The Application of Web of Data Technologies in Building Materials Information Modeling for Construction Waste Analytics

for

Sustainable Materials and Technologies

(Journal)

By

Muhammad Bilal[1,a]; Lukumon O. Oyedele[2,a,*] ; Kamran Munir[3,b]; Saheed O. Ajayi[4,a]; Olugbenga O. Akinade[5,a]; Hakeem A. Owolabi[6,a]; Hafiz A. Alaka[7,a];

(Authors)

<u>* Corresponding Author and Address</u>

Professor Lukumon O. Oyedele
Director of Bristol Enterprise, Research and Innovation Centre (BERIC)
Professor and Chair in Enterprise and Project Management
Bristol Business School
University of West of England, Bristol
Frenchay Campus, Coldharbour Lane
Bristol, BS16 1QY
United Kingdom
Tel: +44 (0) 117 32 83443 (Office)
E-mail: Ayolook2001@yahoo.co.uk; L.Oyedele@uwe.ac.uk

<u>Affiliations</u>

[a] Bristol Enterprise, Research and Innovation Centre (BERIC)
Bristol Business School
University of West of the England, Bristol, United Kingdom

[b] Faculty of Environment and Technology (FET)
University of West of the England, Bristol, United Kingdom

The Application of Web of Data Technologies in Building Materials Information Modelling for Construction Waste Analytics

Abstract

Predicting and designing out construction waste in real time is complex during building waste analysis (BWA) since it involves a large number of analyses for investigating multiple waste-efficient design strategies. These analyses require highly-specific data of materials that are scattered across different data sources. A repository that facilitates applications in gaining seamless access to relatively a large and distributed data sources of building materials is currently unavailable for conducting the BWA. Such a repository is the first step to developing a simulation tool for the BWA. Existing product data exchange ontologies and classification systems lack adequate modelling of building materials for the BWA. In this paper, we propose a highly resilient and data-agnostic building materials database. We use ontologies at the core of our approach to capture highly accurate and semantically conflicting data of building materials using the Resource Description Framework (RDF) and Web Ontology Language (OWL). Owing to the inherent capabilities of RDF, the architecture provides syntactical homogeneity while accessing the diverse and distributed data of building materials during the BWA. We use software packages such as Protégé and Oracle RDF Graph database for implementing the proposed architecture. Our research provides technical details and insights for researchers and software engineers who are seeking to develop the semantic repositories of similar kind of simulation applications that can be used for building waste performance analysis.

Keywords: Building materials database, RDF/OWL, Ontologies, Building waste analysis, Construction waste minimisation, NoSQL systems, Big data analytics

1 INTRODUCTION

With rising cost of construction projects and growing environmental concerns, the construction industry is under immense pressure to minimise construction waste. The construction industry in the UK is responsible for producing one-third of overall waste going to landfill [1]. By consuming over 400 million tonnes of material, it stands out as the leading consumer of natural resources. It is also responsible for generating more than 120 million tonnes of construction waste yearly. It is noticed that around 25 million tonnes of this waste can be recovered by employing proper recovery and reuse measures [2, 3]. For sustainable construction, considering the efficient usage of materials is the fundamental principal to reduce environmental impacts of construction such as reduced landfill and depletion of limited natural resources [4]. This consideration is likely to contribute to the economic efficiency of the sector not only in the UK but also in the whole world. Due to these reasons, it has become obligatory for the construction industry to take drastic steps to minimise construction waste [5].

Pointedly, a large number of activities occurring throughout the building lifecycle influence generation of construction waste [5]. Particularly, activities undertaken during the design stage have been highlighted to summate to one-third of the construction waste [6, 7]. The recently proposed idea of designing out waste [8] aims to achieve waste minimisation by emphasising incorporation of appropriate waste minimisation measures at the early design stages [3]. Designing out waste requires robust ways to pre-emptively analyse, estimate, predict, visualize, and minimise construction waste at early design stages. We are coining the term Building Waste Analysis (BWA) to capture the whole process of designing out construction waste. Using the BWA, the designers will be able to evaluate building waste performance in a timelier manner to eradicate construction waste.

Data of the building design and construction materials must be known to analyse the building waste performance accurately. Whereas, the data of the building design is widely captured and accessed through the Building Information Modelling (BIM) and the Industry Foundation Classes (IFC) respectively [9, 10, 11, 12], the data of building materials such as cost, dimensions, alternative materials, and waste potential remained uncaptured yet. According to knowledge principle, a great deal of real world knowledge is the prerequisite for an intelligent program to perform the complex analytical tasks (like the BWA) accurately [13]. Thereby, availability of the machine-readable data of building materials and their specific properties is fundamental to performing the BWA.

Currently, no such source of truth available could facilitate the applications in gaining seamless access to a large and distributed building materials data while carrying out the BWA [14]. Besides, material libraries accompanied with BIM authoring tools such as Autodesk Revit or other simulation tools/engines such as DOE-2 and EnergyPlus support tiny proportion of the standard building materials data [15]. Thus, the creation of a semantic repository containing the highly specific semantic representations of the construction materials, including both standard and alternative materials is the need of developing an efficient simulation tool for analysing the building waste performance. Existing product data exchange ontologies and classification systems (such as UniClass, OmniClass, MasterFormat, UNSPSC, eCl@ss, eOTD, and RNTD) lack effective modelling for describing the construction materials, required to carry out the BWA [16, 17, 18, 19]. As the focus underlying the majority of these initiatives is to classify construction materials/products than to describe their contents at the fine-grained level semantically.

Building materials can be standard, alternative, composite, and smart materials. The availability of 4000 different types of the metallic alloy and 5000 varieties of plastic, ceramics, and glass reveals that building materials data are highly diverse [20]. Furthermore, these materials possess thousands of unique properties of their physical, mechanical, thermal, chemical, optical, acoustical, and physiochemical characteristics [15]. Apart from the challenging task of capturing the semantics of large and complex properties of the building materials, another major challenge is associated with the strenuous data management issues related to the storage and maintenance of building data [14]. In this context, we employ the technique of ontologies in this study to achieve syntactically homogeneous representation of the materials data and allied knowledge that will be exploited during the process for BWA. Ontologies provide a powerful mechanism to organise and represent knowledge of a particular domain [21]. The inherent flexibility and simplicity of ontology data model enable knowledge of arbitrary domains to be represented in a straightforward fashion [22, 23]. In our work, we utilise the Web Ontology Language (OWL), the de facto standard for developing ontologies advocated by the World Wide Web Consortium (W3C) for our modelling purposes [23].

The aim of this study is to explore recent technological advancements in the fields of Web of Data technologies to harness the development of a highly scalable and data-agnostic semantic repository for building materials information modelling.

The specific objectives underlying this study include:

- *Developing the semantic models for representing construction materials and their associated properties.*
- *Developing building materials database for storing semantic models using the Web of Data technologies.*

Contributions of this work: Our work builds upon the considerable extant literature on building materials, Web of Data technologies, and RDF-based storage systems. In this paper, we have critically analysed existing standards for construction materials and product. The developments in Semantic Web technology, and particularly in ontology languages, are reviewed to determine the suitability of such techniques for representing building materials knowledge. The twofold technical contributions of this work include proposing *(i)* semantic models for describing the contents of building materials, and *(ii)* a framework to develop building materials database. In addition, this study contributes towards state-of-the-art in BWA by implementing the building materials database, which is an indispensable milestone to develop tools for construction waste minimisation. This study contains detailed insights and technical guidelines for researchers and software engineers interested in developing semantic repositories using Web of Data technologies.

