owl:NamedIndividual rdf:about="&fc;V_W106">
    <rdfs:label xml:lang="en">XD2 wet, seldom dry</rdfs:label>
</owl:NamedIndividual>



<!-- http://www.freeclass.eu/freeclass_v1#V_W167 -->

<owl:NamedIndividual rdf:about="&fc;V_W167">
    <rdfs:label xml:lang="en">CEM I</rdfs:label>
</owl:NamedIndividual>
```

```
    <!-- http://www.freeclass.eu/freeclass_v1#V_W87 -->


    <owl:NamedIndividual rdf:about="&fc;V_W87">
        <rdfs:label xml:lang="en">XA1 chemically attacking environment
(weak)</rdfs:label>
    </owl:NamedIndividual>



    <!-- http://www.freeclass.eu/freeclass_v1#V_W91 -->


    <owl:NamedIndividual rdf:about="&fc;V_W91">
        <rdfs:label xml:lang="en">XF1 moderate water saturation without
deicing</rdfs:label>
    </owl:NamedIndividual>



    <!-- http://www.freeclass.eu/freeclass_v1#V_W95 -->


    <owl:NamedIndividual rdf:about="&fc;V_W95">
        <rdfs:label xml:lang="en">XM1 moderate wear stress</rdfs:label>
    </owl:NamedIndividual>
</rdf:RDF>
```

APPENDIX E
RDF/XML REPRESENTATION OF THE SEMANTIC DESCRIPTION FOR A CONCRETE MATERIAL SUPPLIER SEMANTIC WEB SERVICE

```
<?xml version="1.0"?>


<!DOCTYPE rdf:RDF [
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <!ENTITY swrl "http://www.w3.org/2003/11/swrl#" >
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
  <!ENTITY service "http://www.daml.org/services/owl-s/1.2/Service.owl#" >
  <!ENTITY process "http://www.daml.org/services/owl-s/1.2/Process.owl#" >
  <!ENTITY profile "http://www.daml.org/services/owl-s/1.2/Profile.owl#" >
  <!ENTITY grounding "http://www.daml.org/services/owl-s/1.2/Grounding.owl#" >
  <!ENTITY list "http://www.daml.org/services/owl-s/1.2/generic/ObjectList.owl#" >
  <!ENTITY expr "http://www.daml.org/services/owl-s/1.2/generic/Expression.owl#" >
]>


<rdf:RDF xmlns="http://www.example.org/service.owl"
    xml:base="http://www.example.org/service.owl"
    xmlns:expr="http://www.daml.org/services/owl-s/1.2/generic/Expression.owl#"
    xmlns:list="http://www.daml.org/services/owl-s/1.2/generic/ObjectList.owl#"
    xmlns:process="http://www.daml.org/services/owl-s/1.2/Process.owl#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
```

```
 xmlns:swrl="http://www.w3.org/2003/11/swrl#"
 xmlns:grounding="http://www.daml.org/services/owl-s/1.2/Grounding.owl#"
 xmlns:service="http://www.daml.org/services/owl-s/1.2/Service.owl#"
 xmlns:owl="http://www.w3.org/2002/07/owl#"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:profile="http://www.daml.org/services/owl-s/1.2/Profile.owl#">
<owl:Ontology rdf:about="http://www.example.org/service.ow">
    <owl:imports rdf:resource="http://www.daml.org/services/owl-s/1.2/Grounding.owl"/>
    <owl:imports rdf:resource="http://www.daml.org/services/owl-s/1.2/Profile.owl"/>
    <owl:imports rdf:resource="http://www.daml.org/services/owl-s/1.2/Service.owl"/>
</owl:Ontology>



<!--
///////////////////////////////////////////////////////////////////////////////////////
//
// Object Properties
//
///////////////////////////////////////////////////////////////////////////////////////
 -->




