Automation in Construction 68 (2016) 102–113

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

A linked data system framework for sharing construction defect information using ontologies and BIM environments

Do-Yeop Lee [a], Hung-lin Chi [c], Jun Wang [c], Xiangyu Wang [b,c], Chan-Sik Park [a,*]

[a] School of Architecture and Building Science, Chung-Ang University, South Korea
[b] Department of Housing and Interior Design, Kyung Hee University, South Korea
[c] Australasian Joint Research Centre for Building Information Modelling, Curtin University, Australia

ARTICLE INFO

Article history:
Received 19 March 2015
Received in revised form 12 April 2016
Accepted 1 May 2016
Available online 20 May 2016

Keywords:
Defect management
Linked data
Ontology
Building information modeling
SPARQL query

ABSTRACT

Defect data contains knowledge about specific work conditions. In order to prevent reoccurrence of defects, a data feedback mechanism is required. However, most defect data are stored in unstructured ways, resulting in the fundamental problem of data utilization. This paper proposes a novel framework by using BIM and linked data technologies for sharing defect data between heterogeneous data sources in a new way. To demonstrate, a defect ontology is developed, work context information is extracted from BIM models, extracted BIM data is converted to RDF format, and SPARQL queries are implemented. The proposed approach could help BIM software applications to take into account information stemming from the defect management domain. Also, it can reduce data search time and improve the accuracy of search results as well. Therefore, this framework may enable reductions of defect occurrence and improvements in current defect management practices.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Construction defects are perceived as one of the primary causes of low project productivity, resulting in delays, additional costs, materials, and manpower for defect rectification [1,2]. In order to prevent the reoccurrence of defects, various researches have been conducted, focusing on identifying defect causations [3,4], and analyzing cost–time impacts [5,6]. The results of these research efforts provide valuable statistical information for decision-making, allowing project stakeholders to evaluate their enterprise performance, and identify the factors influencing defect occurrence from a holistic perspective. For instance, 'Poor planning and coordination of resources' was ranked first among site management causes of rework [3] and 'Structural steel' had the most significant impact on the defect cost among subcontractor trades [6]. Such information only provides generalized concepts or terminologies at a high level of abstraction. Therefore, practitioners cannot make a proper decision for defect prevention in practical situations by using statistical information alone without access to actual defect cases that include complex information of specific design/construction tasks. In order to address this issue, it is necessary to look into previous research on data feedback for defect prevention. Palaneeswaran [7] argues that the analysis of defect causation should focus on developing a suitable knowledge feedback mechanism. Chong and Low [8] point out that many defects continue to occur repeatedly as designers fail to obtain important feedback about defects and suggest developing a database system by using existing knowledge. Jang and Seo [9] emphasize the necessity of timely data feedback through a case study on apartment defects. For instance, a window condensation defect had occurred in a previous project and then the same defect reoccurred in their next two similar projects due to the lack of appropriate information feedback. However, there has been little research on the data structure required for implementing actual data feedback systems. Moreover, traditional database systems and keyword-based searches appear to be insufficient to improve information discovery and retrieval [10]. Therefore, an appropriate information platform should be considered to enable users to easily search for individual defect cases and directly access them.

In order to develop this platform, the flow of defect data should be closely linked from the perspective of knowledge management processes. However, the utilization of defect data in practice is very low, due to characteristics such as lack of formal structure in data representations, insufficient data input, only text-based search available, and insufficient sharing of defect data. These characteristics will be discussed in detail in Section 2. The features of defect data could be improved by taking advantages of state-of-the-art technologies. In information systems, ontologies are considered as ideal formal tools to represent domain knowledge, as they allow modeling concepts with semantic relationships. In the context of the Semantic Web, ontologies play an important role for publishing and connecting structured data on the Web known as linked data [11]. In the architecture engineering and construction (AEC) domain, building information models (BIM)

* *Corresponding author.
E-mail address: cpark@cau.ac.kr (C.-S. Park).

http://dx.doi.org/10.1016/j.autcon.2016.05.003
0926-5805/© 2016 Elsevier B.V. All rights reserved.

have been utilized as central repositories of building data to facilitate the exchange and interoperability of information in digital format across the project lifecycle. However, within the wider AEC context, BIM models are only single silos of information [12]. In this regard, a defect linked data framework is proposed underlying the ideas developed in the authors' previous paper [13]. The target of this framework is to convert defect data to the ontology-based linked data format for linking and searching defect data between different data sources. To develop this framework, various technical issues are addressed in this paper.

1) Development of a defect ontology for publishing defect data following linked data principles
2) Extraction of work context information from BIM models for generating machine-readable defect cases
3) Conversion of extracted BIM data to the RDF data model
4) Search of relevant defect cases by using SPARQL queries

The proposed approach could help BIM software applications to take into account information stemming from the defect management domain. Also, it could reduce data search time and improve the accuracy of search results as well. Improving the process of utilizing defect information in design and construction phases can prevent the same defect or error occurrence. In addition, this approach can provide a technical solution towards the usage of valuable data stored in public websites or corporate repositories as data silos.

This paper is organized as follows. Section 2 discusses the current obstacles for sharing defect data. In Section 3, related research on BIM, ontology, and linked data is reviewed in order to find applicable solutions to solve the limitations discussed in Section 2. In Section 4, actual defect cases are analyzed to build a defect ontology. Next, a linked data system framework is proposed and tested in Section 5. Finally, the paper concludes by discussing findings and issues to be addressed in future work.

2. Issues of utilizing defect information

Information regarding defects which are identified during a project lifecycle is stored in various ways. However, because of the lack of structure in the defect data and its various characteristics, the data cannot be used for retrieving related information for a given situation during the design and construction phase. This section discusses the problems and issues to be addressed for defect data reusability through comprehensive literature reviews and usage analysis of defect data.

2.1. Lack of formal structure in defect data representations

With the information-intensive nature of the construction industry, large amounts of data are produced throughout project tasks and usually stored in cooperation's knowledge management systems. However, most data are scattered across various company systems and recorded in unstructured formats. Unstructured data is data that does not follow a machine-readable format, so, there is no reliable way to access, analyze, or search for desired information. In addition, it is difficult to generate valuable knowledge through the vast amounts of existing data.

Most defect data are typically available in unstructured formats. The number of defects that occur in a project typically goes in the thousands, including minor and major defects [14]. Such defect data is usually modified and published in defect case books or stored as individual data files for sharing knowledge. However, these files exist as unstructured text-based data, such as PDF or Word documents, spreadsheets, HTML pages, and so on. The contents of defect cases normally consist of the title, description, picture, diagram, defect cause, prevention method, etc. However, the stored contents differ from case to case. Even if defect cases are stored using a machine-readable format, users cannot find and analyze them without an appropriate data collection structure which describes defect case situations in a standard way. Therefore, in order to retrieve and analyze defect data effectively, a comprehensive defect data collection template is needed and should include various types of contents for multiple purposes [13].

First of all, construction context information provides important clues when searching for defect cases similar to the situations that architects and constructors are working on. Context can be defined as 'situations in which a construction entity acts or exists; whereby an entity can be a person involved in construction, an action taken to complete a job, a component built in a project, or a resource utilized to perform the action or build the component' [10]. In the case of defects, the context information that needs to be stored includes construction method, space, component, material, etc. In addition, defect analysis data such as defect root cause, defect type, impact cost and time needs to be input in order to assist further analysis. By combining context and analysis information, it is possible to provide valuable knowledge to project stakeholders both at the specific task level for retrieving relevant data and at the cooperation strategy level for analyzing data. Finally, to easily understand the situation around which a defect occurred, descriptive information from materials such as 2D plans, section drawings, pictures, and specifications is necessary. Therefore, defect data should be recorded using a machine-readable format and should include various types of information for comprehensively explaining defect cases with a standardized structure.

2.2. Insufficient data input

As discussed previously, different types of detailed context information should be recorded for generating structured defect data and improving usability and search accuracy of defect data. When looking at the contents of defect case books, most context information is included in each case. However, this context information is expressed textually in the title and defect description section. Hence, it is only human-readable but not machine-readable. To make the recorded contents machine-readable, defect data collection structures have been proposed by using codes or industry standard classification systems [13,15]. However, entering data manually by following codes (e.g., painting 9915 and internal walls 75) as suggested in [15] or standard classifications (e.g., cement stucco 22-092423 and wall finishes 21-032010) in [13] is labor-intensive and erodes the quality manager's work efficiency.