Organization of this work: In section 2, semantic technologies such as RDF, OWL, ontologies, querying, inference and storage systems are briefly introduced. In Section 3, representational challenges posed by building materials datasets are discussed, and use of RDF data model to describe building materials semantically is justified. The critical features of the building materials database are discussed in section 4. In section 5, the literature review of existing product data exchange ontologies and classification systems is deliberated and it is explained that why these existing systems are insufficient for capturing the semantics of building materials. In section 6, the functionality and limitations of different components of the proposed building materials database are explained. In section 7, the implementation of proposed architecture is illustrated using Oracle Database 11g with spatial and RDF graph feature. Finally, in section 8, we conclude the paper and briefly provide an outlook of the future research directions.

2 Web of Data Technologies

Semantic Web is envisioned to shift the web of documents to the web of meaning, in which ontologies play a central role in knowledge modelling and representation [24, 25]. A domain essentially comprises set of concepts and their relationships [26]. Semantic Web technologies are developed to solve critical issues of the Web and are frequently adopted by the wider applications pertaining to finance, government, and enterprise applications [23]. Some of the prominent concepts used in Semantic Web technologies include triple (subject, predicate, and object), URIs, blank nodes, plain/typed literals, RDF, OWL, SPARQL, inference engines, and storage systems. This section describes these concepts to provide an adequate understanding of these technologies for building materials information modelling.

Selecting ontology development language is the first step in developing ontology-based applications. Substantial efforts are carried out for developing ontology languages. RDF is the World Wide Web Consortium (W3C) standard for managing the distributed data in semantic web applications [27]. The notion of resources is fundamental to RDF data model that are identified by Universal Resource Identifiers (URIs) [28]. Semantic data is effectively modelled in RDF using triples, comprising <s, p, o> where s, p, and o are respectively the subject, predicate, and object of the triple. Subjects and predicates must be URIs whereas the object can be a URI, plain/typed literal or a blank node. Each triple models a unique and complete fact of the domain. A bunch of triples forms the RDF graph. RDF data can be serialized in different formats, including RDF/XML (eXtensible Markup Language), Notation-3 (N3), Turtle, N-Triple, RDFa, and RDF/JSON (JavaScript Object Notation). Whilst RDF offers flexible, graph-based data model to describe resources but it lacks the ability to attach meaning with the resources [23].

To cope with this, a simplified technique of standardizing the most popular domain terms is adopted by some of the initiatives like Dublin Core Metadata Initiative (DCMI) where a set of terms like dc:title, dc:creator, and dc:publisher with agreed meanings are introduced [29]. Nevertheless, this approach is less flexible and suits only in constrained settings. Alternatively, ontologies are proposed to define domain-specific terms and meanings. Consequently, first ontology language RDFS (RDF Schema) is developed to extend RDF by introducing a number of modelling constructs like rdfs:Class, rdfs:subClassOf, rdfs:subPropertyOf, rdfs:domain, and rdfs:range [30]. The rdf:Class is used to define RDF classes. The rdf:subClassOf and rdfs:subPropertyOf define class and property hierarchies. The rdfs:domain and rdfs:range are used to specify domain and range of properties. RDFS provides considerable subtlety for modelling domains but using constructs inappropriately are subject to misinterpretations. Other ontology languages are also developed such as Simple HTML Ontological Extensions (SHOE), the Ontology Inference Layer (OIL), and DAML + OIL with different modelling capabilities [31, 32, 33].

These languages have set the stage for the development of Web Ontology Language (OWL)—that is the de facto standard in the industry for creating ontologies for its better expressive power and reasoning capabilities [34]. OWL has three sub-languages *(i)* OWL-Lite, *(ii)* OWL-DL, and *(iii)* OWL-Full. The building blocks of OWL ontologies are mainly individuals, classes, and properties [34]. Individuals are the basic elements of the domain; classes classify individuals with similar characteristics, and properties elicit the relationships between individuals. Classes and properties can have the hierarchical relationship (subClassOf, subPropertyOf, and equivalentClass). Classes can be defined using expressions involving logical operators like intersectionOf, unionOf, complementOf or enumeration of the specified objects. Properties can be described as transitiveOf, inverseOf, symmetric, functional, or inverseFunctional. Individuals can be assigned to specific classes. OWL allows the definition of classes using quantification restrictions like someValuesFrom and allValuesFrom. Cardinality constraints can be imposed through minCardinality and maxCardinality [22]. Furthermore, OWL ontologies support two types of axioms. The ones that place constraints on the structure of domain are terminology boxes (TBOX), whereas others that describe facts about the real situations are called assertion boxes (ABOX). Collectively, these TBOX and ABOX axioms form the knowledge base [35].

Recently, OWL 2.0 is released with new improvements. The ontologies, developed using OWL-1.0, are compatible with OWL-2.0 [35]. To bring efficiency in reasoning, OWL 2.0 has three profiles types, including *(i)* OWL-EL++, *(ii)* OWL-QL, and *(iii)* OWL-RL. The reasoning tasks and structure of the domain influence their selection. OWL-EL++ is suitable with ontologies with enormous classes and properties whereas OWL-QL suits the ontologies with many individuals. OWL-RL is right for scalable reasoning with much expressivity. OWL-2.0 supports new features such as property chains, richer datatypes, data ranges, and qualified cardinality restrictions [23, 35]. A feature-based comparison amongst these ontology languages is shown in the Table 1. To sum up, OWL has greater expressivity than RDFS in spite of having many standard features. OWL is a stronger language with greater machine interpretability and comes with a larger vocabulary and more powerful syntax than RDFS, which can be used not only to define complex ontology restrictions but also adequately capture domain-sensitive information of building models and construction materials.

Since maintaining large and complex ontologies is non-trivial, supporting tools and services are needed to build and maintain ontologies [37]. In this context, ontology-reasoning systems are especially useful for developing and deploying high-quality ontologies in the construction of ontology-based systems. Popular ontology reasoning systems are Fact++, Racer, and Pellet [36, 37, 38]. Horrocks (2008) has discussed that a large number of inconsistencies are sorted out in ontologies used in many real-world ontology-based systems with the help of these reasoning systems. Apart from Semantic Web, ontologies are widely used in the data integration systems for describing domain knowledge, describing the capabilities of existing source, and formulating queries [39].

As ontologies grow, scalability becomes critically important. To this end, efficient methods of storage and retrieval of the RDF data in the ontology-based systems are developed. These systems can be broadly categorized into *(i)* Native Stores—These systems store the data closer to the RDF data model without translating RDF data into secondary format for storage, *(ii)* Non-Native Stores—These systems use DBMSs and other related systems to store the RDF data. Interested readers can find in-depth comparisons of these systems in Faye, et al., (2012).

Table 1: Comparing Ontology Languages—RDFS, OWL-1.0, and OWL-2.0.