<!-- http://www.example.org/service.ow#p_E16 -->

<owl:ObjectProperty rdf:about="http://www.example.org/service.ow#p_E16"/>
```

```
<!-- http://www.example.org/service.ow#p_E26 -->

<owl:ObjectProperty rdf:about="http://www.example.org/service.ow#p_E26"/>



<!-- http://www.example.org/service.ow#p_E27 -->

<owl:ObjectProperty rdf:about="http://www.example.org/service.ow#p_E27"/>



<!-- http://www.example.org/service.ow#p_E28 -->

<owl:ObjectProperty rdf:about="http://www.example.org/service.ow#p_E28"/>



<!-- http://www.example.org/service.ow#p_E29 -->

<owl:ObjectProperty rdf:about="http://www.example.org/service.ow#p_E29"/>



<!-- http://www.example.org/service.ow#p_E31 -->

<owl:ObjectProperty rdf:about="http://www.example.org/service.ow#p_E31"/>



<!-- http://www.example.org/service.ow#p_E32 -->
```

```
<owl:ObjectProperty rdf:about="http://www.example.org/service.ow#p_E32"/>



<!-- http://www.example.org/service.ow#p_E52 -->

<owl:ObjectProperty rdf:about="http://www.example.org/service.ow#p_E52"/>



<!--
///////////////////////////////////////////////////////////////////////////////////////
//
// Classes
//
///////////////////////////////////////////////////////////////////////////////////////
 -->




<!-- http://purl.org/goodrelations/v1#Qualitative -->

<owl:Class rdf:about="http://purl.org/goodrelations/v1#Qualitative"/>



<!-- http://www.example.org/service.ow#ReadyMixConcrete -->

<owl:Class rdf:about="http://www.example.org/service.ow#ReadyMixConcrete">
    <rdfs:subClassOf>
        <owl:Restriction>
```

```
            <owl:onProperty rdf:resource="http://www.example.org/service.ow#p_E29"/>
            <owl:someValuesFrom
rdf:resource="http://purl.org/goodrelations/v1#Qualitative"/>
         </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
         <owl:Restriction>
            <owl:onProperty rdf:resource="http://www.example.org/service.ow#p_E52"/>
            <owl:someValuesFrom
rdf:resource="http://purl.org/goodrelations/v1#Qualitative"/>
         </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
         <owl:Restriction>
            <owl:onProperty rdf:resource="http://www.example.org/service.ow#p_E26"/>
            <owl:someValuesFrom
rdf:resource="http://purl.org/goodrelations/v1#Qualitative"/>
         </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
         <owl:Restriction>
            <owl:onProperty rdf:resource="http://www.example.org/service.ow#p_E32"/>
            <owl:someValuesFrom
rdf:resource="http://purl.org/goodrelations/v1#Qualitative"/>
         </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
         <owl:Restriction>
            <owl:onProperty rdf:resource="http://www.example.org/service.ow#p_E16"/>
```

```
                <owl:someValuesFrom
rdf:resource="http://purl.org/goodrelations/v1#Qualitative"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="http://www.example.org/service.ow#p_E27"/>
                <owl:someValuesFrom
rdf:resource="http://purl.org/goodrelations/v1#Qualitative"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="http://www.example.org/service.ow#p_E28"/>
                <owl:someValuesFrom
rdf:resource="http://purl.org/goodrelations/v1#Qualitative"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="http://www.example.org/service.ow#p_E31"/>
                <owl:someValuesFrom
rdf:resource="http://purl.org/goodrelations/v1#Qualitative"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:comment>This is a class to represent ready mix conceret</rdfs:comment>
    </owl:Class>



    <!--
    ///////////////////////////////////////////////////////////////////////////////////////
    //
    // Individuals
    //
    ///////////////////////////////////////////////////////////////////////////////////////
     -->


   <!-- http://www.example.org/service.ow#Concrete -->


    <owl:NamedIndividual rdf:about="http://www.example.org/service.ow#Concrete">
        <rdf:type rdf:resource="&process;Input"/>
        <rdfs:label>Concrete</rdfs:label>
        <process:parameterType
rdf:datatype="&xsd;anyURI">http://www.w3.org/2002/07/owl#Thing</process:paramet
erType>
    </owl:NamedIndividual>



    <!-- http://www.example.org/service.ow#getUnitcostAtomicProcessGrounding -->


    <owl:NamedIndividual
rdf:about="http://www.example.org/service.ow#getUnitcostAtomicProcessGrounding">
        <rdf:type rdf:resource="&grounding;WsdlAtomicProcessGrounding"/>
        <grounding:wsdlInputMessage
rdf:datatype="&xsd;anyURI">http://ConcreteWS1/#getUnitcost</grounding:wsdlInputM
essage>
        <grounding:wsdlOutputMessage
rdf:datatype="&xsd;anyURI">http://ConcreteWS1/#getUnitcostResponse</grounding:ws
dlOutputMessage>
```