In current practice, defect data is generated after defects are identified and need to be rectified on-site. Defect data modification only occurs for data sharing in the office. These two phases (defect data rectification and modification) take place over different time frames, they require different information, and they are handled by different parties. First, in the defect identification and rectification process, the required information is commonly a location and identified defect situation, for instance, 'Water leak in the toilet on the 3rd floor'. It is transferred to the sub-contractor by the quality manager who orders defect rectification. At this point, detailed context information is unnecessary because stakeholders involved in a project already have related information through drawings and specifications. Also, defect analysis information is not recorded in this phase because the defect management process focuses on defect identification and rectification rather than data reuse. Next, the process of defect modification is that, after defect rectification work, important cases are selected and then modified for information sharing. If recorded contents are insufficient for the reasons discussed above, additional efforts are required for finding context and analysis data in order to modify and analyze defect cases. If it is impossible, because of a time lag between the two processes, the reusability of defect data would decrease. For that reason, in most research, defect data is manually classified and reformed to enable useful data for research purposes [4,5,8].

In summary, the current defect management process faces a trade-off relationship between task efficiency and the level of data quality. Therefore, to solve this problem, the context information needs to be

automatically recorded to ensure work efficiency and also allow the reuse of defect cases during the defect rectification phase.

2.3. Only text-based search available

The ways to find relevant information from existing defect data are quite limited. In order to find some information from a defect case book, it is necessary to look through the entire contents because they are typically categorized according to construction work method, space, defect type, etc. For instance, in a chapter classified according to water-leakage type defects, various construction work methods related to water-leakage are mixed. Also, if the data is stored as an HTML or document file, it can be accessed by through a keyword search, based on the text in the case title and description. But not all the context information is included in the individual cases. In addition, the context information is written manually and subjectively without a standardized vocabulary. Although keyword-based searching is the most common searching method, it is troublesome. If the keyword is too specific, the search results exclude documents that would have been of interest. On the contrary, if the keyword is too generic, the search results include too many irrelevant documents [10]. Even if the data is stored in a well-classified and structured format, various keywords should be input to get the desired information.

Moreover, the presence of synonyms also causes critical problems. The difference between terms having the same meaning can lead to inaccurate search results (for example, toilet and restroom, elevator and lift). In addition, even the industry standard classification systems for improving data interoperability use different terms (for instance, Metal Walkways (Code: B1080.70) in the UniFormat 2010 and Metal Catwalks (Code: 05 51 36.13) in the MasterFormat 2004). In conclusion, improved mechanisms for defect data searches are needed and they should take into account the existence of synonyms.

2.4. Insufficient sharing of defect data

The construction industry has become more diverse and complex and has adopted new technologies and materials to meet growing demands from owners and customers. These complex circumstances can result in unexpected defects when unfamiliar technologies or materials are used (e.g., physical connections of different types of materials or environmental factors of exterior and interior spaces). No doubt, there are now a lot of uncertainties compared with typical repetitive projects of the past. The number of projects performed by a company is too limited for an individual to experience the various defects that can occur due to various causes. As discussed in Section 1, even commonly known defects, which are encountered in previous projects within a company, are hard to share across ongoing projects.

In general, defects are identified at various stages of the project lifecycle through activities such as design reviews, constructability checks, quality inspections, maintenance, etc. Also, these tasks are dealt with by different participants such as architects, contractors, sub-contractors, facility managers, etc., with the defect data stored in each organization's repository. Therefore, if it is possible to share defect data occurring under different conditions, construction methods, regions, and weather types, it might also be possible to integrate defect data that is scattered across different sources or data silos.

Data sharing is ideal. However, in practice, defects rarely surface to the public, unless they are considered serious enough [8], because many companies and individuals have a reluctance to expose their errors due to commercial pressure or personal esteem [15]. This situation contrasts with the web community, where experts working in the same field discuss various defect cases and are willing to share their experiences and knowledge. Therefore, in order to improve the sharing of defect data, an appropriate technology platform is required along with cultural changes regarding the openness of defect data.

3. BIM, ontologies, and linked data approaches in construction

3.1. Overview of linked data approaches

The Semantic Web is an extension of the current Web. The Semantic Web is not just about putting data on the Web. It is about making links so that a person or machine can explore the web of data. According to the W3C, the Semantic Web is a web of data to provide a common framework that allows data to be shared and reused across applications, enterprises, and community boundaries [16]. Linked data is defined to describe a recommended best practice for publishing and connecting structured data as RDF on the web [11].

Linked data is based on four basic principles [17]:

1) Use URIs as names for things
2) Use HTTP URIs so that people can look up those names
3) When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)
4) Include links to other URIs so that they can discover more things

Linked data offers significant advantages over current practices [18]. By using HTTP URIs as globally unique identifiers for data items as well as for vocabulary terms, the RDF data model is inherently designed for being used on a global scale, enabling anybody to refer to anything. Clients can look up any URI in an RDF graph over the web to retrieve additional information. Thus, each RDF triple is part of the global web of data, and each RDF triple can be used as a starting point to explore this data space. Linked data publishers may also provide RDF dataset dumps for local replication of data, and SPARQL endpoints for querying the data directly.

Resources including media and scientific publications in various domains have been described using linked data techniques. For instance, in the public sector, the Open Government Data (OGD) movement has emerged to promise government transparency and to meet diverse needs from data consumers by releasing government data on the web [19]. However, OGD raw datasets are typically published in heterogeneous formats, which create interoperability and usability problems. To overcome these limitations, linked open government data (LOGD) is emerging based on linked data technologies [20].

3.2. BIM, ontologies, and linked data in construction area

Ontologies represent knowledge in specific domains and enable semantic interoperability by linking to other external data sources. They have the potential to address the limitations found earlier in sharing and reusing unstructured construction information. Thus, many different concepts and approaches have been tried and implemented, such as the representation of construction domain knowledge, extraction or conversion of semantic data from BIM, connection to external data sources, etc.

The e-COGNOS is a research project aiming to develop construction domain ontologies. It provides a portal documentation and supports consistent knowledge representation and organizational issues regarding access and use of knowledge at any level (project, corporate, domain) [21]. The InteliGrid project is a world-leading project to provide an interoperability platform focusing on semantic collaboration among businesses in the European large-scale engineering industries [22]. Some works have also focused on developing specific ontologies at the task level. For example, Niu and Issa [23] proposed a claim ontology to share comprehensive and formal claim knowledge with all the participants within the claim workflow. Fidan et al. [24] presented a risk ontology to predict and estimate cost overruns. Their ongoing research aims to develop an ontology-based database for capturing risk event histories to be used for the prediction of cost overruns.

Making an agreement and commitment to use the same terms in the same way for a domain of interest is one of the ontology principles. The definition of ontology is ‘a formal, explicit specification of a shared

conceptualization' [25]. In the specification and conceptualization stage, concepts are organized into a superclass–subclass hierarchical structure, which is also known as a tree structure. The backbone of ontology is often a hierarchical tree structure. A variety of taxonomies have been developed in the construction sector following a tree structure as well, ranging from domain dictionaries to specialized taxonomies such as bcXML, IFC, Omniclass, etc. [26]. In the e-COGNOS project, the IFC have provided the seeds of the e-COGNOS ontology, and a domain taxonomy for construction terms was developed and mapped to IFC concepts. It will allow any future expansion in IFC to be dynamically mapped to the e-COGNOS taxonomy [27]. On the other hand, in the case of developing a risk ontology, Fidan et al. [24] have developed a common vocabulary and an ontological structure to explain interrelations between risk sources, risk events, vulnerability, and their consequences through comprehensive literature reviews and expert interviews.

Ontologies provide means to semantically represent domain knowledge with great expressive capacity through defining concepts, relationships, and axioms which enable query and reasoning under a variety of conditions. This characteristic of ontologies could solve the interoperability problem between the applications deployed by AEC stakeholders. In this context, various research efforts have focused on transforming IFC into RDF or OWL. Pauwels et al. [28] have investigated the applicability of semantic web technologies for improving interoperability by using rules, querying, and reasoning. Beetz et al. [29] have introduced a method to convert an EXPRESS schema into a logically rigid and semantically enhanced ontological level encoded in the OWL.

Some research efforts have focused on extracting specific information from BIM data using semantic web technologies and domain-specific ontologies. Nepal et al. [30] proposed ontology-based feature modeling for information extraction from BIM using ifcXML and query language. For example, opening, penetration, and component intersection data is extracted. Demir et al. [31] have extracted the instance of 'exterior wall' from an ifcOWL building model by using SPARQL mapping and query functions.

The next research area is ontology-based information extraction from BIM for performing specific construction tasks. Lee et al. [32] proposed the automated inference of standard work items to mitigate the problem of cost estimator's subjectivity in searching for an appropriate work item. In this research, work condition information such as tile size, thickness, construction method and joint width is extracted from IFCXML. Next, this data is converted to RDF to infer the most appropriate work item using a semantic reasoning system. Pauwels et al. [33] deployed the N3-logic semantic rule language to check compliance with acoustic performance regulations. A BIM model was developed with explicit building information, such as element and façade surface area, and then acoustic calculations could be automatically inferred after providing the rule engine with the RDF graph and the required rule set.