S#	Description	Construct	RDFS	OWL-1.0	OWL-2.0
1	Class definition	rdfs:Class	√	√	√
2	Class-instance specification	rdf:type	√	√	√
3	Subsumption relationships	rdfs:subClassOf	√	√	√
		rdfs:subPropertyOf	√	√	√
4	Domain and range qualifiers	rdfs:domain	√	√	√
		rdfs:range	√	√	√
5	Construct for concept descriptions	owl:intersectionOf	×	√	√
		owl:unionOf	×	√	√
		owl:complementOf	×	√	√
6	Properties	owl:transitiveOf	×	√	√
		owl:inverseOf	×	√	√
		owl:symmetric	×	√	√
		owl:functional	×	√	√
		inverseFunctional	×	√	√
7	Existential Quantification	owl:someValuesFrom	×	√	√
		owl:allValuesFrom	×	√	√
8	Advanced options	owl:reflexive	×	×	√
		owl:irreflexive	×	×	√
		owl:asymmetric	×	×	√
		owl:disjoint	×	×	√
9	Data types specification	-	×	×	√
10	Profiles	OWL-EL++	×	×	√
		OWL-RL	×	×	√
		OWL-QL	×	×	√
11	Horn-like rules	SWRL Inference	×	×	√
12	Modular design and extraction	-	×	×	√

In recent years, NoSQL (for "not only SQL") systems have emerged as a substitute for classical DBMSs to persistently store and query RDF data [40, 41, 42, 43]. In simple terms, NoSQL systems store unstructured data in a highly efficient and flexible key-value format. However, RDF data requires more specialized features to process graph data, thereby a graph-based data model is recently proposed for NoSQL systems for the efficient processing of RDF data [44, 45]. Notable graph-based NoSQL systems are Oracle NoSQL, Apache Cassandra, Voldemort, and MongoDB, among others [46, 47, 48, 49].

3 SEMANTIC DESCRIPTIONS OF THE BUILDING MATERIALS

Building materials are the standardized substances used to construct building elements, such as bricks, blocks, timber, steel, concrete, etc. Since building materials are the key determinants for measuring the building waste performance, the analytical tasks underpinning the BWA often require precise measurements of different properties of the building materials [50]. Building materials can be broadly divided into four major categories, namely standard materials, alternative materials, composite materials, and smart materials.

For a single material, a lot of variation is available as shown in Figure 2. These materials possess thousands of unique properties of their physical, mechanical, thermal, chemical, optical, acoustical, and physiochemical characteristics [15]. Materials have basic as well as advanced properties. The basic properties include the name of material, weight, and price. The advanced properties refer to highly accurate data required to carry out certain analytical tasks. For example, the dimensions of materials are useful to investigate floorplans of the design and guide designers to optimise them such that the optimised plans require relatively less onsite cutting and fitting. The design for dimensional coordination of the floorplans is consequently improved, eventually resulting in the waste-efficient building designs.

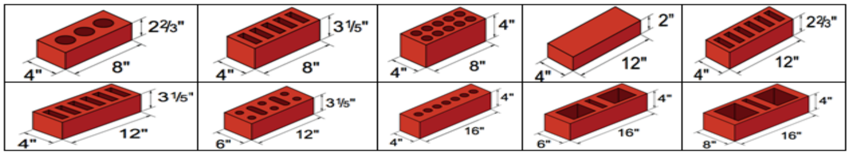


Figure 1: Different types & sizes of brick

Describing properties of building materials is a crucial first step towards developing a simulation tool to perform the BWA [14]. However, capturing the semantics of building materials is challenging due to a large number of inherent heterogeneities [51, 52, 53]. For example, it is unrealistic, and contrary to practice, to assume that all designers shall use the same name to refer to building materials. Several different names are often used to follow the company's naming conventions, such as "Concrete", "Conc", and "Con'c".

Table 2: A simplified conceptual schema for representing material properties

Property Name	Data Type	Description
Id	String	Global unique Identifier of each material
Name	String	Material name
Synonym	String	Other names used by professionals
Width	Float	Thickness of the brick in mm
Length	Float	Length of the brick in mm
Height	Float	Height of the brick in mm
Weight	Float	Weight of the brick in KGs
Price	Float	Price of the brick
ClassSystem	String	Classification System
ClassId	String	Class code in the respective classification system
AlternativeMaterials	List	Alternative materials

Being the promising application areas for Semantic Web technologies, ontologies are employed to capture the fine-grained semantics of building materials. Table 2 depicts the conceptual schema for describing the building materials required to carry out the analysis for standardisation and dimensional coordination (S&DC). A mix of both OWL built-ins and user-defined constructs are used to describe material properties. For example, mdb:synOf property is defined to reconcile the naming conflicts present in modelling building materials information. mdb:synOf is an acronym of "synonym of". Thus, the statement mdb:synOf (Concrete, Conc) says that "Concrete", and "Conc" is the same material. To further enhance the reasoning for mdb:synOf property, a rule mdb:synOf(Conc, Concrete) <- mdb:synOf(Concrete, Conc) is defined to express if "Concrete" and "Conc" are related via the mdb:synOf property, the same relationship also holds in reverse. The formal semantics are furthermore enriched for transitivity by adding the axiom mdb:synOf(Concrete, Con'c) <- mdb:synOf(Concrete, Conc) ^ mdb:synOf(Conc, Con'c) that explains if "Concrete" is the synonym of "Conc" and "Conc" is synonym of "Con'c" then "Concrete" is also the synonym of "Con'c". This way the bi-directional associativity and transitivity is achieved and the issues of data incompleteness are overcome.

The materials ontology for conceptual schema shown in Table 2 is implemented using two types of OWL properties. Firstly, the object property mdb:alternativeOf property is defined as the subclass of OWL owl:ObjectProperty for linking alternative materials. Secondly, the data type properties (mdb:hasID, mdb:hasName, mdb:hasWeight, mdb:hasPrice, mdb:hasDimensions, etc.) are defined to extend the built-in OWL owl:DataType property for assigning plain/literal string values to the properties of the construction materials. Figure 2 depicts the snapshot of the resultant graph. This ontology-driven representation enables the syntactic homogeneity in representing highly accurate and complex data of building materials through the RDF/OWL constructs.

