KSCE Journal of Civil Engineering (0000) 00(0):1-12

DOI 10.1007/s12205-013-0739-5

Construction Management

pISSN 1226-7988, eISSN 1976-3808
www.springer.com/12205

Ontological Inference of Work Item based on BIM Data

Seulki Lee*, Karam Kim, and Jungho Yu*****

Received December 17, 2013/Revised March 26, 2014/Accepted March 27, 2014/Published Online December 1, 2014

Abstract

When engineers prepare a construction cost estimate for budgeting purposes, they use plans, specifications, and available cost data at the completion of the building design phase. They usually take off the quantities of material and related work items and assign appropriate unit costs. In this case, unit cost assignment is solely at the engineer's professional discretion. Building Information Modeling (BIM) is widely adopted in the building construction industry. Tools can be used to automate material quantity, minimizing the time necessary for engineers to engage in the quantity measuring process. This function, however, does not provide any information on work items that are related to materials in the Bill of Quantity (BOQ). Thus, engineers still need to verify associated work items and assign unit costs. This research proposes an ontological inference of work item that enables an automated search of the most appropriate work items and their associated unit costs. This ontology contains semantic information for work items and work conditions, as well as a semantic reasoning rule that activates the ontology. A case study confirms that the proposed ontology and semantic reasoning rule can work in real-world situations. This paper contributes by eliminating subjective decision-making via search of appropriate work items for cost estimation and the use of BIM data extracted from IFCXML. The proposed ontological approach to building cost estimation will assist engineers in more readily using BIM data from IFCXML and will be helpful in automation of the whole estimation process.

Keywords: *BIM, work condition, work item, cost estimation, ontology*

1. Introduction

When engineers prepare a construction cost estimate for budgeting purposes at the completion of the building design phase, they use plans, specifications, and available cost data. They usually take off the quantities of material and related work items and then assign appropriate unit costs of each material or work. In this case, unit cost assignment is solely at the engineer's professional discretion.

In the construction industry, there is a growing interest in the use of Building Information Modeling (BIM) for coordinated, consistent, and computable building information/knowledge management from design and construction to maintenance and the operation stages of a building's lifecycle (Eastman *et al*., 2008). In this regards, there is increasing interest in building an information/knowledge management system via computing techniques in various domains. These technologies would assist in management and re-use of the engineer's expertise by utilizing computerized applications. There are data-based methodologies such as data mining technologies, which is an interdisciplinary field involving concepts from machine learning, statistics, and mathematics (Kim *et al*., 2011; Taormina *et al*., 2012; Zhang and Chau, 2009).

As a visual database of building components, BIM can provide accurate and automated quantification and assist in significantly reducing variability in cost estimation. BIM-based cost estimation software includes ToCoMan from TocoSoft, CI Estimator from CRC, and VicoOffice from Graphisoft. While most cost-estimation software can perform quantity take-off or import quantity take-off data, all current software requires the use of an associated and/or additional database that contains unit costs and other industry data to prepare an estimate. Theoretically, information for construction management can be easily or automatically obtained from BIM.

However, there are some potential problems in using the BIM data for the cost estimation process. First, this is possible only in cases when BIM is fully acquired; otherwise, the information obtained from BIM is very limited. If architects or engineers use BIM tools such as ArchiCAD or Revit to model a building model, the tools provide functions that automate material quantity take-off, therefore reducing the work required of engineers in the quantity measuring process. However, this function does not provide any information on work items that are related to materials in the Bill of Quantity (BOQ). Second, since there is

*Member, Lecturer, Dept. of Architectural Engineering, Kwangwoon University, Seoul 139-070, Korea (E-mail: selkizz@kw.ac.kr)
**Ph.D. Student, Dept. of Architectural Engineering, Kwangwoon University, Seoul 139-070, Korea (E-mail: karamiz@kw.ac.kr)
***Member, Associate Professor, Dept. of Architectural Engineering, Kwangwoon University, Seoul 139-070, Korea (Corresponding Author, E-mail: myazure@kw.ac.kr)

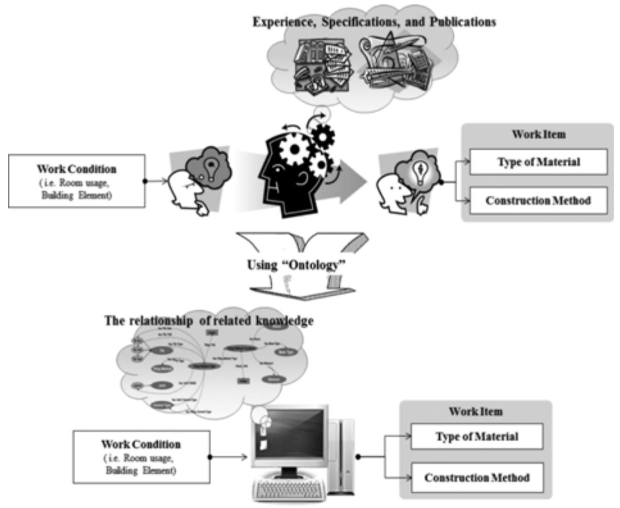

Fig. 1. Concept of Automated Inference Work Item

not enough information to determine a work item with BIM-based data of work conditions, some additional information/knowledge is necessary from an estimation engineer, including knowledge from experience, specifications, and publications. In this regard, the results depends on the engineer's skill level and subjective decisions. Engineers still need to verify associated work items and to assign their unit costs based on the subjective decision making of estimators (see Fig. 1). Finally, as an engineer implements the estimation based on subjective judgment, human errors may occur that lower consistency in the results and thus reduce reliability. Even when work conditions are the same, the determined work item might be different depending on each engineer's skill level and preferences.

This research proposes an ontological inference approach that can be applied to a work item that enables automated search of the most appropriate work items by work conditions. We conducted a case study pertaining to brick work to validate the proposed approach, and an ontology was established based on the type of brick, type of bond, thickness, type of structural reinforcement, and usage context such as room usage, building elements, height and length of building elements, and so forth. This ontology contains semantic information on a brick work item and brick work conditions as well as a semantic reasoning rule that activates the ontology. The case study confirms that the proposed ontology and semantic reasoning rule can be applied to real-world situations. We verify the proposed ontologies using HermiT Reasoner v. 1.5.2 included in Protégé v.3.4.7.

The proposed ontological approach provides a methodology for complete automation of cost estimation software. The work condition is understood by the system semantically and the work item is automatically searched. If the same BIM data are used, the same work items can be consistently provided. In addition, the proposed ontologies and process can assist in using BIM data more easily and expanding BIM-based construction management.

2. Review of Related Works

2.1 BIM-based Cost Estimation

Cost estimation is the process of looking into the future and trying to predict project costs and resource requirements (Halpin and Woodhead, 1998). Estimation can be conducted throughout the project life-cycle. The level of detail in the model varies depending on the project phase. For cost estimation, estimators use a standard Work Breakdown Structure (WBS) previously established to hierarchically divide the project into sub-parts. Estimators manually extract the quantities for each WBS item from paper-based drawings. After extraction of the quantities, estimators price each of the cost items using historical company records and catalogs that compile cost information from previous projects. To price physical items using historical cost data, estimators also account for other aspects that incur costs, such as site conditions, duration of construction, work schedule, and working methods. These work conditions, depending on the specific project, influence costs related to required equipment, total hours of labor, or costs incurred by the need to obtain special building permits. This step requires in-depth knowledge and experience and is the least structured step in the estimation process. While most cost estimation software can perform quantity take-off or import quantity take-off data, all cost estimation software requires the use of associated and/or additional databases that contain unit costs and other industry data needed to prepare an estimate.