On the other hand, some studies have focused on developing ontologies for using external data to perform a specific task without utilizing BIM data. Zhong et al. [34] proposed the ontological approach for checking construction quality regulation compliance. Based on regulation documents, construction process constraints, and inspection task constraints are modeled into OWL axioms and SWRL rules, in order to ensure activities follow the right sequences and the necessary inspection tasks take place at the right time. In the construction safety domain, Wang et al. [10] used ontologies to structure the job hazard analysis document for identifying safety rules applicable to a given context, such as activities, job steps, site conditions, and potential hazards. In the sustainable building technologies area, many suppliers introduce thousands of their products via the web. To help users in locating specific products, Tah and Abanda [35] developed photovoltaic products ontology to support the right decision-making by using description logic query.

Finally, the research efforts considering combining linked data with BIM technologies are reviewed. Curry et al. [36] proposed a linked data system to link BIM environments with external data for energy usage tracking. For example, an IFC entity (IfcSpace) is extracted from BIM environments and mapped to an RDF entity (Room), and then this RDF entity is linked to energy sensor information (Energy consumption). Costa and Madrazo [37] have presented the application of linked data technologies to integrate BIM models with a web-based catalog of structural precast concrete components.

3.3. Challenges of combining BIM and linked data technologies

While ontology researches are at an early stage, this technology is being applied in various construction management areas, and the research efforts provide insights in the representation of domain knowledge or the conversion of existing data semantically for data interoperability, reasoning, and querying. However, little effort has been put into linking external sources with BIM data by extracting BIM object information. Also, literature reviews showed that no study attempts to utilize defect data as linked data, which is to be addressed in this paper.

The various characteristics holding back the efficient reuse of defect data identified in Section 2 can be addressed by utilizing BIM technology and ontologies. First, the lack of formal structure in defect data representations can be addressed by adopting the defect collection template for standardizing and analyzing defect data and by developing the defect ontology for representing defect data semantically. Second, the insufficient data input can be improved by extracting the context information from a BIM model based on an object or space at the defect occurrence phase and then this information can be converted into RDF format based on the defect ontology. Through this approach, computer-readable data can be automatically generated. Third, the lack of alternatives to text-based searches can be improved by extracting context information from a BIM model for running SPARQL queries. Also, the synonym problem in data searching can be solved by adopting an industry standard classification system such as Omniclass, which is developed for data interoperability in the construction industry. Finally, if enterprises or individuals could publish the converted RDF data on the web, the insufficient exchange of defect data can be addressed.

4. Defect data analysis and categorization

This section documents the results of defect data analysis and categorization. The purpose of defect data analysis is to reconfirm that defect cases should be provided to different persons at different times and that they should be treated using different management methods depending on the underlying cause. The purpose of defect data categorization is to allow the definition of classes and relationships for developing a defect ontology. Defect cases were collected from defect casebooks, which were published by an enterprise from 2010 to 2012. In total, 180 defect cases related to waterproofing were selected, after which data that did not contain sufficient information for categorization was removed. Finally, 110 defect cases were utilized in this section.

4.1. Defect data analysis

For the analysis of defect data, defect cause classifications proposed in [38,39] were adopted with some modifications. The analyzed results were classified into 'Phase', 'Responsibility', and 'Defect cause'. 'Phase' shows the time of an initial cause generation that induces a defect and 'Responsibility' indicates a participant who needs to receive defect case information. 'Defect cause' provides information on how to manage a defect. The result of defect data analysis is shown in Table 1.

Most defect cases were found during the occupancy phase, except some cases found during the construction phase such as 'Lack of constructability (4%)' and 'Work interference (4%)'. This means that the designer or constructor was not aware of any errors that were generated during the design and construction phase. The 'Responsibility'

Table 1
Result of defect data analysis.

Phase	Responsibility	Defect cause	Num.	%
Design	Architect	Inappropriate construction methods	20	18%
		Inaccurate design	16	15%
		Design omission	5	5%
		Lack of constructability	4	4%
Construction	Manager	Improper construction plan	8	7%
		Work interference	4	4%
		Inadequate supervisory skills	2	2%
	Worker	Lack of the workmanship	31	28%
		Non-compliance to standards	12	11%
		Omissions of some activity of task	8	7%
		Total	110	100%

analysis showed that almost half of the defects were caused by the architectural designer and appeared during occupancy ('Inappropriate construction methods (18%)', 'Inaccurate design (15%)', and 'Design omission (5%)'). This analysis result is similar to those found in a study on latent defects conducted by Chong and Low [8]. According to their research, most latent defects appear only during the occupancy stage and are difficult to detect because any faults committed by designers would not usually appear during construction. Also, many latent defects were repeated in many of the buildings and designers were not aware of them. For this reason, these defect cases should be delivered to architects and latent defects need to be eliminated beforehand. However, within the current data management environment, transferring latent defects data to the architect is quite difficult because constructor and property managers retain such latent defect information, not the architect. Actually, the defect casebooks used in this research were published by a construction company for in-house use, and it was difficult for architects to access this defect information.

On the other hand, during construction, the defects caused by the site manager are due to 'Improper construction planning (7%)', 'Work interference (4%)', and 'Inadequate supervisory skills (2%)', while the defects caused by workers are due to 'Lack of the workmanship (28%)', 'Non-compliance to standards (12%)', and 'Omissions of some activity of task (7%)'. Depending on the cause and responsibility, the ways in which defects are managed are different. Therefore, the quality manager should understand both analyzed information and actual defect cases to make a proper management plan before construction. From a defect cost perspective, according to Love and Irani [6], defects which occurred during construction incur an unnecessary cost to every participant involved in a project. Defect information should be fully exploited not only by constructors but also by sub-contractors. In summary, the defect data analysis has reconfirmed that defect data should be provided at the right time to the right parties.

4.2. Defect data categorization

A total of 110 defect cases were categorized according to the defect data collection template proposed in our previous research [13]. The various contents of this template were developed to meet various participants' requirements for searching and analyzing defect data through comprehensive literature reviews. Omniclass was adopted for data interoperability, with the details described in [13]. The example case is about a urethane peeling failure in an underground parking area, which is illustrated in Fig. 1. In this paper, the contents of this template were divided into three parts, as indicated in Fig. 1 (left).

The first part of the document lists contextual information, which may eventually enhance search accuracy. Context information

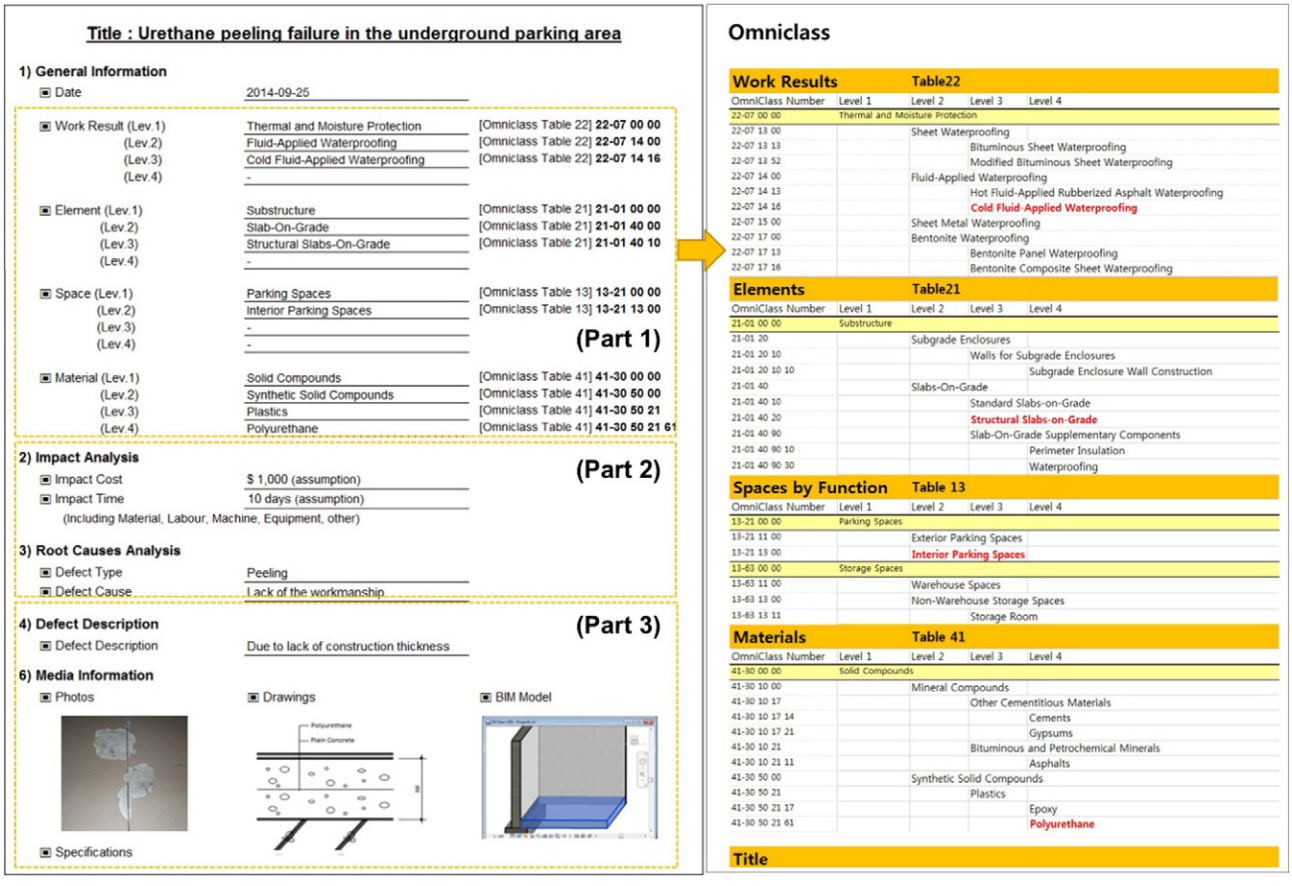

Fig. 1. Example case of defect data collection template with Omniclass hierarchy.