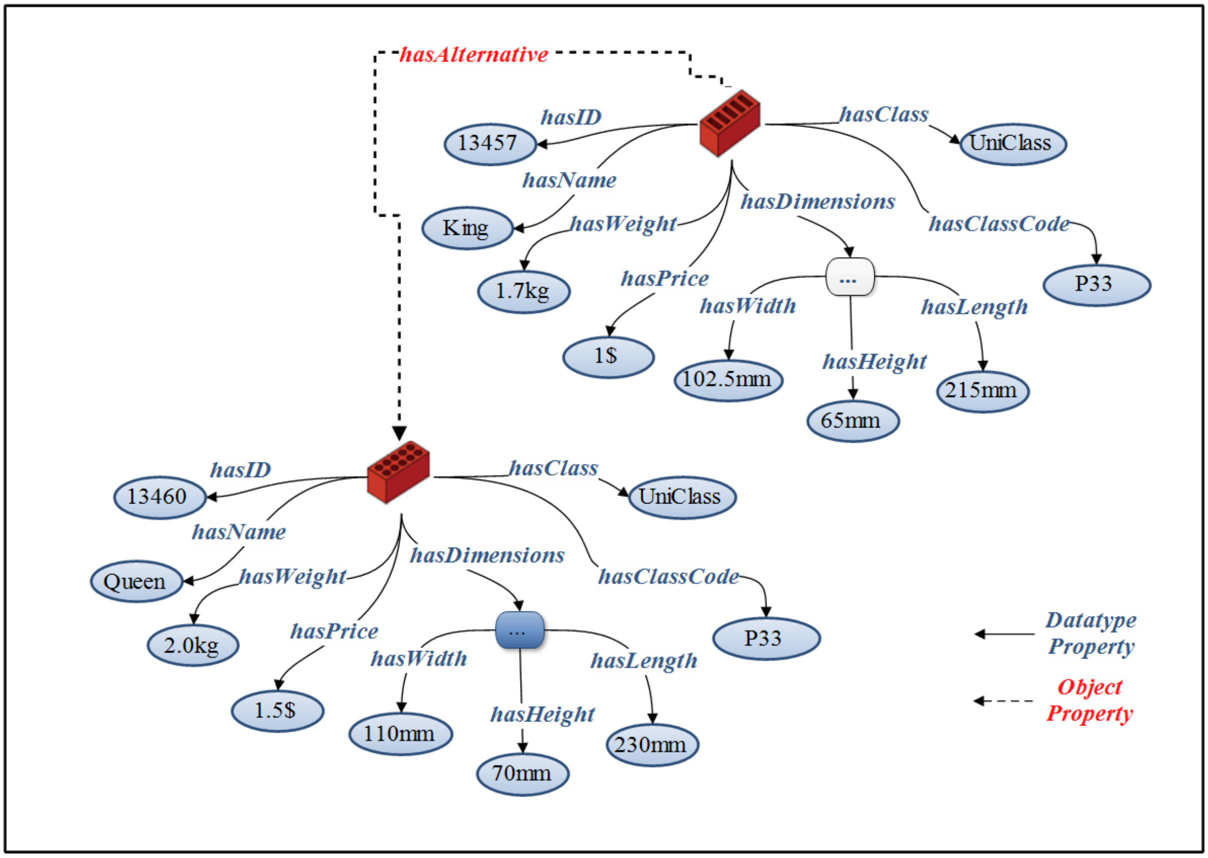
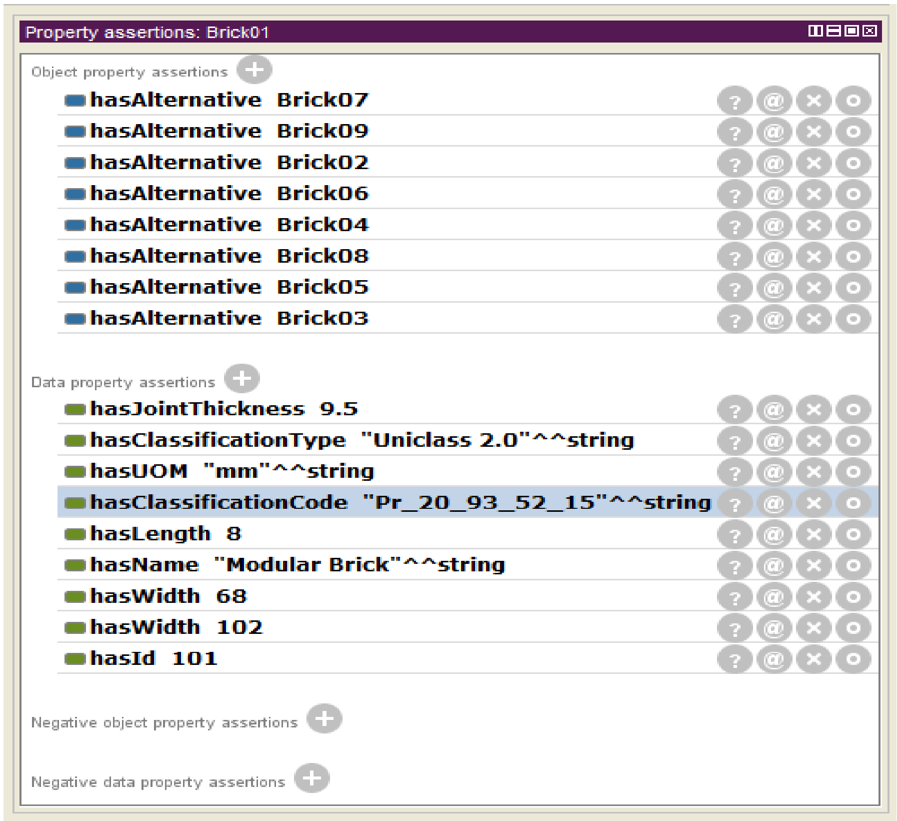


Figure 2: Semantic models of bricks using RDF based graph data models

In this work, we use Protégé 4.3 to further aid to the development of the RDF/OWL models for building materials. Protégé is the leading open source ontology-engineering tool, which provides graphical user interface to create different artefacts of the ontologies declaratively. Classes, datatype/object properties, axioms, individuals are created using a number of visual editors. Users can check the consistency of the RDF/OWL models developed in Protégé using the built-in reasoners. Figure 3 illustrates the property assertion editor displaying the brick instance showing object and data types properties populated in it. The resultant RDF/OWL model is exported in RDF/XML serialization format as is shown in Listing 1.

In this work, we use Protégé 4.3 to aid further to the development of the RDF/OWL models for building materials. Protégé is the leading open source ontology engineering tool, which provides a graphical user interface to create different artefacts of the ontologies declaratively. Classes, datatypes, axioms, individuals are created using visual editors. Users can check the consistency of the RDF/OWL models developed in Protégé using the built-in reasoners. Figure 3 illustrates the property assertion editor displaying the brick instance showing object and data types properties populated in it. The resultant RDF/OWL model is exported in RDF/XML serialization format as is shown in Listing 1.