BIM offers the ability to generate take-offs, counts, and measurements directly from a model. The main benefit of applying BIM-based tools to estimate project costs occurs during the quantity take-off step (Eastman *et al.*, 2008). In particular, BIM needs to fulfill the following requirements (Hartmann *et al*., 2012):

- Provide the required level of detail to generate an estimate;
- Allow estimators to extract quantities of the building components grouped by the company's WBS; and
- Allow the take-off of accurate quantities for each of the cost items defined in the WBS.

At present, several BIM-based cost estimation software applications have been developed to improve the efficiency of estimators, including Innovaya Viusal Estimating (Innovaya, 2010), Vico Estimator 2009 (Vico. 2010), Tokmo Production System (Tokmo, 2009), Success Design Exchange (Cost, 2010), Timberline Extended (Sage, 2010; AECCafe, 2002), and Winest Design Estimation Pro (Winest, 2010).

Some researchers have applied BIM to cost estimation. Lee *et al.* (2009) proposed a BIM property information model that can support quantity take-off of a structural element in order to reflect the cost estimate of a public building. Kwon *et al.* (2011). suggested ways of securing BIM quality for correct quantity take-off, which is critical for BIM applied to the building construction process and its cost estimation. They also set up BIM modeling guides that make it possible to extract correct quantity take-off from BIM data. Staub-French *et al.* (2003)

presented an Industry Foundation Classes (IFC)-based cost estimating system that could directly use the results of IFC files and then automatically apply corresponding prices to accomplish cost estimation according to component geometries and properties. Fu *et al*. (2004) developed a system for life-cycle cost assessment that could automatically extract cost estimating data from the design results of IFC files and then transfer the data to a pre-existing component of life-cycle cost assessment. Ma *et al.* (2010) focused on establishing a framework for BIM-based Construction Cost Estimating (CCE) software based on Chinese standards. The framework laid a solid foundation for developing next-generation CCE software. Ma *et al.* (2011). created a discrimination model for Bill of Quantity (BQ) items and the corresponding rule database and semantic database. An IFC-based discrimination model for BQ items was established, and a mechanism for intelligent generation of bill of quantity from IFC data was formulated. Finally, the mechanism was programmed and verified through an actual project. Hartmann *et al.* (2012) described the implementation of BIM-based tools to support the activities at an estimating department of a construction company on a large infrastructure project. They complement existing implementation theories in construction management that advocate “technology push” implementations during which existing work processes need to be radically changed to align with the functionality of the BIM-based tools.

Estimation accuracy differs depending on estimator capability and the cost estimation method. Lack of design information is an issue for BIM-based cost estimation. Research in this area mostly focuses on automation of quantity take-off or accuracy improvement of take-off. There are not enough studies on the inference of work items or the standardization required for cost estimation. The current research proposes an automation process for inference of work items using BIM and an ontology.

2.2 Ontology in Construction

Berners-Lee *et al.* (2001) proposed an extension to the World Wide Web (WWW), namely, the Semantic Web, which can handle Web data without human intervention. The Semantic Web allows for a semantic search that finds more accurate information compared to the current Web and makes it easy to share and reuse information. Semantic Web data are expressed in a computer-readable format using an ontology. The key technology of the Semantic Web is defined as formal and explicit specification of shared conceptualization (Gruber, 1993). Using an ontology that provides various concepts and their relationships enables automatic reasoning about data from different sources in order to manipulate relevant data or find new knowledge.

Today, ontology is applied to many domains, such as information and communications, medical information, and the Internet, and it has been studied using various methodologies. Several studies have applied ontology to the AEC (Architecture, Engineering, and Construction) industry. Park *et al.* (2013) proposed a framework for proactive construction defect management using an ontological data collection template. The proposed framework provided an integrated defect data management process to utilize inferred knowledge for other projects. Tserng *et al.* (2009) proposed an ontology-based risk management framework of construction projects through a project life cycle. In the research, knowledge extraction for effective risk management by establishing project risk ontology was explored. It demonstrated that ontology-based risk management can apply to risk management work flow and increase the effectiveness of project risk management. El-Diraby and Osman (2011) presented a philosophically-influenced model of infrastructure products that focuses on their functions, roles and semantic attributes in order to facilitate human representation of construction knowledge. The developed ontology provided a conceptualization for knowledge in civil infrastructure. Lee and Jeong (2012) proposed an ontological filter-based collaboration model and presented its computational implementation using a sample model. In the research, to reflect the characteristics of multi-disciplinary collaborative design, they developed the ontological approach using a semantically rich representational method with domain specific and intelligent filters of a building model. Chau (2007) proposed an ontology-based knowledge management system to automatically generate knowledge search components for flow and water quality modeling. The proposed system has ontologies divided into information ontology and domain ontology in order to realize the objective of semantic match for a related knowledge search. Benevolenskiy *et al.* (2012) proposed a methodology for an ontology–based process modeling. The proposed system supported the generation of process schedules for construction projects that could later be used in discrete-event simulation software or workflow programs.

Ontologies are structured through classification systems to support effective information exchange. IFC, an international standard file format, is intended to enable effective information sharing within the AEC industry using ontologies (IAI, 1999). In the construction field, the e-COGNOS ontology (Ei-Diraby *et al.,* 2003) is used to facilitate Semantic Web-based knowledge management. A set of actors use a set of resources to produce a set of products following certain processes within a work environment according to certain conditions. It was developed and implemented through collaboration between researchers and leading European contractors. Additionally, the Semantic Web-based Open Engineering Platform (SWOP) project in Europe has developed a generic and reusable set of Product Modeling Ontologies (PMO) that can be imported and used by end-user product ontologies for any parametric/configurable product type (Commonwealth scientific and industrial research organization, 2005). Kristian *et al.* (2010) suggest ontologies to support a Radio Frequency Identification (RFID)-based link between virtual models and construction components. The ontologies are reviewed from an ontology consumer (system developer) point of view and categorized according to their applicability to specification of technical services, resources, organizational relations, business processes, and overall frameworks for ontology descriptions and their relations.

In summary, there have been quite a number of studies on ontology in construction during the last decade. Some of them apply ontology to product modeling, while others address key concepts in construction or with a focus on conceptualization of construction knowledge. However, it appears that there has not been any specific research on building and utilizing an ontology to automatically search for the most appropriate work items for cost estimation.

3. Process of Ontological Inference of Work Item

The overall process for automated inference of work items is summarized in Fig. 2. During design architects develop a building model using the BIM design tool, and BIM data such as work conditions are input into the building model. In this regard, work conditions can be included under the entities of the IFC schema such as IfcSpace, IfcBuildingElement, and IfcMaterial entities. To use the work conditions, the BIM data has to be converted to the Resource Description Framework (RDF) data of the knowledge database in the work conditions conversion layer. The reasoning layer creates inferred knowledge that includes a work item by means of a reasoning process that is based on two ontologies (Work Condition Ontology (WCO) and Work Item Ontology (WIO)) and the work condition RDF data of the knowledge database. Thus, the knowledge database has all of the information about the work items, including specific work conditions extracted from the building model. Finally, data from the knowledge database can be consistently and efficiently used for cost estimation, since arbitrary human judgement of the cost estimators is eliminated in the process of determining the work items.