```
        <grounding:wsdlDocument
rdf:datatype="&xsd;anyURI">http://localhost:8080/ConcreteWebService1/ConcreteWS1
?WSDL</grounding:wsdlDocument>

        <grounding:owlsProcess
rdf:resource="http://www.example.org/service.ow#getUnitcostProcess"/>

        <grounding:wsdlOutput>

            <rdf:Description>

                <rdf:type rdf:resource="&grounding;WsdlOutputMessageMap"/>


<grounding:wsdlMessagePart>http://localhost:8080/ConcreteWebService1/ConcreteWS1
?WSDL#return</grounding:wsdlMessagePart>

                <grounding:owlsParameter
rdf:resource="http://www.example.org/service.ow#return"/>

            </rdf:Description>

        </grounding:wsdlOutput>

        <grounding:wsdlOperation>

            <rdf:Description>

                <rdf:type rdf:resource="&grounding;WsdlOperationRef"/>

                <grounding:operation
rdf:datatype="&xsd;anyURI">http://localhost:8080/ConcreteWebService1/ConcreteWS1
?WSDL#getUnitcost</grounding:operation>

            </rdf:Description>

        </grounding:wsdlOperation>

        <grounding:wsdlInput>

            <rdf:Description>

                <rdf:type rdf:resource="&grounding;WsdlInputMessageMap"/>

                <grounding:xsltTransformationString>
```

```
&lt;xsl:stylesheet                                    version=&quot;1.0&quot;
xmlns:xsl=&quot;http://www.w3.org/1999/XSL/Transform&quot;
xmlns:rdf=&quot;http://www.w3.org/1999/02/22-rdf-syntax-ns#&quot;
xmlns:MyService=&quot;http://www.freeclass.eu/freeclass_v1#&quot;
xmlns=&quot;urn:ch:unibas:dbis:services&quot;&gt;

        &lt;xsl:template match=&quot;//MyService:ReadyMixConcrete&quot;&gt;

                &lt;xsl:variable                                 name=&quot;X1&quot;
select=&quot;MyService:p_E52/@rdf:resource&quot;/&gt;

                &lt;xsl:variable                                 name=&quot;X2&quot;
select=&quot;MyService:p_E27/@rdf:resource&quot;/&gt;

                &lt;xsl:variable                                 name=&quot;X3&quot;
select=&quot;MyService:p_E31/@rdf:resource&quot;/&gt;

                &lt;xsl:variable                                 name=&quot;X4&quot;
select=&quot;MyService:p_E32/@rdf:resource&quot;/&gt;

                &lt;xsl:variable                                 name=&quot;X5&quot;
select=&quot;MyService:p_E28/@rdf:resource&quot;/&gt;

                &lt;xsl:variable                                 name=&quot;X6&quot;
select=&quot;MyService:p_E29/@rdf:resource&quot;/&gt;

                &lt;xsl:variable                                 name=&quot;X7&quot;
select=&quot;MyService:p_E26/@rdf:resource&quot;/&gt;

                &lt;xsl:variable                                 name=&quot;X8&quot;
select=&quot;MyService:p_E16/@rdf:resource&quot;/&gt;

                &lt;ReadyMixConcrete&gt;

                        &lt;p_E52&gt;

                                &lt;xsl:value-of                 select=&quot;substring-
after($X1,&apos;#&apos;)&quot;/&gt;

                        &lt;/p_E52&gt;


                        &lt;p_E27&gt;

                                &lt;xsl:value-of                 select=&quot;substring-
after($X2,&apos;#&apos;)&quot;/&gt;

                        &lt;/p_E27&gt;
                        &lt;p_E31&gt;
                            &lt;xsl:value-of                    select=&quot;substring-
after($X3,&apos;#&apos;)&quot;/&gt;
                        &lt;/p_E31&gt;

                        &lt;p_E32&gt;
                            &lt;xsl:value-of                    select=&quot;substring-
after($X4,&apos;#&apos;)&quot;/&gt;
                        &lt;/p_E32&gt;

                        &lt;p_E28&gt;
                            &lt;xsl:value-of                    select=&quot;substring-
after($X5,&apos;#&apos;)&quot;/&gt;
                        &lt;/p_E28&gt;