Property assertions: Brick01

Object property assertions

hasAlternative Brick07
hasAlternative Brick09
hasAlternative Brick02
hasAlternative Brick06
hasAlternative Brick04
hasAlternative Brick08
hasAlternative Brick05
hasAlternative Brick03

Data property assertions

hasJointThickness 9.5
hasClassificationType "Uniclass 2.0"^^string
hasUOM "mm"^^string
hasClassificationCode "Pr_20_93_52_15"^^string
hasLength 8
hasName "Modular Brick"^^string
hasWidth 68
hasWidth 102
hasId 101

Negative object property assertions

Negative data property assertions

Figure 3: Protégé Editor Property Assertions for Modular Brick

```
-- Classes Specification
<owl:Class rdf:about="&mdb;Material">
        <owl:disjointWith rdf:resource="&mdb;Vendor"/>
    </owl:Class>
-- Object Properties
<owl:ObjectProperty rdf:about="&mdb;hasAlternative">
        <rdf:type rdf:resource="&owl;TransitiveProperty"/>
        <rdfs:domain rdf:resource="&mdb;Material"/>
        <rdfs:range rdf:resource="&mdb;Material"/>
        <owl:inverseOf rdf:resource="&mdb;isAlternativeOf"/>
    </owl:ObjectProperty>
-- Data Properties
<owl:DatatypeProperty rdf:about="&mdb;hasName">
        <rdfs:domain rdf:resource="&mdb;Material"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&mdb;hasWidth">
        <rdfs:domain rdf:resource="&mdb;Material"/>
        <rdfs:range rdf:resource="&xsd;decimal"/>
    </owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&mdb;hasHeight">
        <rdfs:domain rdf:resource="&mdb;Material"/>
        <rdfs:range rdf:resource="&xsd;decimal"/>
    </owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&mdb;hasLength">
        <rdfs:domain rdf:resource="&mdb;Material"/>
        <rdfs:range rdf:resource="&xsd;decimal"/>
    </owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&mdb;hasJointThickness">
        <rdfs:domain rdf:resource="&mdb;Material"/>
        <rdfs:range rdf:resource="&xsd;decimal"/>
    </owl:DatatypeProperty>
--Individuals
<owl:NamedIndividual rdf:about="&mdb;Brick01">
        <rdf:type rdf:resource="&mdb;Material"/>
        <hasId rdf:datatype="&xsd;integer">101</hasId>
        <hasWidth rdf:datatype="&xsd;decimal">102</hasWidth>
        <hasWidth rdf:datatype="&xsd;decimal">68</hasWidth>
        <hasLength rdf:datatype="&xsd;decimal">8</hasLength>
        <hasJointThickness rdf:datatype="&xsd;decimal">9.5</hasJointThickness>
        <hasName rdf:datatype="&xsd;string">Modular Brick</hasName>
        <hasClassificationCode rdf:datatype="&xsd;string">Pr_20_93_52_15</hasClassificationCode>
        <hasClassificationType rdf:datatype="&xsd;string">Uniclass 2.0</hasClassificationType>
        <hasUOM rdf:datatype="&xsd;string">mm</hasUOM>
        <hasAlternative rdf:resource="&mdb;Brick02"/>
        <hasAlternative rdf:resource="&mdb;Brick03"/>
    </owl:NamedIndividual>
```

Listing 1: Subset of the RDF/OWL models of the building materials

4 Critical Features of Building Materials Database

The development of the building materials database is more than just proposing the RDF/OWL models of building materials. It is noticed that the data of the building materials is scattered and fragmented across the multiple and heterogeneous data sources within a construction company, preferably in relational databases and text files. Data about the standard building materials can be obtained from the built-in material libraries accompanied with BIM authoring tools such as Autodesk Revit or other simulation tools/engines such as DOE-2 and EnergyPlus [15]. However, the data about the complete list of alternative building materials is not managed in either of construction companies and material suppliers' databases since they maintain data for just the materials they are accustomed with. Obviously, the choices of designers are confined to a predefined list of non-optimal materials [1]. There are unprecedented opportunities for optimal materials specification by integrating material data from the data sources spanning multiple construction companies. However, combining this data requires reconciling a large number of schematic and semantic heterogeneities that poses many data integration challenges [54, 55, 56].

Furthermore, the materials industry is highly innovative and produces materials with different properties over the period to meet the current design, production and construction needs [57]. As a result, large collections of materials are available in the market: e.g., bricks are produced in many sizes and can be laid in a variety of patterns (see Figure 1). This knowledge about the properties of materials is required to carry out different types of analysis, which in itself will turn into huge dataset requiring distributed storage and parallel processing. The use of big data applications is of immense relevance to maintain and query this emerging size of material data. Based on the issues mentioned above, the intended list of critical features required to build building materials database include:

(i) Supporting the rich and machine-readable descriptions of building materials

(ii) Integrating the data of building materials from diverse classification system seamlessly

(iii) Handling the incompleteness inherently underlies the building materials data

(iv) Collating the data across scattered and fragmented data sources

(v) Enabling semantic searching and findability of building materials in real time

(vi) Handling the storage and processing of massive datasets of building materials

(vii) Being highly available and accessible

5 LITERATURE SURVEY

Substantial efforts are reported in the literature to develop similar standards/ontologies for augmenting the textual description of products for e-commerce such as UNSPSC, eCl@ss, eOTD, and RNTD [18]. Few of their limitations include information loss, limited expressiveness, and uneven/sparse categorization [16]. GenTax approach is developed to tackle the issue of information loss [17], whereas eClassOWL and unspscOWL ontologies are proposed to improve their semantic capabilities [19]. Furthermore, GoodRelations is developed to describe the commercial aspects of products for online searching [16]. The construction industry has undertaken similar efforts to standardise the descriptions of construction materials. Eurobau database is developed, and the materials data from ten European countries is inserted in the Eurobau database [14]. This database is further enhanced for querying and data integration capabilities in the EurobauWeb project. Besides these efforts, efforts are made to develop electronic catalogues for materials. Kong et al. (2005) (Kong *et al.*, 2005) have developed a Web-based electronic catalogue for searching construction materials information. Beetz (2009) has proposed a methodology for embedding the descriptions of construction materials in the standard HTML pages through RDFa. Nikam & Karshenas (2015) investigated the linking the construction materials information with BIM models. Zhang et al. (2015) proposed a novel ontology-driven knowledge sharing framework for engineering materials selection. However, the framework is constrained to materials selection for the manufacturing process and is well-suited for mold-making materials.

The majority of these works are inadequate to be adopted as-is for developing the ontology for building waste performance analysis (BWA). Pointedly, materials classification and materials description systems have different scopes and requirements. Although, existing ontologies such as UniClass and OmniClass are mainly designed for the classification purposes to unify and organize building materials data but they are still a long way off the actual ability to describe the highly specific data of building materials semantically. In addition, these ontologies are often employing static and stringent coding schemes with predefined levels [16], which limits their flexibility in storing and querying the data of building materials for the intended computation underpinning the BWA. Table 3 evaluates some of the prominent ontologies with the critical features for building the building materials database and clearly shows that most of the features are not provided by the prevalent standards/ontologies. This calls for the development of a highly generic and data-agnostic building materials database that could be utilized in the development of highly performance construction waste simulation tool.

Table 3: Evaluating existing standards based on critical features of material database

EXISTING STANDARDS CRITICAL FEATURES		UNPSC	eCl@ss	eOTD	RNTD	eClassOWL	unspscOWL	GoodRelations	FreeClass	FreeClassOWL
1	Rich and machine-readable descriptions	√	√	√	√	√	√	√	√	√
2	Supports multiple classification systems	×	×	×	×	×	×	×	×	×
3	Handles incompleteness & uncertainty	×	×	×	×	×	×	×	×	√
4	Integrates heterogeneous data across	×	×	×	×	×	×	×	×	√
5	Semantic searching	√	√	√	√	√	√	√	√	√
6	Handles large datasets	×	×	×	×	×	×	×	×	×
7	Highly available and accessible	×	×	×	×	×	×	×	×	×

6 PROPOSED BUILDING MATERIALS DATABASE

The critical features required for developing construction materials database are identified in the previous section. This section discusses the proposed ontology-centric architecture for developing a highly resilient building materials database. In addition to discussing the high-level descriptions of the architectural components, technical details are demonstrated to explain the reference implementation. This includes code snippets to meets some of the essential specifications for testing the functionality of the proposed architecture. Figure 4 shows the proposed architecture and highlights the key components and their interactions. These components are designed with scalability and performance in mind. More details of these components are provided in the subsequent sections.