3.1 Work Condition Conversion Layer

To use the work conditions from a building model, BIM data must be extracted through a specific extraction algorithm. The extraction algorithm has to be developed per the IFC schema as an international standard file format for data interoperability. For example, the data for room usage is provided by the IfcSpace entity in the IFC schema as the property of name. Thus, to extract the work conditions data, the BIM data of work conditions must be pre-defined as an extraction algorithm. After the work conditions data has been extracted by the Extensible Markup Language (XML) format, it is converted to a Web Ontology Language (OWL)-based RDF data format to infer upon using ontological technology (Schevers and Drogemuller 2006). Following the development of OWL, knowledge based on concepts and their relationships in the OWL ontology were expressed in the RDF, which is represented in a triple structure of subject, predicate, and object (Klyne *et al*., 2004).

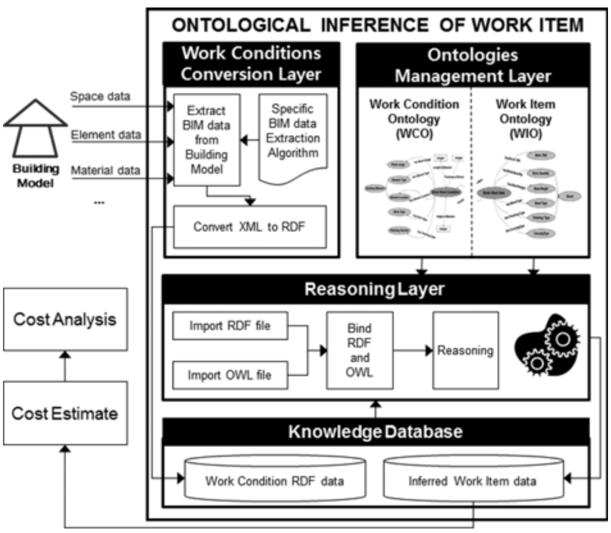

Fig. 2. Overall Process for Automated Inference of a Work Item

3.2 Ontologies Management Layer

In the ontologies management layer, both ontologies (WCO and WIO) are logically and systemically combined to infer the work item using the type of work condition. The role of WCO is to provide concepts and relationships for sharing information when determining a work item. To investigate the specific work conditions and their relationships to the building model, we referred to elements (including space) and properties in the IFC schema. These data were defined in OWL and classified into each type of work item. The classification of the work condition might be variously classified by the type of work item.

In the WIO, every work item is classified by a standard classification, such as Omniclass or Uniclass. In order to infer a work item, the specific work conditions are organized as an individual attribute of a work item. Since some specific work conditions are combined as an attribute, it has been recognized as a necessary and sufficient condition of a work item.

3.3 Reasoning Layer

After the work condition RDF data have been stored in the knowledge database, the reasoning layer creates inferred work item data by reasoning based on the two ontologies and the work condition RDF data before being added to the inferred knowledge database. On the reasoning layer, the RDF formatted work condition file and the OWL-formatted two ontology files are bound using a binding engine, such as the Jena Application Program Interface (API), which is a simple, non-redundant collection of ontologies and RDF data. Then the reasoning engine takes the bound data and generates inferred work item data that is placed into the knowledge database.

4. Case Study

4.1 Information Extraction from IFCXML and Conversion to RDF

To automatically infer the work item, BIM data such as work conditions are first extracted from IFCXML. Then the extracted BIM data are converted into RDF data in a machine-understandable format. To define the work conditions determining a brick work item from an IFCXML file, we analyze the IFC schema and

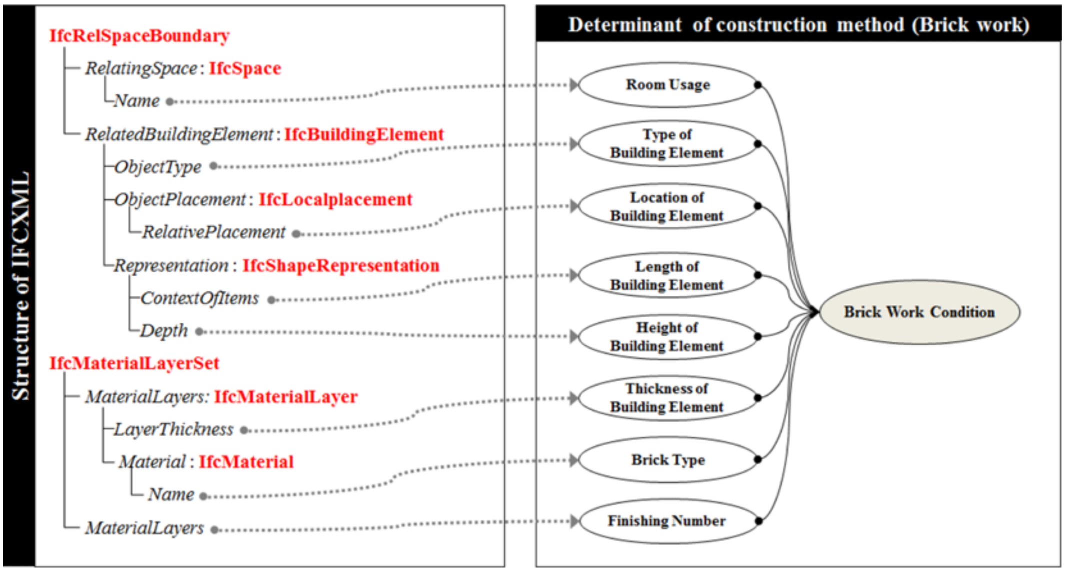

Fig. 3. Relationships among Elements of the IFCXML

practical considerations. Some examples include brick work; room usage; type, location, height, length and thickness of building elements; brick type; and the finishing number, which are defined as work conditions and, are determinants of the brick work item. In order to show the process for automated inference, we built a sample building model using ArchiCAD14, which supports 3D-based Computer Aided Design (CAD) modeling. Then, as shown in Fig. 3, we analyzed the IFCXML of the sample model.

The basic structures of BIM data are based on the IFCXML (Liebich, 2009), and the elements and attributes of the IFCXML are derived from the standard IFC. In this research, eight entities were extracted from IFCXML: IfcRelSpaceBoundary, IfcSpace, IfcElement, IfcLocalPlacement, IfcShapeRepresentation, IfcMaterialLayerset, IfcMaterialLayer, and IfcMaterial. Fig. 3 shows the relationships among these elements. That is, room usage can be extracted from the Name attribute of IfcSpace in IfcRelSpaceBoundry; the building element of its space can be extracted from ObjectType attribute of IfcBuildingElement in IfcRelSpaceBoundry; and location of the building element can be extracted from the RelativePlacement attribute of IfcLocalPlacement in IfcBuildingElement. Length and height of the building element can be extracted from ContextOfItems and Depth attributes of IfcShapeRepresentation in IfcBuildingElement; thickness of the building element can be extracted from LayerThickness attribute of IfcMaterialLayer in IfcMaterialLayerSet; brick type can be extracted from the Name attribute of IfcMaterial in IfcMaterialLayerSet; and the finishing number can be confirmed from the number of referenced MaterialLayers attributes of IfcMaterialLayerSet. The BIM data is extracted according to relationships among elements of the IFCXML that should be converted into RDF.