includes 'Work_Result', 'Element', 'Space' and 'Material'. Each category consists of several levels of detail by following the Omniclass classification system. This context information can be extracted from BIM data. It can include more detailed information for describing defect situations with standardized terms, which are difficult to express in existing defect case documents. For example, the 'Work Result' of urethane waterproofing method can be found in Omniclass table 22 by following 'Thermal and Moisture Protection' (Level 1), 'Fluid-Applied Waterproofing' (Level 2), and 'Cold Fluid-Applied Waterproofing' (Level 3).

The second part includes analysis information for further analyzing defect impact and identifying management factors. Analysis information includes 'Impact cost', 'Impact time', 'Defect type', and 'Defect cause'. Unlike context information, analysis information should be input manually after completing defect rectification tasks. In this example, the defect type is a peeling problem on the surface, and the defect cause can be categorized as a lack of workmanship caused by the worker. The impact cost and time were assumed for research purposes.

The third part includes other relevant information for understanding the specific defect occurrence situations such as defect description, defect photo, as-plan drawings, BIM model, specifications, etc. In light of data quality, this information is very important. Especially, the 2D section drawing and specification help participants to recognize the faulty parts by comparing their current situations with defect cases. Most information exists as electronic documents or in BIM data. Hence, it is possible to store this information automatically. However, this issue is not within the scope of this research.

5. Ontology-based defect data

5.1. Framework of the ontology-based defect-linked data system

Establishing a defect data feedback mechanism is essential to prevent defect reoccurrences. To achieve this, a series of knowledge management processes should be integrated systematically. If any of them is not working properly, it is hard to expect the desired result. Therefore, in order to link the disconnected information flow, a comprehensive framework is proposed by adopting BIM and linked data technologies. However, as this framework is based on ongoing research, some processes expressed by the dashed lines are out of this paper's scope. The overall process is illustrated in Fig. 2.

Developing a standard defect data structure serves as the basis to facilitate other system features. To develop the structure, a defect ontology has been proposed based on analyzed findings in the previous section, and it has been developed with the protégé ontology authoring tool. This process is shown as (1) defect ontology development in Fig. 2. The efficiency of generating structured data has been improved by using BIM data extraction and RDF conversion processes. To do this, the BIM model has been created to extract context information based on the selected component. This context information is extracted and exported in a spreadsheet by using embedded functions in BIM tools. The information in the spreadsheet is then converted to RDF format by using an RDF converter. These processes are illustrated as (2) context information extraction and (3) RDF conversion in Fig. 2. The data search functions have been tested through the query functionality in the ontology editing tool by using input defect cases as instances. This process is shown as (4) query and results in Fig. 2.

Although the rest of the processes are out of the scope of this paper, they are required for implementing the entirety of the framework. To utilize this framework properly, it should be connected with the field management practice, in order to better prevent and rectify defects. When checking possible defects after design or before construction, the context information extracted from a BIM environment should be used as query information to retrieve similar defect cases that have been published on the web. When rectifying the identified defects during the construction or occupancy stage, the defect data collection template and the extracted context information from a BIM environment should be indicated in existing defect management systems, such as in [40], to allow capturing context information automatically and converting it into RDF format that can be published as linked data. The development of a BIM API or add-on program could make these two processes possible. Efficient data publishing and search methods are also required to maximize data sharing. To implement this, generated RDF data should be easily published in the RDF store and then it would be accessible using a SPARQL endpoint.

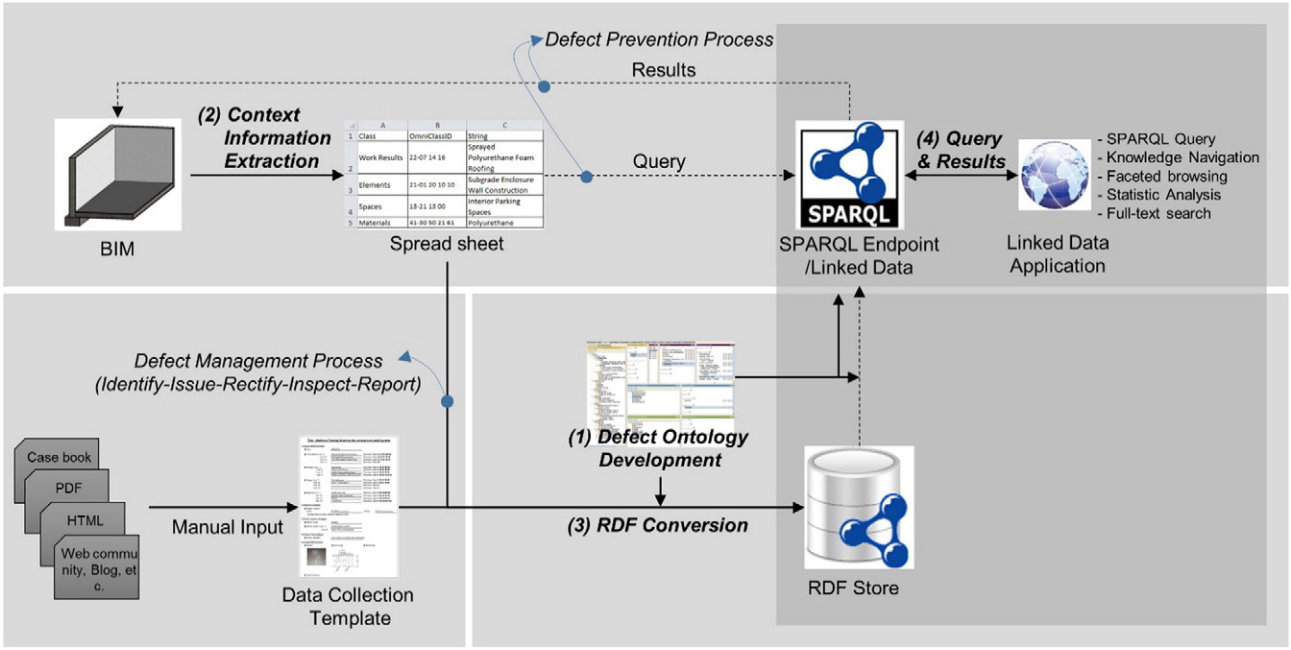

Fig. 2. The framework of defect-linked data system.

5.2. Ontology development

The defect ontology has been built based on the analysis and categorization in Section 4 using Protégé 4.3 ontology editing tool. The defect ontology is illustrated in Figs. 3 and 4, consisting of six classes, six object properties, and seven data properties, which represent the main content of the defect data collection template.

The classes *Work_Result*, *Space*, *Element*, and *Material* capture the context information to be extracted from BIM. Each of these classes has several subclasses, thereby following the Omniclass hierarchy. For instance, *Concrete* and *Thermal_Moisture_Protection* are subclasses of *Work_Result* and *Sheet_Waterproofing*, and *Bentonite_Waterproofing* are subclasses of *Thermal_Moisture_Protection*. The classes *Defect_Type and Defect_Cause* should be input manually. The terms used in each subclass are defined from literature reviews. The *Defect_Type* class contains the subclasses *Crack*, *Peeling*, *Water_Leaking*, and *Damage*. The *Defect_Cause* class contains the subclasses *Design_Omission*, *Work_Interference*, *Non-compliance_to_Standards*, etc. Each subclass provides two kinds of information as discussed in Table 1. The first represents the moment when an initial cause was generated such as design and construction phase. The second represents the person responsible, such as architect, site manager, and worker. For instance, in the case of *Design_Omission*, this kind of defect cause is generated in the design phase, and the architect is responsible for it.