6.1 SEMANTIC REPOSITORY

In the proposed architecture, semantic repository performs the pivotal role of materials data management repository where the construction materials are described as RDF/OWL models. Apart from storing the RDF data, it will provide the capabilities of inference engine while querying this data since inference is vital for querying the RDF data. Like any other database management system (DBMS), the semantic repository will provide all the features (such as indexing) required managing and processing the construction materials data efficiently. Since the underlying implementation details and complexities of managing RDF data are hidden, applications are provided with an abstraction to access the materials data from a centralized repository using semantic queries. Furthermore, it also interprets the schemas/ontologies expressed in RDF/OWL languages to transport the materials data back and forth with other systems. A key distinguishing feature of the proposed semantic store is that it can be used as standalone materials database or can be hosted on cloud services or can be distributed physically over a set of commodity servers. This will not influence any code changes to the applications developed on top of materials database. According to Domingue et al. (2011), the semantic repository is composed of two main components including *(i)* Triplestore and *(ii)* Querying and inference engine.

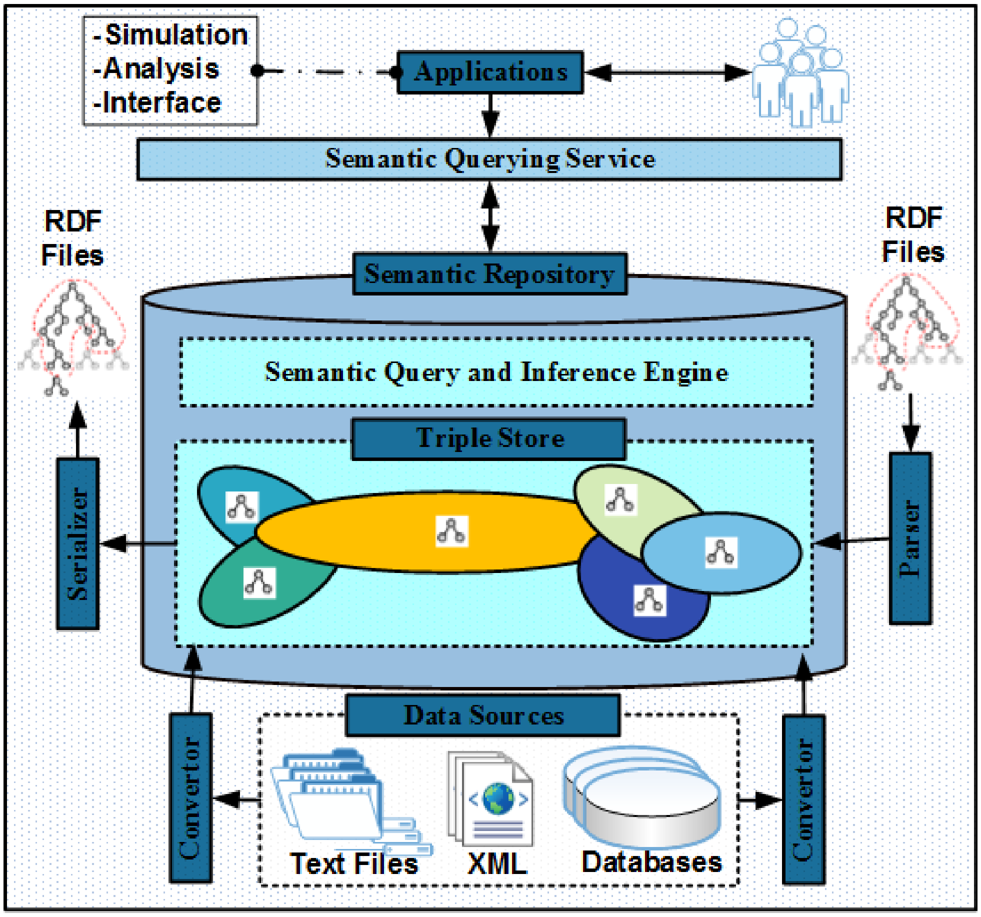


Figure 4: The proposed ontology-centric architecture of building materials database

The Triplestore is the storage component, responsible for handling the data management requirements of the construction materials data. We are adopting the RDF data model to represent the materials data in the Triplestore, so it is inherently suited to integrate the data stemming from the heterogeneous data sources. Namespaces are used for merging the properties of construction materials together arising from multiple data sources to unify the RDF graphs. Logically, the construction materials have data informing its schema and structure called TBOX and the data representing the values for the schema elements called ABOX. This data is stored collectively in shared RDF space. There are various strategies to implement Triplestore. The most popular strategy exploits the relational implementation. Additional features provided by Triplestore include indexing, support for importing RDF data, publishing RDF data in a wide variety of formats, and more importantly integrated interfaces to query construction materials data from applications and other systems.

Another defining characteristic of a semantic repository is to support vibrant environment for querying the RDF data. In contrast to relational queries, evaluating the semantic queries is non-trivial since the query engine has to consider asserted as well as inferred triples. Inferred triples are the ones that are computed on the fly by applying rules on the RDF data. It is on this basis that inference engine is the vital constituent of query engine and stands between the applications and Triplestore. A large number of RDF querying languages are developed with varying inference capabilities. SPARQL query is the W3C recommendation language for querying the RDF data from semantic repositories. Further, the indices supported by Triplestore also generally utilized by the query engine to execute the given SPARQL query at the real time efficiently.

6.2 SEMANTIC QUERYING SERVICE

Semantic querying service enables the applications to query the RDF/OWL models of building materials. It exposes interfaces, which are utilized by applications to query and manipulate semantic data and ontologies. Semantic querying service exposes the RDF data in two different ways. Firstly, through the SPARQL API comprising a set of libraries and Java classes that can be imported into the application code directly. The API provides all functionality to make database connections, write/execute semantic queries and retrieve the results, declaratively. The second mechanism to query and manipulate the RDF/OWL models through the web service interface. This option is widely adopted and is commonly known as SPARQL endpoints. The applications get the handle of web services interface and then query and execute the data using the communication protocols supported by the SPARQL endpoints.

6.3 CONVERTORS/PARSERS

The data of construction materials comes from different sources, including relational tables, XML files, spreadsheets, web pages, etc. Convertors are the programs that convert the source-specific data into homogeneous RDF triples and loads it into the semantic store. They provide a systematic processing and archiving methodology for converting RDF triples, constructed from materials data, to populate the Triplestore. It is not possible to write a generic convertor that is capable of converting data from every type of materials data source. The proposed architecture employs the source-specific convertors to reconcile schematic and semantic heterogeneities.