The purpose of this research is conceptual is order to generate automated inference of work items. We used an ontology based on the assumption that BIM data can be extracted from IFCXML. Therefore, technologically, extraction and conversion tools should be further developed in future research.

4.2 Ontology for Automatic Inference of Standard Work Item

This research introduces the knowledge structure of cost estimation using an ontology that enables an automated search of the most appropriate work items. The BIM tool or related cost estimation software automatically enables reasoning for searching of work items. This research provided brick work as a case study to evaluate the proposed approach. The ontology of the case study was established based on brick thickness, bond methods, and their usage context such as room usage, building elements, the height and length of building elements, and so forth. This ontology contains semantic information on brick work items and brick work conditions, as well as a semantic reasoning rule that activates the ontology. To establish the ontology, we referenced standard of estimate, standard technical specifications, historical data, and publications about construction methods. It must still be considered, however, that a work item which consists of materials and a construction method, depends on construction type and technological advancement.

To enable the automated inference, a work condition ontology, which consists of determinants of a work item, and a work item ontology that consists of factors defining a type of construction method should be established. In this research, a Brick Work Condition Ontology (BWCO) and a Brick Work Item Ontology (BWIO) were designed using Protégé v3.4.7, which is a Java-based general ontology editor. The Protégé program is developed to design an OWL-based ontology and it can validate the ontology using a Hermit-reasoning engine. OWL has three types of sublanguages: OWL full, OWL Description Logic (DL)

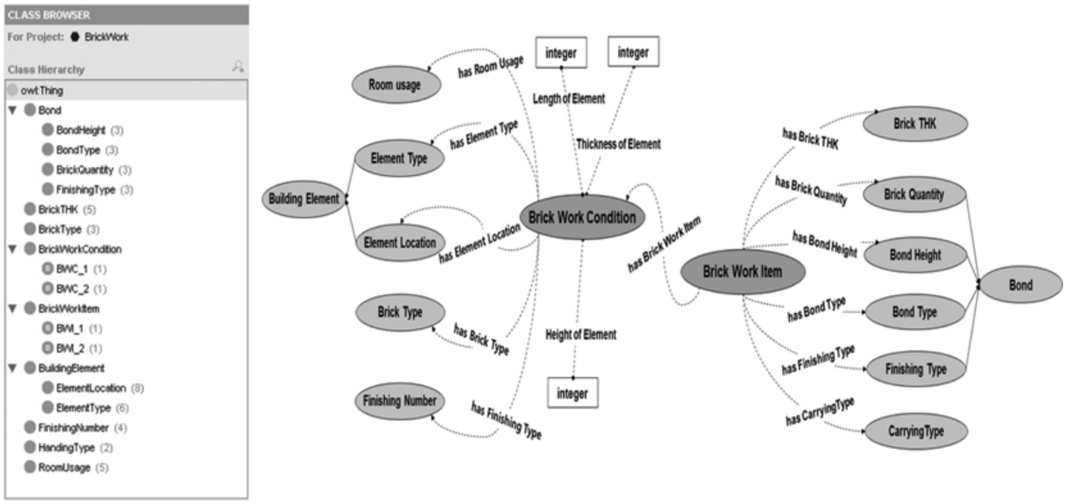

Fig. 4. Frameworks of BWCO and BWIO

Table 1. Class Definitions

Class		Description and Instances
Room Usage		– Room Usage: kitchen, toilet, laundry room, hall, entrance, bath
Building Element	Element Type	– Element Type: internal wall, external wall, floor, ceiling, column, arch
	Element Location	– Element Location: Lev. 1, Lev. 2, Lev. 3, …. , Lev. 7, over Lev. 7
Brick Type		– Brick Type: Cement brick, Red brick, Fire brick
Finishing Number		– Finishing Number: 0, 1, 2,over 2
Brick Work Condition		– Brick Work Condition: room usage, the type, location, height, length and thickness of building elements, and brick type
Brick THK		– Brick Thickness: 0.5B, 1.0B, 1.5B, 2.0B, 2.5B
Bond	Brick Quantity	– Brick Quantity: under 5,000, 5,000~10,000, over 10,000
	Bond Height	– Bond Height: under 3.6 m, 3.6~7.2, over 7.2 m
	Bond Type	– Bond Type: English bond, Dutch bond, Flemish bond, American bond
	Finishing Type	– Finishing Type: Spray coat, Masonry, Mortar
Handing Type		– Handing Type: Human handing, Handcart, Mechanical handing
Brick Work Item		– Brick Work Item: brick thickness, brick quantity, bond height, bond type, finishing type, handing type

(Baader *et al.*, 2003)., and OWL lite. OWL full is the most expressive, OWL lite is the least expressive, and OWL DL falls between the two but is significantly more expressive than OWL lite. Since OWL DL facilitates an automatic check of how classes defined in the ontology are subsumed and whether they are consistent, we selected OWL DL for the development of our ontology for a brick work item. The Protégé program has a simple user interface to develop an ontology, and it is especially feasible to design OWL DL. BWCO was used to recognize BIM data, which were extracted from IFCXML and converted into a RDF format as work conditions. The RDF data contains room usage; type, location, height, length and thickness of building elements; brick type; and the finishing number. BWIO was used to define the work items, which consisted of brick thickness, brick quantity, bond height, bond type, finishing type, and carrying type. Fig. 4 shows the framework for BWCO and BWIO. A continuous line (subclassof) was used to express the hierarchical relationship between class and sub-class, and a dotted line (objectproperty and datatypeproperty) was used to express the relationship between classes and the relationship between class and data value.

4.2.1 Definition of Class

BWIO contains four classes: BrickTHK, Bond, CarryingType and BrickWorkItem. The Bond class contains three sub-classes: BrickQuantity, BrickHeight, BondType, and FinishingType. The BrickWorkItem class contains semantic reasoning rules that are defined by combining other BWIO classes.

BWCO contains five classes: RoomUsage, BuildingElement, BrickType, FinishingNumber and BrickWorkCondition. The BuildingElement class contains two sub-classes: ElementType, and ElementLocation. The BrickWorkCondition class contains semantic reasoning rules that are defined by combining other BWCO classes as well as the semantic reasoning rules to connect a work item defined in BWIO with a work condition recognized by BWCO. The classes of the proposed ontologies are described in Table 1.

Table 2. Definition of Properties

Object Property	Description
has Room Usage	*Brick Work Condition* has only one member of the *RoomUsage* class
has Element Type	*Brick Work Condition* has only one member of the *ElementType* class
has Element Location	*Brick Work Condition* has only one member of the *ElementLocation* class
has Brick Type	*Brick Work Condition* has only one member of the *BrickType* class
has Finishing Number	*Brick Work Condition* has only one member of the *FinishingNumber* class
has Brick Work Item	*Brick Work Condition* has only one member of the *BrickWorkItem* class
has Brick THK	*Brick Work Item* has only one member of the *BrickTHK* class
has Brick Quantity	*Brick Work Item* has only one member of the *BrickQuantity* class
has Bond Height	*Brick Work Item* has only one member of the *BondHeight* class
has Bond Type	*Brick Work Item* has only one member of the *BondType* class
has Finishing Type	*Brick Work Item* has only one member of the *FinishingType* class
has Handing Type	*Brick Work Item* has only one member of the *HandingType* class

4.2.2 Property Definition

The OWL property consists of two types: owl:ObjectProperty and owl:DatatypeProperty. Object property (owl:ObjectProperty) is used to express the relationship between objects. Data type property (owl:DatatypeProperty) is used to express the relationship between an object and data value such as a numerical value (i.e., height, length and thickness of building elements). Although brick thickness may vary, the type of brick thickness is usually fixed. Thus, we defined brick thickness as an object property. The object properties of the proposed ontologies are described in Table 2.