The object properties of each class are defined using the same method. For instance, the *hasWork_Result* is an object property, and the domain and range are restricted to the class *Thing* and *Work_Result*, respectively. Also, the *hasSpace* object property has the class *Thing* as a domain and *Space* as a range. Data properties have the same domain but different *DataPropertyValue*s, depending on the data type, such as text, number, and URI. For example, the *Title* and *Description* data properties refer to a *String*, *Impact_Cost*, and *Time* refers to an *Integer*, *Date* refers to a *dateTime* value, and *Specification* and *Source* refer to an *anyURI* value for linking actual defect documents or web pages. An instance in this ontology has one of the six defined classes as a type and has seven data properties with instance values. A total of 110 selected defect cases have been converted to 110 instances by following this process.

5.3. Extraction of context information from BIM environments

The purpose of this section is extracting context information in a structured format from a BIM environment. The extracted information will be used to generate a structured defect case by following the defect

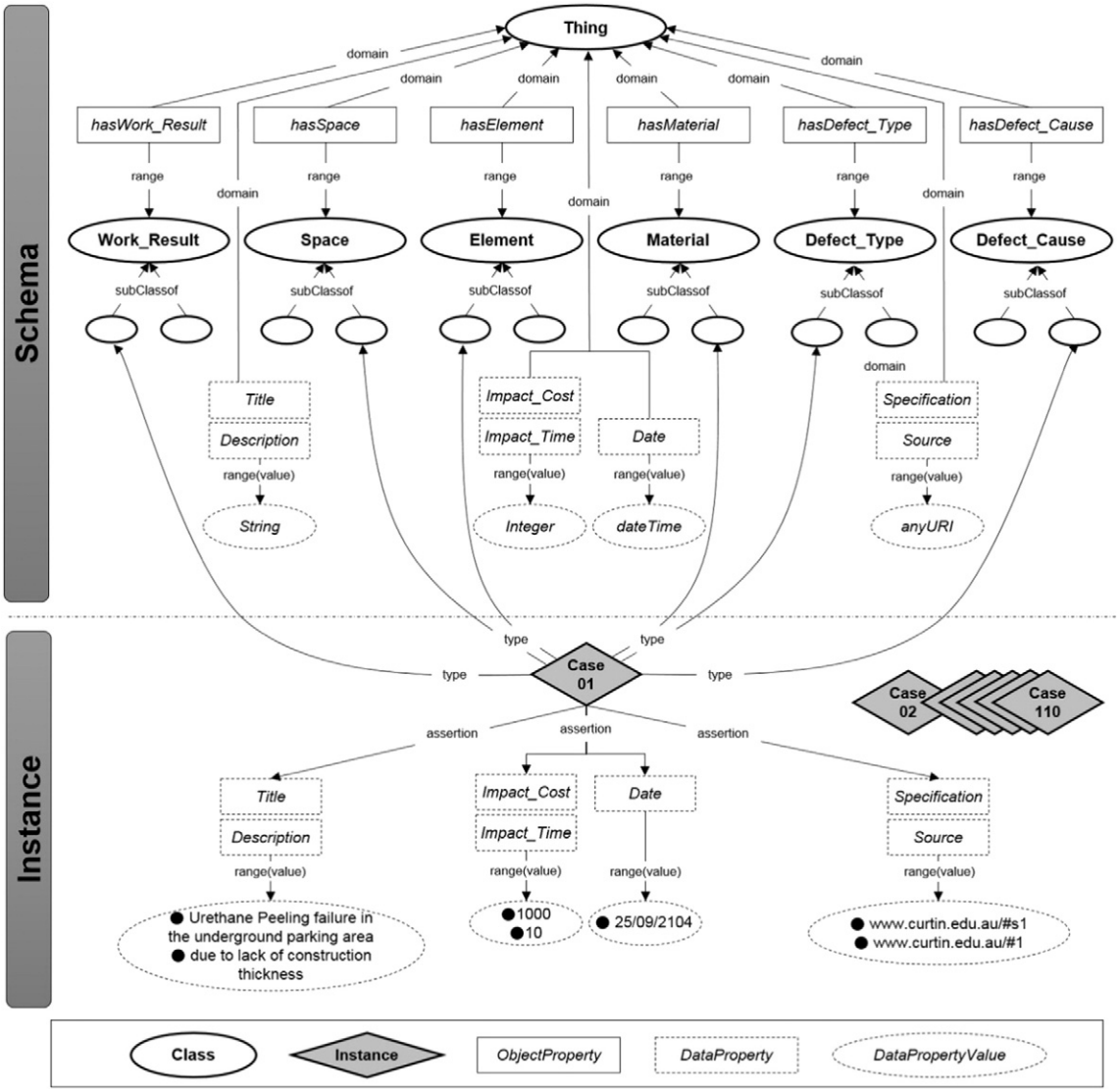

Fig. 3. Defect ontology schema.

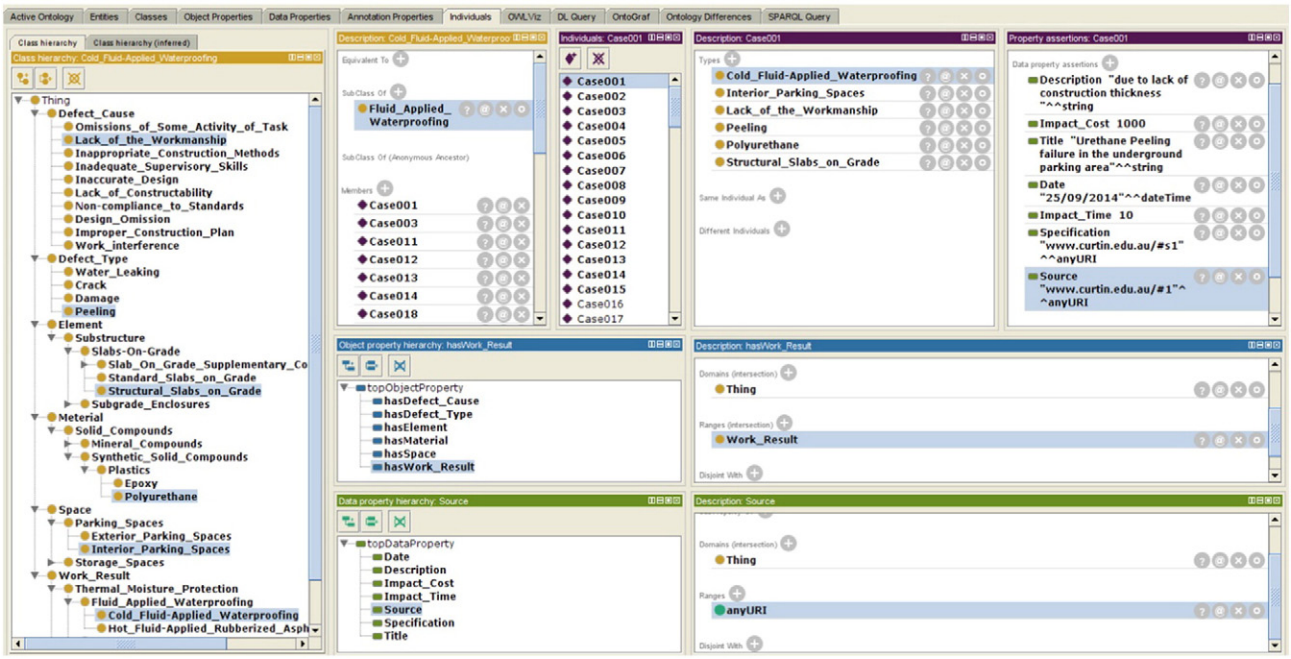

Fig. 4. The screenshot of the defect ontology development in Protégé.

ontology in Section 5.4. The Autodesk Revit program is used to create a sample model reflecting the situation of Fig. 1 and the applicability of a standard classification system has been examined.

Revit software supports various classification systems such as MasterFormat, UniFormat, and Omniclass. These classification systems can be assigned to BIM objects in the Type Properties window. UniFormat classifies data by systems and assemblies representing functional elements for preliminary project estimates. Also, MasterFormat classifies data by work results produced by construction trades for organizing construction specifications. However, they are not available for rooms or spaces. In the case of Omniclass, only Omniclass Table 23 (Products) is embedded, which represents single manufactured items such as metal window, curtain wall, and common brick. Therefore, it is difficult to use various types of Omniclass categories for our research purpose only using features offered by Revit software.