These converters fetch materials data using a variety of source-specific strategies. The selection of these strategies will be influenced by various requirements for scalability and types of data available in the underlying data sources. For example, W3C has recommended R2RML (stands for RDB to RDF mapping language) for interacting with relational tables. Oracle Database 12c supports this feature to seamlessly access data stored in relational tables as RDF triples. Listing 2 shows the PLSQL-based convertor to transform and query data from relational database using RDF view feature.

```
-- PLSQL code to create RDF view on a relational table
BEGIN
  SEM_APIs.Create_RDFview_Model (model_name => 'MTS',
    tables => SYS.ODCIVarchar2List('INV_ITEMS'), prefix => 'http://bimwaste.org.uk/',
    options => 'KEY_BASED_REF_PROPERTY=T');
END;
-- The SQL query to fetch data from the RDF view defined above
SELECT DISTINCT p FROM TABLE (SEM_MATCH( '{?s ?p ?o}', SEM_Models ('MTS'), NULL, NULL, NULL));
```

Listing 2: Creating and querying RDF view on relational tables in Oracle Database 12c

After the data is fetched, next step starts loading data into the Triplestore. Initially, all triples are loaded into a temporary staging area for intermediate processing. The duplicate values and collisions are probed and resolved. A reference structure comprising collision details is constructed to guide transformation and loading of RDF triples into the Triplestore. Unique identifiers are allocated to the RDF triples and data is loaded to the Triplestore. Indexes are rebuilt to reflect the new state to achieve consistency and performance for answering the semantic queries.

6.4 Serialiser

The data exchange is a vital aspect of real-world applications for moving data between different applications. The serialiser enables the data exchange of partial or full RDF triples. It reads the RDF triples stored in the Triplestore based on the user-specified criteria and exports the data into a variety of serialization formats such as XML, N3, Turtle, etc. In terms of the functionality, serialiser generally reverses the operations offered by the convertor/parser as discussed above.

6.5 APPLICATIONS

As discussed earlier, this study is part of efforts to develop the construction waste simulation tool. There are a large number of applications for building materials models in carrying out fascinating analyses. Application layer in the proposed architecture is the primary consumer of this entire materials database. It is pertinent that real applications are more to materials database. With a particular objective, these applications support different functionalities for displaying and analysing the contents. These applications can be developed using various programming languages such as Java, C#, C, Python, etc. Well-articulated interfaces are pivotal to support these applications. Unified querying is supported through semantic querying service to access the contents of the semantic repository as shown in Figure 3. Applications can either exploits the SPARQL libraries or invoke the functions by calling semantic web services directly to query and manage RDF data in materials database.

7 A CASE STUDY OF MATERIALS DATABASE IMPLEMENTATION

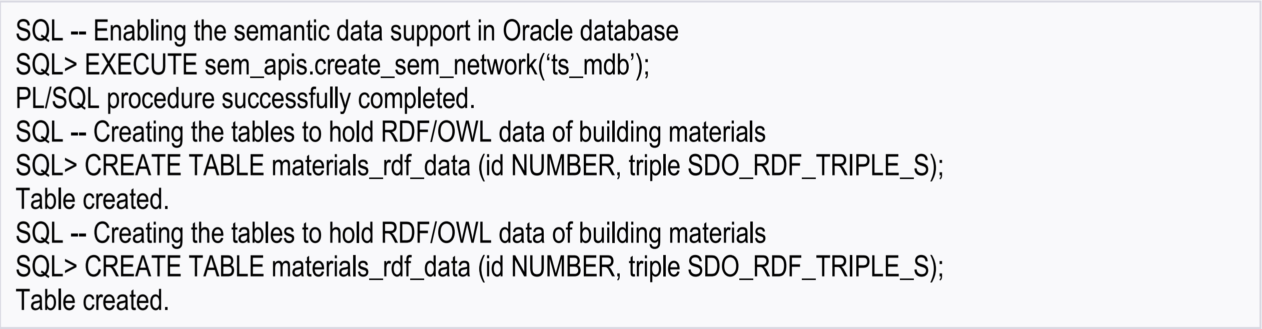
This section discusses the implementation details of the proposed architecture for building materials information modelling. Oracle Database 11g is chosen for this implantation. Semantic query service is implemented to support manipulating the RDF data stored in Oracle Database. This study exploits the Oracle supplied built-in function SEM_MATCH function for querying the Triplestore. Besides, the Jena adapter is implemented to store and query semantic data and ontologies through SPARQL endpoints.

The implementation of the materials database is made up of two steps. Firstly, Oracle Database is configured as the semantic repository and secondly materials information models (expressed as RDF triples) have been created and queried from the semantic store. These steps are described in the following sections.

7.1 SETTING UP SEMANTIC REPOSITORY

A tablespace is required in Oracle Database to hold the actual contents of the semantic store. Tablespaces are the logical structure in Oracle storage management hierarchy (Murray, 2016;

Oracle, 2015). We have created a tablespace for storing the RDF data and ontologies for building materials. A database user schema and a table are also created in the tablespace. Finally, RDF models of the construction materials are inserted in this table, for the Subject, Predicate, and Object of the RDF triples. Indexes are created to execute semantic queries in a reasonable response time. Listing 3 shows SQL statements to configure the Oracle Database 12c as the semantic store.

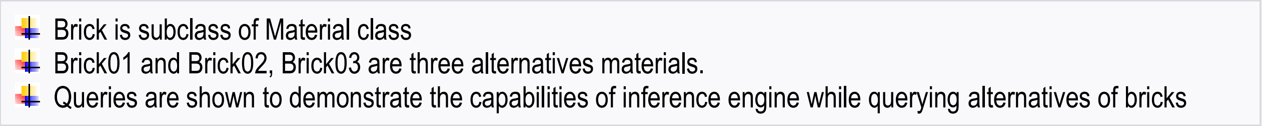


```
SQL -- Enabling the semantic data support in Oracle database
SQL> EXECUTE sem_apis.create_sem_network('ts_mdb');
PL/SQL procedure successfully completed.
SQL -- Creating the tables to hold RDF/OWL data of building materials
SQL> CREATE TABLE materials_rdf_data (id NUMBER, triple SDO_RDF_TRIPLE_S);
Table created.
SQL -- Creating the tables to hold RDF/OWL data of building materials
SQL> CREATE TABLE materials_rdf_data (id NUMBER, triple SDO_RDF_TRIPLE_S);
Table created.
```

Listing 3: Configuring Oracle Database 12c as the Semantic Store

7.2 POPULATING AND QUERYING THE SEMANTIC REPOSITORY

This section demonstrates the knowledge representation for building materials information modelling. To showcase the flexibility of the proposed architecture, a subset of the knowledge relating to building materials (see Listing 4) is described as the RDF triples (see Listing 5) in the Triplestore. Next, the rules index is created for the Oracle supplied built-in RFDS and OWLPrime rule-bases (see Listing 6). Afterward, the semantic query is executed on Triplestore to fetch materials alternatives of the Brick01. It is evident that these integrated rule-bases are unable to capture our notions of bi-directional associativity and transitivity as explained in section 3.