4.3 Semantic Reasoning

The conceptual process for semantic reasoning using BWCO and BWIO is shown in Fig. 5. The first step is to recognize the work condition: BIM data extracted from IFCXML are recognized as one of the work conditions using BWCO. The second step is to select the work item: the most appropriate work item is selected using BWIO.

For example, each kitchen, wall, level 2, 5000 mm (length), 4000 mm (height), 190 mm (thickness), and cement brick is automatically recognized as an instance of “RoomUsage”, “ElementType”, “ElementLocation”, “ElementLength”, “ElementHeight”, “ElementThkiness”,“FinishingNumber”, and BrickType classes. Then, the data are recognized as an instance of BWC_1, one of the sub-classes of the “Brick Work Condition” class. Finally, the most appropriate work item is recommended, because the BWC_1 class contains BWI_1, a sub-class of the Brick Work Item class, is defined as a necessary condition.

The semantic reasoning rules for automated inference of work items are as follows:

- **BWC_1**
  – Necessary & Sufficient
  · rhasRoomUsage has ***Kitchen***
  · hasElementType has ***Internalwall***
  · hasElementLocation has ***Level 2***
  · hasElementLength has ***5000***
  · hasElementHeight has ***4000***

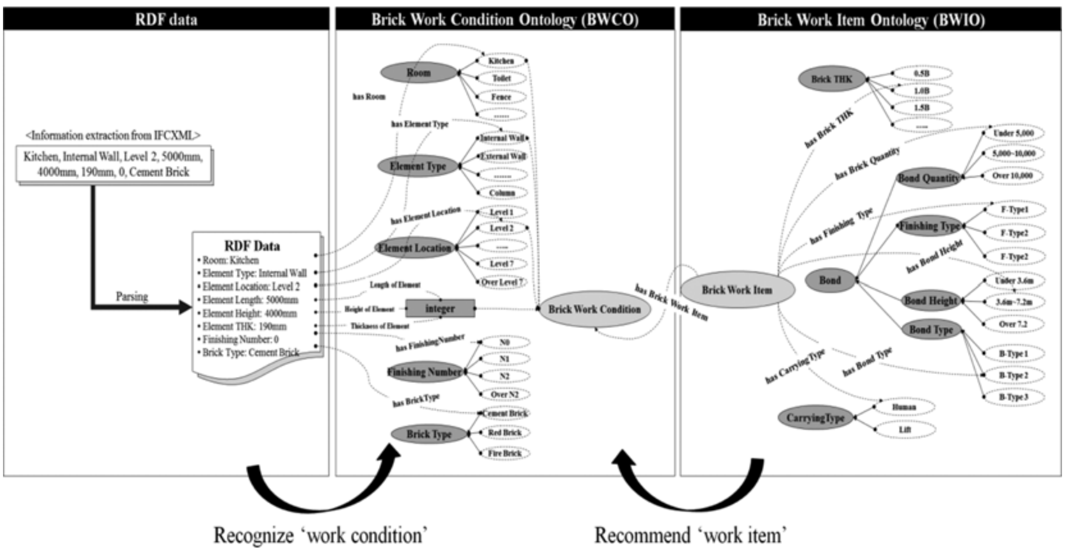

Fig. 5. Conceptual Process of Semantic Reasoning for a Work Item

· hasElementThickness has ***190***
· hasFinishingNumber has ***N0***
· hasBrickType has ***Cement Brick***
– Necessary
· hasBrickWorkItemType has ***BWI_1***

- **BWI_1**
  – Necessary & Sufficient
  · hasBrick THK has 1.0B
  · hasBrickQuantity has under 5,000
  · hasBondHeight has 3.6m-7.2m
  · hasBondType has B-Type 2
  · hasFinishingType has F-Type1
  · hasCarryingType has Human

When the RDF data have been stored in the knowledge base, the reasoning engine creates inferred knowledge based on two domain ontologies (i.e., BWCO and BWIO) and RDF data before adding the inferred knowledge into the base.

We verified BWCO, BWIO, and the semantic reasoning rule using HermiT Reasoner v 1.5.2 included in Protégé v.3.4.7.

4.4 Evaluation

4.4.1 Process of Evaluation

To evaluate the proposed approach, we present a building model to illustrate the proposed ontology. A sample building model was created with ArchiCAD 16 with a net area of 170 interior square meters including 2 stories, 10 walls (8 exterior walls and 2 interior walls), and 4 spaces. Fig. 6 shows the floor plans of the sample building model.

The 10 walls have different types of material components. There are four walls made of cement brick (wall-1 to wall-4) and four walls made of red brick (wall-5 to wall-8). In addition, there are two interior walls made of fire brick (wall-9 and wall-10). All of the materials are of a finishing defined as a plaster. Table 3 shows the wall types of the 10 walls in the sample building model.

According to the conditions of the 10 walls from the sample building model, we compared the results of the inference of work items between the proposed ontological approach and the traditional approach. The traditional approach consisted of an interview with three practicing engineers on building work items in situ in South Korea. They determined a brick work item considering the same brick work conditions of the 10 walls as in the sample building model. We compared the results of determined brick work items for both approaches.

4.4.2 Extraction and Conversion the BIM Data

To extract the BIM data of the sample building model, the building model was exported to an IFCXML file in an

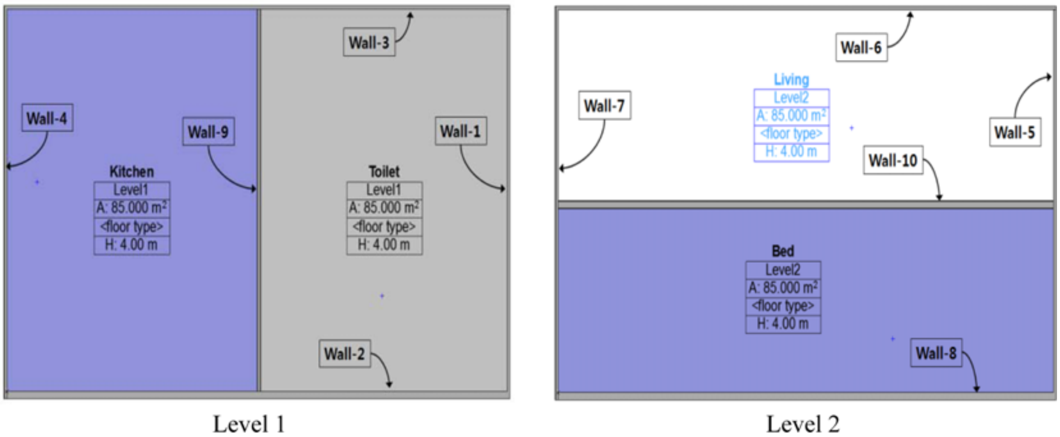

Fig. 6. Floor Plans of the Sample Building Model

Table 3. Definitions of the Wall Types

Wall types	Used Wall	Thickness	Components
Cement Brick 0.5B	Wall-1	90 mm	Cement Brick
Cement Brick 1.0B	Wall-2	190 mm	Cement Brick
Cement Brick 0.5B finishing1	Wall-3	100 mm	Cement Brick + Plaster
Cement Brick 0.5B finishing2	Wall-4	110 mm	Plaster + Cement Brick + Plaster
Red Brick 0.5B	Wall-5	90 mm	Red Brick
Red Brick 1.0B	Wall-6	190 mm	Red Brick
Red Brick 0.5B finishing1	Wall-7	100 mm	Red Brick + Plaster
Red Brick 0.5B finishing2	Wall-8	110 mm	Plaster + Red Brick + Plaster
Fire Brick 0.5B finishing2	Wall-9	110 mm	Plaster + Fire Brick + Plater
Fire Brick 0.5B finishing2	Wall-10	110 mm	Plaster + Fire Brick + Plater