To assign Omniclass to BIM objects and extract context information in this paper, four kinds of shared parameters are created at the project level, namely, *Work_Result*, *Element*, *Space*, and *Material*. Then, the corresponding context information is filled in for each item manually. The information can be automatically extracted into a spreadsheet by the embedded software functions. As shown in Fig. 5, when clicking the foundation slab, the related contextual information can be triggered in the top-left corner. In addition, all this context information can be exported into a spreadsheet as a kind of structured format (as shown

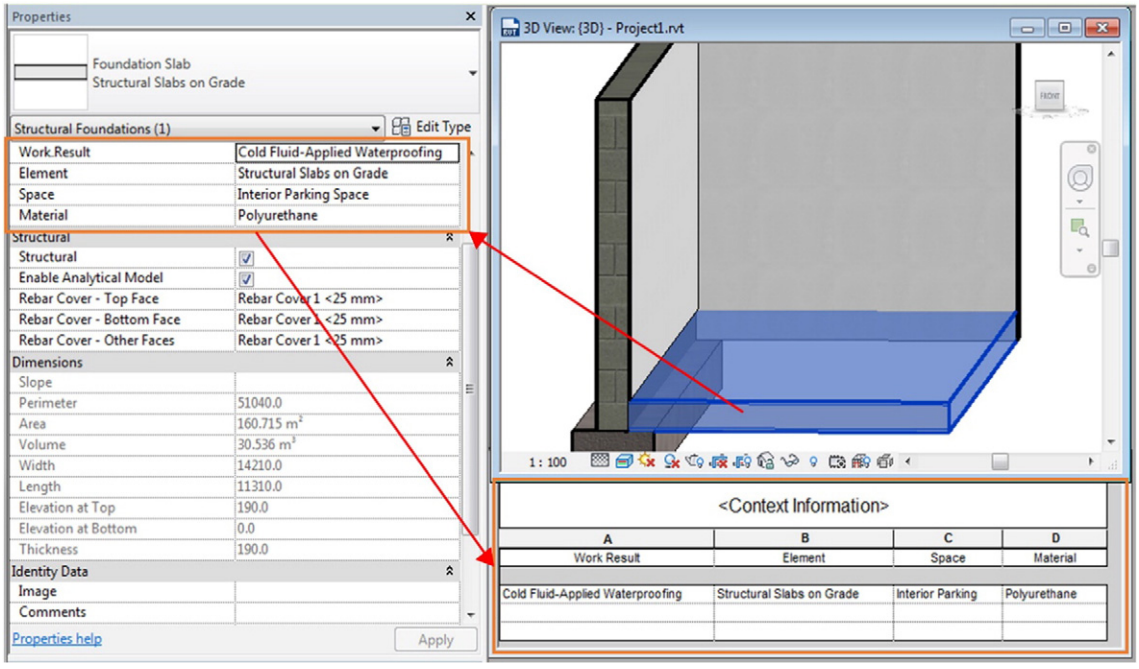

Fig. 5. Extraction of context information from a BIM environment.

in the bottom-right corner of Fig. 5). With this function, the user can easily get context information about the *Work_Result* (cold fluid-applied waterproofing), *Element* (structural slabs on grade), *Space* (interior parking space), and *Material* (polyurethane) of the foundation slab.

5.4. RDF conversion

In order to establish the linked data for data searching and sharing on the web, the context information extracted from the BIM environment has been converted from spreadsheet to an RDF/XML file. An open source library called dotNetRDF [41] is utilized to handle the conversion process. The library is developed based on the .NET platform, and helps handling RDF, SPARQL, and the semantic web. The customized conversion tool utilizing this library is developed and written in C# language. To use the tool, the user needs to specify the path of the target spreadsheet. The tool will then handle the relationship generation among different data items according to the defect ontology mentioned in previous sections. A file in RDF/XML format will then be the output. Fig. 6 illustrates a defect case using the RDF/XML format which has been converted from a spreadsheet by the developed tool. The defect is the polyurethane peeling on the structural slabs on grade that occurred in the interior parking space of the building. The defect is caused by a lack of workmanship during sprayed polyurethane foam roofing. All this information is defined using the classes in the defect ontology. As the ontology schema is published online, their relationship can be traced whenever SPARQL queries are made on the web. Other than type information, data properties can be explicitly expressed in the RDF file. For example, in this case, the Impact_Time of the defect is equal to 10. The subject of the presented statement is the defect case. Its predicate is impact time and the object is the property value 10. All these data properties are formed as triples in the file and are easy to retrieve in response to various data queries and further analysis.

5.5. Using SPARQL queries to access the information

This section presents four fundamental query examples for the specific purposes that have been identified in the literature review. Query statements have been tested in the Protégé platform, and the results of the queries show flexibility and the ability to retrieve meaningful information from organized RDF/XML data. To demonstrate the usage of the SPARQL queries, a preliminary test with the 110 defect has been used in this research.

The first example shows how to query similar defect cases compared to certain specific context information. In this query, a variable representing the cases of interest is used and listed as *?Case*. *SELECT* is a SPARQL reserved word for listing the query results of the variable. All the conditions are specified in the *WHERE* section. It is stated that cases of interest should all belong to the classes, *dm*:*Cold_Fluid_Applied_Waterproofing* subclass of *Work_Result*, *dm*:*Structural_Slabs_on_Grade* subclass of *Element*, *dm*:*Polyurethane* subclass of *Material*, and *dm*:*Interior_Parking_Spaces* subclass of *Space*. It should be noticed that the prefix *dm*: represents the URI resource of the defect ontology. The prefix is similar to that of other RDF standard ontologies like *rdf*:, *rdfs*:, and so on. The prefix statements of *Query 1* in front of *Query 2*, *3*, and *4* as well.

The query result of *Query 1* can be seen in Fig. 7. Of the 110 cases, three instances belong to all given classes. These cases all fit the four conditions the user specified. By following this method, the user can retrieve other groups of defects by simply changing the class names of this query.

The second example shows how to identify the inherited relationships and the locations of external resources among different defect cases. As can be seen in *Query 2*, the query helps to find all the defect cases that belong to the selected class (*dm*:*Thermal_Moisture_Protection*) or any of its subclasses. This query allows the user to recognize what kind of *Work_Result* is applied, even if the user does not exactly know about any existing defects that belong to the class or subclass of *dm*:*Thermal_Moisture_Protection*. In this query, *?Case*, *?Work_Result*, and *?anyURL* are three variables, and *SELECT* is a SPARQL reserved word for listing the query results of these three variables. All the conditions that describe the cases of interest are specified in the *WHERE* section. The conditions also include listings of all the resource URLs of those cases for users to access the related documents.

The query result of *Query 2* can be seen in Fig. 8. In total, 72 cases have shown up to be of the type *Thermal_Moisture_Protection* and containing a *dm*:*Source* property. *dm*:*Source* property means the web address to access an individual defect case. By simply changing the class name of this query, other inherited relationships and their resource locations can be found. It should be noted that duplicated case results could show up if the case has more than one *?anyURL* value.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE rdf:RDF [      <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
                <!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#'>
                <!ENTITY xsd 'http://www.w3.org/2001/XMLSchema#'>
                <!ENTITY dm 'http://example.org/'>
]>
<rdf:RDF xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
        xmlns:dm="http://example.org/"                          # Declare URL of
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">   defect ontology schema
  <rdf:Description rdf:about="&dm;Case1">
    <rdf:type rdf:resource="&dm; Cold_Fluid-Applied_Waterproofing" />   # Work_Result
    <rdf:type rdf:resource="&dm; Structural_Slabs-on-Grade" />          # Element
    <rdf:type rdf:resource="&dm;Interior_Parking_Spaces" />             # Space
    <rdf:type rdf:resource="&dm;Polyurethane" />                        # Material
    <rdf:type rdf:resource="&dm;Lack_of_the_workmanship" />             # Defect_Type
    <rdf:type rdf:resource="&dm;Peeling" />                             # Defect_Cause
    <dm:Date rdf:datatype="&xsd;dateTime">25092014</dm:Date>
    <dm:Description rdf:datatype="&xsd;string">due to lack of construction thickness </dm:Description>
    <dm:Impact_Cost rdf:datatype="&xsd;integer">1000</dm:Impact_Cost>
    <dm:Impact_Time rdf:datatype="&xsd;integer">10</dm:Impact_Time>
    <dm:Title rdf:datatype="&xsd;string">Urethane Peeling failure in the underground parking area</dm:Title>
    <dm:Specification rdf:datatype="&xsd;anyURI"> http://www.curtin.edu.au/#s1</dm:Specification>
    <dm:Source rdf:datatype="&xsd;anyURI"> http://www.curtin.edu.au/#1</dm:Webpage>
  </rdf:Description>
</rdf:RDF>
```

Fig. 6. Defect case conversion to RDF.

SPARQL Query 1

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dm: <http://www.semanticweb.org/.../dm#>
SELECT   ?Case
    WHERE {
        ?Case   rdf:type   dm:Polyurethane .
        ?Case   rdf:type   dm:Interior_Parking_Spaces .
        ?Case   rdf:type   dm:Structural_Slabs_on_Grade .
        ?Case   rdf:type   dm:Cold_Fluid-Applied_Waterproofing .
    }
```

Query Result:

Case
Case001
Case018
Case091

Fig. 7. SPARQL query 1 and result.