                        &lt;p_E29&gt;
                            &lt;xsl:value-of                    select=&quot;substring-
after($X6,&apos;#&apos;)&quot;/&gt;
                        &lt;/p_E29&gt;

                        &lt;p_E26&gt;
                            &lt;xsl:value-of                    select=&quot;substring-
after($X7,&apos;#&apos;)&quot;/&gt;
                        &lt;/p_E26&gt;
                        &lt;p_E16&gt;
                            &lt;xsl:value-of                    select=&quot;substring-
after($X8,&apos;#&apos;)&quot;/&gt;
                        &lt;/p_E16&gt;
                    &lt;/ReadyMixConcrete&gt;
                &lt;/xsl:template&gt;
```

```
&lt;/xsl:stylesheet&gt;

                    </grounding:xsltTransformationString>

<grounding:wsdlMessagePart>http://localhost:8080/ConcreteWebService1/ConcreteWS1
?WSDL#Concrete</grounding:wsdlMessagePart>
            <grounding:owlsParameter
rdf:resource="http://www.example.org/service.ow#Concrete"/>
        </rdf:Description>
      </grounding:wsdlInput>
  </owl:NamedIndividual>



  <!-- http://www.example.org/service.ow#getUnitcostGrounding -->


  <owl:NamedIndividual
rdf:about="http://www.example.org/service.ow#getUnitcostGrounding">
      <rdf:type rdf:resource="&grounding;WsdlGrounding"/>
      <grounding:hasAtomicProcessGrounding
rdf:resource="http://www.example.org/service.ow#getUnitcostAtomicProcessGrounding
"/>
      <service:supportedBy
rdf:resource="http://www.example.org/service.ow#getUnitcostService"/>
  </owl:NamedIndividual>



  <!-- http://www.example.org/service.ow#getUnitcostProcess -->


  <owl:NamedIndividual
rdf:about="http://www.example.org/service.ow#getUnitcostProcess">
      <rdf:type rdf:resource="&process;AtomicProcess"/>
```

```
        <rdfs:label>getUnitcostProcess</rdfs:label>

        <process:hasInput rdf:resource="http://www.example.org/service.ow#Concrete"/>

        <service:describes
rdf:resource="http://www.example.org/service.ow#getUnitcostService"/>

        <process:hasOutput rdf:resource="http://www.example.org/service.ow#return"/>

    </owl:NamedIndividual>



    <!-- http://www.example.org/service.ow#getUnitcostProfile -->


    <owl:NamedIndividual
rdf:about="http://www.example.org/service.ow#getUnitcostProfile">

        <rdf:type rdf:resource="&profile;Profile"/>

        <profile:serviceName>getUnitcost</profile:serviceName>

        <profile:textDescription>Auto                    generated                    from
http://localhost:8080/ConcreteWebService1/ConcreteWS1?WSDL</profile:textDescripti
on>

        <profile:hasInput rdf:resource="http://www.example.org/service.ow#Concrete"/>

        <service:presentedBy
rdf:resource="http://www.example.org/service.ow#getUnitcostService"/>

        <profile:hasOutput rdf:resource="http://www.example.org/service.ow#return"/>

    </owl:NamedIndividual>



    <!-- http://www.example.org/service.ow#getUnitcostService -->


    <owl:NamedIndividual
rdf:about="http://www.example.org/service.ow#getUnitcostService">

        <rdf:type rdf:resource="&service;Service"/>

        <service:supports
rdf:resource="http://www.example.org/service.ow#getUnitcostGrounding"/>
```

```
        <service:describedBy
rdf:resource="http://www.example.org/service.ow#getUnitcostProcess"/>

        <service:presents
rdf:resource="http://www.example.org/service.ow#getUnitcostProfile"/>

    </owl:NamedIndividual>



    <!-- http://www.example.org/service.ow#return -->


    <owl:NamedIndividual rdf:about="http://www.example.org/service.ow#return">

        <rdf:type rdf:resource="&process;Output"/>

        <rdfs:label>return</rdfs:label>

        <process:parameterType
rdf:datatype="&xsd;anyURI">http://www.w3.org/2001/XMLSchema#string</process:pa
rameterType>

    </owl:NamedIndividual>

</rdf:RDF>
```