Table 4. Extracted BIM Data from the IFCXML File of the Sample Building Model

Wall Names	Width	Height	Elevation	Related Space	Related Material
Wall-1	10,000	4,000	0	Level1_Toilet	CementBrick_0.5B
Wall-2	17,000	4,000	0	Level1_Toilet Level1_Kitchen	CementBrick_1.0B
Wall-3	17,000	4,000	0	Level1_Toilet Level1_Kitchen	CementBrick_0.5B finishing1
Wall-4	10,000	4,000	0	Level1_Kitchen	CementBrick_0.5B finishing2
Wall-5	10,000	4,000	4,000	Level2_Living Level2_Bed	RedBrick_0.5B
Wall-6	17,000	4,000	4,000	Level2_Living	RedBrick_1.0B
Wall-7	17,000	4,000	4,000	Level2_Living Level2_Bed	RedBrick_0.5B finishing1
Wall-8	10,000	4,000	4,000	Level2_Bed	RedBrick_0.5B finishing2
Wall-9	10,000	4,000	0	Level1_Toilet Level1_Kitchen	FireBrick_0.5B finishing2
Wall-10	17,000	4,000	4,000	Level2_Living Level2_Bed	FireBrick_0.5B finishing2

```
<rdf:Description rdf:ID="Wall-1">
<hasRoomUsage rdf:resource="#Level1_Toilet"/>
<hasElementType rdf:resource="#ExternalWall"/>
<hasElementLocation rdf:resource="#Level1"/>
<hasElementLength rdf:datatype="http://www.w3.org/2001/XMLSchema#int">10000</hasElementLength>
<hasElementHeight rdf:datatype="http://www.w3.org/2001/XMLSchema#int">4000</hasElementHeight>
<hasElementTHK rdf:datatype="http://www.w3.org/2001/XMLSchema#int">90</hasElementThickness>
<hasFinishingNumber rdf:resource="#N0"/>
<hasBrickType rdf:resource="#CementBrick"/>
</rdf:Description>
```

Fig. 7. Part of the Created RDF File (Wall-1)

international standard format using ArchiCAD 16. The exported IFCXML file contained the BIM data including material and representation data regarding the 10 walls and related spaces using an IFC 2 × 3 schema. Thus, we can extract the BIM data including the wall name, the width, the height, the elevation, the related space, and the related material data from the IFCXML file and convert this into an RDF file. Table 4 shows the extracted BIM data from the IFCXML file.

Using the extracted BIM data, the BIM data was converted to an RDF file in order to infer a work item via BWCO and BWIO. In this regard, all of the extracted BIM data must be converted into the RDF format including all of the relationships between the extracted BIM data. There are eight attributes per instance on the RDF file to be recognized by BWC on the BWCO. Fig. 7 shows a part of the created RDF file.

4.4.3 Semantic Reasoning

After the RDF file has been created using the extracted BIM data, the reasoning engine creates inferred knowledge based on both ontologies (i.e., BWCO and BWIO) and the created RDF file. We used the Jena API (The apache software foundation, 2014). to bind the ontologies and the RDF file, which is a simple, non-redundant collection of ontologies and material RDF data. In addition, we also used the Bossam Reasoner (Jang and Sohn, 2004). for reasoning the proposed ontological inference, which has a Rete-based rule engine with native support for reasoning for both RDF and OWL files. Since both the Jena API and the Bossam Reasoner are provided as open source and allows for simple and powerful development of the ontological approach, this paper has chosen them to evaluate the proposed approach. The semantic reasoning consists of three sub-processes: 1) the reasoning layer binds the OWL and RDF files using the Jena API into bound knowledge; 2) the Bossam Reasoner takes the bound knowledge and generates inferred knowledge using reasoned OWL-DL; and 3) the knowledge base stores the knowledge.

In this case, the whole process, from step 1 to step 3, is executed automatically by the prototype system to evaluate the proposed ontologies as a reasoning step. Using the inferred knowledge, we can retrieve the brick work item with the brick work conditions. Fig. 8 shows the process for semantic reasoning.

4.4.4 Results of Evaluation

To validate the proposed ontological approaches, we interviewed three engineers, who have more than three years of experience in South Korea, to determine the brick work items using the same brick work conditions as the sample building model. The brick work conditions were presented as follows: RoomUsage, ElementType, ElementLocation, BrickType, FinishingNumber, and the wall’s representation data, such as height and width. With the

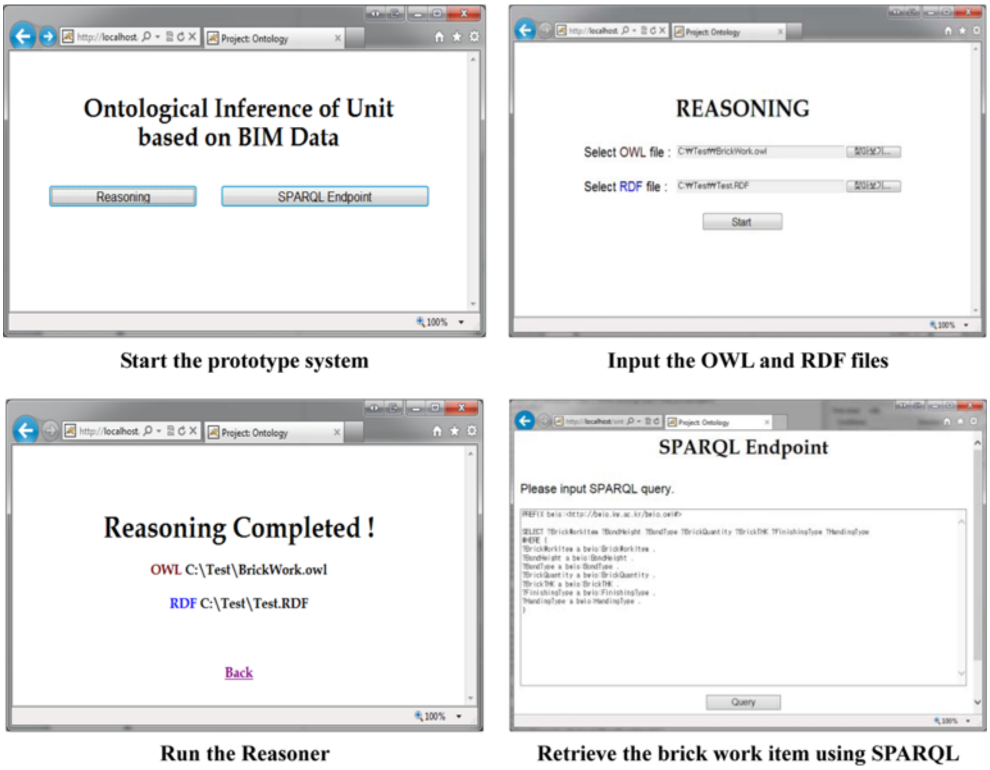

Fig. 8. Semantic Reasoning Process

Table 5. Each Engineer's Choice of Brick Work Items (Wall-1 to Wall-5)