SPARQL Query 3

```
SELECT   ?Case   ?Cost
    WHERE {
        ?Case   rdf:type   dm:Water_Leaking.
        ?Case   dm:Impact_Cost   ?Cost .
            FILTER (?Cost > 1500)
    }
```

Query Result:

Case	Cost
Case004	"2520"^^< http://.../XMLSchema#integer>
Case002	"2500"^^< http://.../XMLSchema#integer>
Case007	"2500"^^< http://.../XMLSchema#integer>
Case018	"2500"^^< http://.../XMLSchema#integer>
Case055	"2500"^^< http://.../XMLSchema#integer>
Case062	"2500"^^< http://.../XMLSchema#integer>

Fig. 9. SPARQL query 3 and result.

As shown in the third example, the query approach allows users to analyze defect impact and identify critical management factors. This query approach allows users to find specific defect cases based on search parameters exemplified in *Query 1* and *2*. Furthermore, users can restrict numeric values by constraining data property variables assigned to *Impact_Cost* or *Impact_Time* classes. The reserved word *FILTER* helps to place further restrictions on data properties.

The query result of *Query 3* can be seen in Fig. 9. Of the 110 defect cases, 6 cases related to *Water_Leaking* were found with an impact cost higher than 1500 dollars. Similar restrictions can be set to *Impact_Time* or other data properties as well.

For dealing with massive amounts of defect information, a historical database is useful to determine the quality of defect management and help the ease of communication. To capture such information efficiently and effectively, *Query 4* can be used to retrieve statistical information, such as the frequency of the defect occurrence in the time period of interest. In this query, the reserved word *COUNT* is used to show the number of defect cases of type *Thermal_Moisture_Protection* that happened in 2014.

The query result of *Query 4* can be seen in Fig. 10. In total, 31 defect cases were found. A histogram of the frequency of defects with regard to time period can be drawn from the results of queries like *Query 4*. This is expected to be an appropriate reference for further analyses.

6. Discussion

The linked data framework for defect data sharing and reuse aims to provide technical feasibilities for integrating fragmented flows of defect information. Some technical solutions of the framework are implemented. However, during the implementation, various limitations and issues are observed, which need to be solved in future research.

First, the authors observed the inefficiency of assigning a classification system to objects in a BIM environment. In Section 5.3, shared parameters are created, and Omniclass titles and numbers are manually entered. Whenever a defect occurs, inputting or selecting these kinds of information manually is a cumbersome procedure because Omniclass consists of highly detailed hierarchical levels. Therefore, using Revit family feature is an alternative by predefining Omniclass title and number values and by reusing it. Hence, if these kinds of Revit families were widely utilized in project design, context information for each building

SPARQL Query 2

```
SELECT   ?Case   ?Work_Result   ?anyURL
    WHERE {
        ?Case   rdf:type   ?Work_Result .
        ?Case   dm:Source   ?anyURL .
        ?Work_Result   rdfs:subClassOf*   dm:Thermal_Moisture_Protection .
    }
```

Query Result:

Case	Work_Result	anyURL
Case008	Sheet_Metal_Waterproofing	"www.curtin.edu.au/#8"^^< http://.../XMLSchema#anyURL>
Case091	Cold_Fluid-Applied_Waterproofing	"www.curtin.edu.au/#91"^^<http://.../XMLSchema#anyURL>
Case010	Bentonite_Waterproofing	"www.curtin.edu.au/#10"^^<http://.../XMLSchema#anyURL>
Case027	Sprayed_Polyurethane_Foam_Roofing	"www.curtin.edu.au/#27"^^< http://.../XMLSchema#anyURL>
Case015	Bituminous_Sheet_Waterproofing	"www.curtin.edu.au/#15"^^< http://.../XMLSchema#anyURL>
⋮	⋮	⋮

Fig. 8. SPARQL query 2 and result.

SPARQL Query 4

```
SELECT (COUNT(*) AS ?Case)
    WHERE {
        ?Case   rdf:type    dm:Thermal_Moisture_Protection.
        ?Case   dm:Date     ?Time .
            FILTER (?Time >= "2014-01-01T00:00:00Z"^^xsd:dateTime
            && ?Time < "2015-01-01T00:00:00Z"^^xsd:dateTime)
    }
```

Query Result:

Count
"31"^^< http://.../XMLSchema#integer>

Fig. 10. SPARQL query 4 and result.

component could be automatically triggered and extracted from the BIM environment.

Second, in the Revit family parameters, *Work_Result*, *Element*, and *Material* information can be predefined. However, *Space* classification is not suitable to predefine it because one object can be installed in various spaces. Therefore, research is needed regarding how to identify and extract the relationship between an element and the space. As a possible solution, the IFC data representation standard can be utilized through the *IfcRelSpaceBoundary*, which is defined as an objectified relationship that handles the element to space relationship by objectifying the relationship between an element (*IfcElement*) and the space (*IfcSpace*) it bounds [42].

Third, in this paper, BIM is utilized solely for extracting context information and converting to the RDF format. Further research should figure out how to enable the use of extracted context information as SPARQL query statements in the BIM environment.

Finally, limitations in the usability of the SPARQL query language were found. SPARQL provides efficient ways of accessing linked data by allowing users to write flexible queries. However, in order to use the SPARQL query language, the user has to be familiar with the defect ontology. Therefore, user friendly query interfaces are required to allow general users to use SPARQL without the need to understand the SPARQL query language or the defect ontology. This issue was well-discussed in [43] by proposing several advanced search features such as faceted browsing, content-specific browsing, full-text search, navigation mechanisms, statistical analysis, etc. In the case of faceted or navigation browsing, the user could be easily provided with subclass concepts related to certain superclasses through the inheritance capacity of classes in a user-friendly manner. For instance, when the user selects 'Rooftop' in the 'Space' class, related work types applied in that space, which cause defects, are automatically provided by using the ontology relations. This means that the user can identify defect possibilities without inputting specific queries or knowledge under certain work conditions. End users can then navigate the data in an exploratory manner and support decision making before completing design or construction. Alternatively, applications that simplify the creation of SPARQL queries such as Visual Query Builder [43] could be adopted.

7. Conclusions

Construction standards, codes, and procedures have evolved through the lessons learned from various failures [44]. Similarly, defect data should be utilized for the advancement of the construction domain because it contains valuable experimental knowledge. The importance of defect data utilization is generally recognized in practice, so most companies put efforts into developing information management systems. However, traditional ways of managing defect data provide limited support to end-users.

This limitation can be divided into three main issues: lack of formal structure in data representations, inefficient inputting of detailed context information, and insufficient sharing and access to defect data from different data sources. To solve these issues, a new approach is considered and the flow of defect data should be technically interconnected from the perspective of knowledge management processes. Therefore, this study has proposed the linked data system framework for sharing construction defect information by leveraging the state of art technologies: BIM as central repositories of building data in digital format, ontologies to represent domains knowledge, and linked data for linking and searching defect data between different data sources. This approach might be able to address the long-standing data feedback challenges in the AEC domains.

In this paper, the applicability of BIM and linked data as possible solutions to improve the current limitations was investigated. First, defect cases related to waterproofing were collected and analyzed to reconfirm the necessary data feedback and to establish the standard defect data structure. Also, the defect ontology has been proposed based on analyzed findings and it has been developed with the protégé ontology authoring tool. Next, the processes of converting unstructured defect data to structured RDF format have been conducted. The proposed defect ontology consists of several classes which require multi-leveled Omniclass information. As a result, manually inputting this information is cumbersome. To automate it, the predefined Omniclass information was assigned to a single BIM object and extracted as a spreadsheet from BIM environments. Then, the program, which is able to convert the information in spreadsheet to an RDF/XML file, was developed by using dotNetRDF. Finally, the data search functions were executed through the SPARQL query in Protégé. Expected results were successfully attained in the test scenarios designed to meet various requirements, which have been identified in the literature reviews. The query functions developed in this research could support decision making by allowing access each defect data directly and by providing statistical information.

This paper has presented the overall visions of ongoing research efforts to implement a linked data approach to facilitate the dissemination of defect data for addressing defect reoccurrence problems. Further in-depth implementations are needed for evaluating the practical feasibility of this linked data framework, which is discussed in Section 5. The proposed framework is expected to be useful because it can provide new approaches of capturing the right information from different conditions, finding the right information when it is needed, and sharing the defect information to other parties. Moreover, the proposed approach, which enables linking BIM applications with other external data sources, can be applied to other domains that need to refer to external sources such as safety, specification, product, and regulation.