Brick work items	Wall-1				Wall-2				Wall-3				Wall-4				Wall-5			
	A	B	C	Cons.	A	B	C	Cons.	A	B	C	Cons.	A	B	C	Cons.	A	B	C	Cons.
BondType	2	4	4	2/3	3	4	2	0/3	3	4	4	2/3	4	4	4	3/3	2	4	4	2/3
Finishing Type	3	3	2	2/3	2	3	2	2/3	2	3	3	2/3	2	3	3	2/3	3	3	2	2/3
CarryingT ype	3	3	2	2/3	3	3	2	2/3	2	3	2	2/3	2	3	2	2/3	3	3	2	2/3

Table 6. Each Engineer's Choice of Brick Work Items (Wall-6 to Wall-10)

Brick work items	Wall-6				Wall-7				Wall-8				Wall-9				Wall-10			
	A	B	C	Cons.	A	B	C	Cons.	A	B	C	Cons.	A	B	C	Cons.	A	B	C	Cons.
BondType	1	4	2	0/3	3	4	4	2/3	1	4	4	2/3	3	1	3	2/3	2	1	3	0/3
Finishing Type	3	3	2	2/3	3	3	3	3/3	3	3	3	3/3	2	3	1	0/3	3	3	1	2/3
Carrying Type	2	3	2	2/3	3	3	2	2/3	2	3	2	2/3	1	2	1	2/3	3	2	1	0/3

brick work conditions, the engineers determined the options for the brick work items as BrickQuantity, BondHeight, BondType, FinishingType, and CarryingType. Since the brick work items have no right answer, the engineers answered in various ways. The given options of the brick work items were BondType (1: English bond, 2: Dutch bond, 3: Flemish bond, and 4: American bond), FinishingType (1: Spray coat, 2: Masonry, and 3: Mortar), and CarryingType (1: Manually carried, 2: Handcart, and 3: Mechanically carried). In addition, the consistency value was calculated by how many engineers chose the same option. Although the maximum value of the consistency is 3/3 for several walls, the consistency had different values. This means that the decision is made on a case-by-case basis depending on the expert's subjective judgment. Our research presents three engineers opinions, but if the number of engineers is larger, the consistency might be smaller. For example, while engineer A chose BondType-2 (Dutch bond), engineers B and C chose BondType-4 (American bond). In this case, the consistency of the BondType for Wall-1 becomes 2/3. Table 5 and Table 6 show each engineer's choice of brick work items.

As a result, the BrickQuantity and BondHeight values have full consistency because these values were inferred from the same brick work condition by the representation data of the sample building model (see Table 7).

However, BondType, FinishingType, and CarryingType had different values, since these components depended on the

Table 7. Brick Quantity and Bond Height Data of Brick Work Items

Brick work items	Wall-1	Wall-2	Wall-3	Wall-4	Wall-5	Wall-6	Wall-7	Wall-8	Wall-9	Wall-10
Brick Quantity	3,000	5,100	5,100	3,000	3,000	5,100	5,100	3,000	3,000	5,100
Bond Height	4,000	4,000	4,000	4,000	4,000	4,000	4,000	4,000	4,000	4,000

subjective judgment of each engineer. When consistency is not secured, an engineer has to verify associated work items and assign the unit costs based on subjective decision-making.

In comparison, the ontological approach always provides a single answer to determine the brick work items by applying pre-defined rules. In this case, the proposed ontological approach made the same decision (BondType = 1, FinishingType = 3, CarryingType = 2), since the rules of "necessary & sufficient" were defined as the same for BWCO and BWIO. Consequently, the possibility for subjective judgment decreased. Moreover, the reliability of the result dramatically increased and the potential for error decreased.

5. Conclusions

BIM-based construction management has begun to expand, and as the number of practices that use BIM-based information management has significantly increased, the construction industry has augmented its use of IFC or IFCXML during projects. However, there are several problems with the use of BIM data in cost estimation. The existing practice and research into BIM-based cost estimation focuses on quantity take-off of materials. Current BIM-based cost-estimation software does not provide any information on work items related to the materials in the BOQ. As such, engineers still need to verify associated work items and assign unit costs based on subjective decision-making.

Therefore, this research presented an automated method for selecting the most appropriate work item and matching it to a material of a building element using semantic technology. We proposed an ontological inference to automate inference of work items that is appropriate for cost estimation. The ontological inference consists of three layers. First, on the work conditions conversion layer, BIM data such as work conditions are extracted from a building model and converted into RDF data in a machine-understandable format. Second, on the ontologies management layer, two ontologies (WCO and WIO) are logically combined to infer the work item using the types of work conditions. Finally, the reasoning layer creates inferred work item data that includes work items by means of a reasoning process that is based on two ontologies (BWCO and BWIO) and the RDF data of work conditions.

We conducted a case study that considered brickwork, and an ontology was established based on brick thickness, bond type and usage context such as room usage, type, location, height, length and thickness of building elements, brick type, and finishing number. This ontology contains semantic information on brick work items and brick work conditions as well as a semantic reasoning rule that activates the ontology.

The validation results show that the proposed ontologies provide a well-established relationship between work condition and work item. In addition, the proposed ontological approach can increase consistency and reduce errors in estimator results. This approach provides an objective methodology for cost estimation, as the work condition is understood semantically by the system.

There are some assumptions to apply to the proposed approach as following: 1) the building model must be modeling with specific data input about work conditions such as, space, building elements, and material data, 2) naming space and material includes the usage of space and the specific type of material, 3) composite materials should be modeled in an element, and 4) the property data of a work item should contain well-organized required data for the cost estimation process. Moreover, to ensure the range of capacity extends beyond brickwork, the work conditions and the properties of the work items are defined considering practical considerations when adopting the proposed approach.

This work contributes by eliminating subjective decision-making via search of appropriate work items for cost estimation and the use of BIM data extracted from IFCXML. The proposed ontological approach to build a cost estimation will assist engineers in more readily using BIM data from IFCXML and will be helpful in automation of the whole estimation process.

This research has some limitations. Since a work item that consists of materials type and construction method depends on specific work and project environments, we need to develop ontologies that consider all components of work and project environments so as to capture actual work conditions that were not presented as work conditions in this paper, such as buildability and economic feasibility. In addition, since the type of work item can be considered in terms other work items, the ontologies of each specific work item should be expanded to consider all work items. These limitations should be addressed in future studies.

Acknowledgements

The present research has been conducted by the research grant of Kwangwoon University in 2013.

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (NRF-2014R1A2A2A05006437).

References

AECCafe (2002). *Timberline CAD integrator uses IFC standard to create cost estimates directly from CAD*, http://na.sage.com/us/sage-construction-and-real-estate/sage-300-constr uction-and-real-estate.

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P. (2003). *The description logic handbook: Theory, implementation and applications,* Cambridge university press, Cambridge, UK.

Benevolenskiy, A., Roos, K., Katranuschkov, P., and Scherer, R. J. (2012). "Construction processes configuration using process patterns." *Advanced Engineering Informatics*, Vol. 26, No. 4, pp. 727-736, DOI: 10.1016/j.aei.2012.04.003.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). "The semantic web." *Scientific American Magazine*, May.

Chau, K. W. (2007). "An ontology-based knowledge management system for flow and water quality modeling." *Advances in Engineering Software*, Vol. 38, pp. 172-181, DOI: 10.1016/j.advengsoft.2006.07.003.

Commonwealth Scientific and Industrial Research Organization (2005). *Semantic web-based open engineering platform*, Project Co-funded by the European Commission within the Sixth Framework Programme (2002-2006), STRP NMP2-CT-2005-016972.

Cost, U. S. (2010). *Success design exchange*, http://www.uscost.com/designexchange.asp.

Eastman, C., Teicholz, P., Sacks, R., and Liston, K. (2008). *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers, and contractor*, John Wiley & Sons, New Jersey.

Ei-Diraby, T. and Osman, H. (2011). "A domain ontology for construction concepts in urban infrastructure products." *Automation in Construction*, Vol. 20, No. 8, pp. 1120-1132, DOI: 10.1016/j.autcon.2011.04.014.

Ei-Diraby, T., Fies, B., and Lima, C. (2003). *An ontology for construction knowledge management*, 31st Annual conference of the Canadian Society for Civil Engineering, GCM-322.

Fu, C., Ghassan, A., Amanda, M.P., Lee, A., and Wu, S. (2004). "IFC implementation in lifecycle costing." *Journal of Harbin Institute of Technology*, Vol. 11, No. 4, pp. 437-441, Article ID : 100529113 (2004) 0420437205.

Gruber, T. (1993). "A translation approach to portable ontology specification." *Journal of Knowledge Acquisition,* Vol. 5 No. 2, pp. 199-200, DOI: 10.1.1.101.7493.

Halpin, D. W. and Woodhead, R. W. (1998). "Construction Management. 2nd edition". Wiley, New York.

Hartmann, T., van Meerveld, H. J., Vossebeld, N., and Adriaanse, A. M. (2012). "Aligning building information model tools and construction management methods." *Automation in Construction,* Vol. 22, pp. 605-613, DOI: 10.1016/j.autcon.2011.12.011.

IAI (1999). *AEC/FM Processes supported by IFC,* International Alliance of Interoperability, Oakton, Virginia.

Innovaya (2010). *Innovaya visual estimatin*, http://www.innovaya. com/prod_ve.htm.

Jang, M. and Sohn, J. (2004). "Bossam: An extended rule engine for OWL inferencing." *Proceedings of Rules and Rule Markup Languages for the Semantic Web, Lecture Notes in Computer Science*, Vol. 3323, pp. 128-138.

Kim, H., Stumpf, A., and Kim, W. (2011). "Analysis of an energy efficient building design through data mining approach." *Automation in Construction*, Vol. 20, pp. 37-43, DOI: 10.1016/j.autcon.2010. 07.006.

Klyne, G., Carroll, J., Klyne, G., and Carroll, J. (2004). *Resource Description Framework (RDF): Concepts and abstract syntax*, W3C recommendation, http://www.w3.org/TR/rdf-concepts/ 2004.

Kristian, B. S., Per, C., and Kjeld, S. (2010). "Ontologies to support RFID-based link between virtual models and construction components." *Computer-Aided Civil and Infrastructure Engineering*, Vol. 25, No. 4, pp. 285-302, DOI: 10.1111/j.1467-8667.2009.00638.x

Kwon, O. C., Jo, C. W., and Cho, J. W. (2011). "Introduction of BIM quality standard for quantity take-off." *Journal of the Korea Institute of Building Construction*, Vol. 11, No. 2, pp. 171-180, DOI: 10.5345/JKIC.2011.04.2.171.

Lee, J. and Jeong, Y. (2012). "User-centric knowledge representations based on ontology for AEC design collaboration." *Computer-Aided Design*, Vol. 44, pp. 735-748, DOI:10.1016/j.cad.2012.03.011.

Lee, M. C., Jang, K. N., and Oak, J. H. (2009). "A study on the modeling of BIM property information in reflection of the public cost estimate distinctions." *Proceeding of Korea Institute of Construction Engineering and Management 2009*, Vol. 1, pp. 772-777.

Liebich, T. (2009). *IFC 2x edition 3, model implementation guide version 2.0,* BuildingSMART International Modeling Support Group.

Ma, Z., Wei, Z., Zhang, X., Qiu, S., and Wang, P. (2011). "Intelligent generation of bill of quantity from IFC subject to chinese standard." *Proceedings of the 28th ISARC*, Vol. 1, Seoul, Korea, pp. 740-745.

Ma, Z. L., Zhang, X. D., Wu, S., Wei, Z. H., and Lou, Z. (2010). "Framework design for BIM-based construction cost estimating software." *Proceedings of the CIB W78 2010:27th International Conference,* Cairo, Egypyt, Vol. 1, pp. 1-7.

Park, C.-S., Lee, D.-Y., Kwon, O.-S., and Wang, X. (2013). "A framework for proactive construction defect management using BIM, augmented reality and ontology-based data collection template." *Automation in Construction*, Vol. 33, pp. 61-71, DOI: 10.1016/j.autcon.2012.09.010.

Sage (2010). *Success design exchange*, http://www.sagecre.com/products/timberline_office/estimating/estimating_extended_timberliti_office_estimating_software.

Schevers, H. and Drogemuller, R. (2006). "Converting the industry foundation classes to the web ontology language." *Proceedings of the First International Conference on Semantics, Knowledge, and Grid (SKG 2005)*, Vol. 1, Beijing, China, pp. 73-75.

Staub-French, S., Fischer, M., Kunz, J., and Paulson, B. (2003). "A generic feature-driven activity-based cost estimation process." *Advanced Engineering Informatics*, Vol. 17, No. 1, pp. 23-39, DOI: 10.1016/S1474-0346(03)00017-X.

Taormina, R., Chau, K., and Sethi, R. (2012). "Artificial neural network simulation of hourly groundwater levels in a coastal aquifer system of the venice lagoon." *Engineering Applications of Artificial Intelligence*, Vol. 25, pp. 1670-1676, DOI: 10.1016/j.engappai.2012.02.009.

The Apache Software Foundation (2014). *Jena ontology API*, https://jena.apache.org/documentation/ontology/.

Tokmo (2009). *Tokmo production system*, http://www.tokmo.com/product.html.

Tserng, H., Yin, S., Dzeng, R., Wou, B., Tsai, M., and Chen, W. (2009). "A study of ontology-based risk management framework of construction projects through project life cycle." *Journal of Automation in Construction*, Vol. 18, No. 7, pp. 994-1008, DOI: 10.1016/j.autcon.2009.05.005.

Vico (2010). *Vico estimator 2009*, http://www.vicosoftware.com/products/vico-estimator-2009/tabid/85114/ Default.aspx.

Winest (2010). *Winest DesignEst Pro*, http://www.winest.com/products/add-ons/designestpro.aspx.

Zhang, J. and Chau, K. (2009). "Multilayer ensemble pruning via novel multi-sub-swarm particle swarm optimization." *Journal of Universal Computer Science*, Vol. 15, No. 4, pp. 840-858, DOI: 10.3217/jucs-015-04-0840.