References

[1] P.E. Love, H. Li, Quantifying the causes and costs of rework in construction, Constr. Manag. Econ. 18 (4) (2000) 479–490.
[2] A. Mills, P.E. Love, P. Williams, Defect costs in residential construction, J. Constr. Eng. Manag. 135 (1) (2009) 12–16.
[3] P.E. Love, J. Smith, Benchmarking, bench action, and bench learning: rework mitigation in projects, J. Manag. Eng. 19 (4) (2003) 147–159.
[4] M. Macarulla, N. Forcada, M. Casals, M. Gangolells, A. Fuertes, X. Roca, Standardizing housing defects: classification, validation, and benefits, J. Constr. Eng. Manag. 139 (8) (2013) 968–976.
[5] P.E. Josephson, B. Larsson, H. Li, Illustrative benchmarking rework and rework costs in Swedish construction industry, J. Manag. Eng. 18 (2) (2002) 76–83.
[6] P.E.D. Love, Z. Irani, A project management quality cost information system for the construction industry, Inf. Manag. 40 (7) (2003) 649–661.
[7] E. Palaneeswaran, Reducing Rework to Enhance Project Performance Levels, in: E. Palaneeswaran (Ed.), Proceedings of the One Day Seminar on Recent Developments in Project Management in Hong Kong, Centre for Infrastructure and Construction Industry Development (CICID) of The University of Hong Kong, Hong Kong May 12, 2006, pp. 1–10.
[8] W.-K. Chong, S.-P. Low, Latent building defects: causes and design strategies to prevent them, J. Perform. Constr. Facil. 20 (3) (2006) 213–221.

[9] H.S. Jang, C.H. Seo, A study on the improvement of defect information management system of apartment house, J. Korea Inst. Build. Constr. 10 (2) (2010) 115–123.
[10] H. Wang, F. Boukamp, T. Elghamrawy, Ontology-based approach to context representation and reasoning for managing context-sensitive construction information, J. Comput. Civ. Eng. 25 (5) (2011) 331–346.
[11] C. Bizer, T. Heath, T. Berners-Lee, Linked data-the story so far, Int. J. Semant. Web Inf. Syst. 5 (3) (2009) 1–22.
[12] P. Pauwels, Supporting decision-making in the building life-cycle using linked building data, Buildings 4 (3) (2014) 549–579.
[13] C.-S. Park, D.-Y. Lee, O.-S. Kwon, X. Wang, A framework for proactive construction defect management using BIM, augmented reality and ontology-based data collection template, Autom. Constr. 33 (2013) 61–71.
[14] W.-K. Chong, S.-P. Low, Assessment of defects at construction and occupancy stages, J. Perform. Constr. Facil. 19 (4) (2005) 283–289.
[15] P.E. Love, D.J. Edwards, J. Smith, Systemic life cycle design error reduction model for construction and engineering projects, Struct. Infrastruct. Eng. 9 (7) (2013) 689–701.
[16] W3C, http://www.w3.org/2001/sw/ (accessed December 2015).
[17] T. Berners-Lee, Linked data-design issues, http://www.w3.org/DesignIssues/LinkedData.Html2006 (accessed December 2015).
[18] Linked data: evolving the web into a global data space, http://linkeddatabook.com/editions/1.0/ (accessed December 2015).
[19] Linking UK Government Data, http://logd.tw.rpi.edu/ (accessed December 2015).
[20] L. Ding, T. Lebo, J.S. Erickson, D. DiFranzo, A. Graves, G.T. Williams, X. Li, J. Michaelis, J. Zheng, Z. Shangguan, J. Flores, D.L. McGuinness, J.A. Hendler, TWC LOGD: a portal for linked open government data ecosystems, Web Semant. Sci. Serv. Agents World Wide Web 9 (2011) 253–268.
[21] C. Lima, T. El Diraby, B. Fies, A. Zarli, E. Ferneley, The e-COGNOS Project: Current Status and Future Directions of an Ontology-Enabled IT Solution Infrastructure Supporting Knowledge Management in Construction, in: K.R. Molenaar, P.S. Chinowsky (Eds.),Construction Research Congress, Winds of Change: Integration and Innovation of Construction March 19–21, 2003, pp. 1–8, http://dx.doi.org/10.1061/40671(2003)103 (Honolulu, Hawaii, USA).
[22] http://inteligrid.eu-project.info/ (accessed December 2015).
[23] J. Niu, R. Issa, Framework for Production of Ontology-Based Construction Claim Documents, in: R.R. Issa, I. Flood (Eds.), ASCE International Conference on Computing in Civil Engineering, ASCE, Clearwater Beach, Florida, USA June 17–20, 2012, pp. 9–16, http://dx.doi.org/10.1061/9780784412343.0002.
[24] G. Fidan, I. Dikmen, A.M. Tanyer, M.T. Birgonul, Ontology for relating risk and vulnerability to cost overrun in international projects, J. Comput. Civ. Eng. 25 (4) (2011) 302–315.
[25] T.R. Gruber, A translation approach to portable ontology specifications, Knowl. Acquis. 5 (2) (1993) 199–220.
[26] Y. Rezgui, Ontology-centered knowledge management using information retrieval techniques, J. Comput. Civ. Eng. 20 (4) (2006) 261–270.
[27] T.A. El-Diraby, C. Lima, B. Feis, Domain taxonomy for construction concepts: toward a formal ontology for construction knowledge, J. Comput. Civ. Eng. 19 (4) (2005) 394–406.
[28] P. Pauwels, R. De Meyer, J. Van Campenhout, Interoperability for the design and construction industry through semantic web technology, in: Semantic Multimedia–5th International Conference on Semantic and Digital Media Technologies, Lect. Notes Comput. Sci 6725 (2011) 143–158.
[29] J. Beetz, J. Van Leeuwen, B. De Vries, IfcOWL: a case of transforming EXPRESS schemas into ontologies, Artif. Intell. Eng. Des. Anal. Manuf. 23 (1) (2009) 89–101.
[30] M.P. Nepal, S. Staub-French, R. Pottinger, J. Zhang, Ontology-based feature modeling for construction information extraction from a building information model, J. Comput. Civ. Eng. 27 (5) (2013) 555–569.
[31] S. Demir, J.H. Garrett Jr., B. Akinci, O. Akin, M. Palmer, A Semantic Web-Based Approach for Representing and Reasoning with Vocabulary for Computer Based Standards Processing, in: W. Tizani (Ed.), Computing in Civil and Building Engineering, Proceedings of the International Conference, Nottingham University Press, Nottingham, UK June 30–July 2, 2010, pp. 1–7.
[32] S.-K. Lee, K.-R. Kim, J.-H. Yu, BIM and ontology-based approach for building cost estimation, Autom. Constr. 41 (2014) 96–105.
[33] P. Pauwels, D. Van Deursen, R. Verstraeten, J. De Roo, R. De Meyer, R. Van de Walle, J. Van Campenhout, A semantic rule checking environment for building performance checking, Autom. Constr. 20 (5) (2011) 506–518.
[34] B. Zhong, L. Ding, H. Luo, Y. Zhou, Y. Hu, H. Hu, Ontology-based semantic modeling of regulation constraint for automated construction quality compliance checking, Autom. Constr. 28 (2012) 58–70.
[35] J.H. Tah, H.F. Abanda, Sustainable building technology knowledge representation: using semantic web techniques, Adv. Eng. Inform. 25 (3) (2011) 547–558.
[36] E. Curry, J. O'Donnell, E. Corry, S. Hasan, M. Keane, S. O'Riain, Linking building data in the cloud: integrating cross-domain building data using linked data, Adv. Eng. Inform. 27 (2) (2013) 206–219.
[37] G. Costa, L. Madrazo, Connecting building component catalogues with BIM models using semantic technologies: an application for precast concrete components, Autom. Constr. 57 (2015) 239–248.
[38] L.O. Oyewobi, D.R. Ogunsemi, Factors influencing reworks occurrence in construction: a study of selected building projects in Nigeria, J. Build. Perform. 1 (1) (2010) 1–20.
[39] A.R. Fayek, M. Dissanayake, O. Campero, Measuring and Classifying Construction Field Rework: A Pilot Study, Research Rep.(May), Construction Owners Association of Alberta (COAA), The University of Alberta, Edmonton, Alta, Canada 2003, pp. 1–86.
[40] Y.S. Kim, S.W. Oh, Y.K. Cho, J.W. Seo, A PDA and wireless web-integrated system for quality inspection and defect management of apartment housing projects, Autom. Constr. 17 (2) (2008) 163–179.
[41] dotNetRDF - Semantic Web, RDF and SPARQL library for C#. Net, http://www.dotnetrdf.org/ (accessed December 2015).
[42] IfcRelSpaceBoundary, http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/ifcproductextension/lexical/ifcrelspaceboundary.htm (accessed December 2015).
[43] M. König, J. Dirnbek, V. Stankovski, Architecture of an open knowledge base for sustainable buildings based on linked data technologies, Autom. Constr. 35 (2013) 542–550.
[44] K.L. Carper, Failure information: dissemination strategies, J. Perform. Constr. Facil. 1 (1) (1987) 1–10.