- Brick is subclass of Material class
- Brick01 and Brick02, Brick03 are three alternatives materials.
- Queries are shown to demonstrate the capabilities of inference engine while querying alternatives of bricks

Listing 4: A Subset of Domain Knowledge of Building Materials

```
SQL> -- Defining Material Class
SQL> INSERT INTO materials_rdf_data VALUES(1,SDO_RDF_TRIPLE_S('materials','mdb#Material',
'rdf#type','rdfs#Class'));
1 row created.
SQL> -- Defining Brick is a Subclass of Material
SQL> INSERT INTO materials_rdf_data VALUES (2, SDO_RDF_TRIPLE_S('materials','mdb#Brick',
'rdfs#subClassOf', 'mdb#Material'));
1 row created.
SQL> -- Defining hasAlternative as Object Property
SQL> INSERT INTO materials_rdf_data VALUES (3, SDO_RDF_TRIPLE_S('materials',
'mdb#hasAlternative','rdf#type',
 'owl#ObjectProperty'));
1 row created.
SQL> -- Defining the instance of the Brick named Brick01
SQL> INSERT INTO materials_rdf_data VALUES (6, SDO_RDF_TRIPLE_S('materials', 'mdb#Brick01', 'rdf#type>',
'mdb#Brick'));
1 row created.
SQL> -- Capturing the fact that Brick01 is the alternative of Brick02
SQL> INSERT INTO materials_rdf_data VALUES(8, SDO_RDF_TRIPLE_S('materials','mdb#Brick01',
'mdb#hasAlternative','mdb#Brick02'));
1 row created.
```

Listing 5: SQL Statements to Describe Knowledge for Building Materials in Triplestore

```
SQL> --Querying the alternative materials of Brick01
SQL> SELECT o Alternatives FROM TABLE(SEM_MATCH( '{:Brick01 :hasAlternative ?o}',
SEM_MODELS('materials'),
SEM_Rulebases('RDFS','OWLPrime'), SEM_ALIASES(SEM_ALIAS('','http://www.bimwaste.org.uk/')), null));

ALTERNATIVES
--------------------------------------------------------------------------------
http://www.bimwaste.org.uk/Brick02
```

Listing 6: SQL Statements to Query Alternative Materials of Brick01

To enable the Triplestore to consider the bidirectional associativity and transitivity, custom rule-base is created. A rule is defined to capture the fact that if the pair of (x, y) is related by hasAlternative relationship, it shall also hold for the reverse pair (y, x). Listing 7 shows the definition of rules in the rule-base.

```
SQL> --Defining rule to state hasAlternative is true in reverse direction
SQL> INSERT INTO mdsys.semr_materials_rb VALUES('reverse_alt_rule','(?x :hasAlternative ?y)', NULL,
'(?y :hasAlternative ?x)', SEM_ALIASES(SEM_ALIAS('','http://www.bimwaste.org.uk/')));
1 row created.
```

```
SQL> --Defining transitive rule to state of (x,y) (y,z) are related by hasAlternative, it says (x,z) also holds.
SQL> INSERT INTO mdsys.semr_materials_rb VALUES('transitive_alt_rule','(?x :hasAlternative ?y) (?y
:hasAlternative ?z)',
NULL,'(?x :hasAlternative ?z)', SEM_ALIASES(SEM_ALIAS('','http://www.bimwaste.org.uk/')));
1 row created.
```

Listing 7: Configuring Rule-base for Bi-directional Associativity and Transitivity

Rules-index is refreshed and the semantic query (see Listing 8) is executed for alternative materials for Brick01 type, which returns all alternatives for Brick01. In this way, semantic store is made capable to retrieve all the brick alternatives using reification and inference. This example demonstrates the way data completeness is achieved in the proposed materials database using Web of Data technologies.

```
SQL> -- Querying the Alternatives of Brick01
SQL> SELECT a Alternatives FROM TABLE(SEM_MATCH( '{:Brick01 :hasAlternative ?a}',
SEM_MODELS('materials'),
SEM_Rulebases('RDFS','OWLPrime','materials_rb'),
  6  SEM_ALIASES(SEM_ALIAS('','http://www.bimwaste.org.uk/')),
  7  null));

ALTERNATIVES
--------------------------------------------------------------------------------
http://www.bimwaste.org.uk/Brick02
http://www.bimwaste.org.uk/Brick03
http://www.bimwaste.org.uk/Brick01
```

Listing 8: SQL Query for the Alternatives of Brick01 based on User-defined Rules

8 THE RELEVANCE OF MATERIALS DATA FOR BUILDING WASTE PERFORMANCE ANALYSIS (BWA)

Building materials database in a vacuum is useless. The ultimate goal of the construction materials database is to support the development of construction waste simulation tool. Construction waste simulation tool is intended to provide novel techniques for waste estimation and minimization. During the waste evaluation, the Tool will read BIM models of the design and estimate the types and amounts of waste arising from various building elements. Besides, the

Tool will also provide insights to reduce construction waste by specifying alternative materials. Those materials which have lesser waste output. The Tool will visualize the waste output of building elements and highlight ones producing massive amounts of waste. The designers will further investigate these elements and try to use different materials and strategies with the lesser waste output. In this way, construction waste will be reduced through smart materials specification. This ability of the Tool to query alternatives for a given material is of utmost relevance. The Tool will execute SQL statements (as shown in Listing 8) from the BIM authoring tools such as Revit, MicroStation, etc. and present the designers with more materials choices to reduce waste output.

9 CONCLUSIONS

Describing building materials is the crucial first step towards the development of a simulation tool for BWA. Building materials dataset poses particular data representation challenges that are beyond the representational capabilities of the existing standards/ontologies. These standards have been primarily developed for classifying products and building materials for e-commerce purposes. Since semantic web is well-known for resolving similar data and knowledge representation challenges faced by enterprise applications from diverse fields, we have employed the technique of ontologies—a vital semantic web technology—in our research to describe the highly specific data of the properties of building materials. Consequently, a huge dataset comprising RDF/OWL models is created. Surprisingly, no such repository could be exploited to store RDF/OWL models of the building materials. In this regards, a highly flexible and data-agnostic architecture of the materials database is proposed. This architecture is designed to fulfil the specialized requirements imposed by the building materials datasets. Detail implementation of the RDF/OWL models and building materials database is discussed. Protégé ontology engineering tool, Oracle Database 12c based semantic repository, and SEM_MATCH based semantic querying are explained. The approach is limited in the sense that it is using the relational database to store the massive dataset of building materials database, but the architecture is designed to support the emerging NoSQL based RDF stores. In the future, we are planning to migrate this dataset to Oracle RDF based NoSQL database such that the SPARQL queries can be executed from MapReduce jobs for high-performance computing. This study contains detailed insights and technical guidelines for the researchers and software engineers interested in developing the semantic repositories in similar kind of simulation applications.

10 REFERENCES

[1] L. L. Ekanayake and G. Ofori, "Building waste assessment score: design-based tool," *Building and Environment,* vol. 39, no. 7, pp. 851-861, 2004.

[2] L. O. Oyedele, R. Martin, v. M. Jason, A. Ashraf, J. E. Obas and E. Amira, "Reducing waste to landfill in the UK: identifying impediments and critical solutions," *World Journal of Science, Technology and Sustainable Development,* vol. 10, no. 2, pp. 131-142, 2013.

[3] A. Keys and A. Baldwin, "Designing to encourage waste minimisation in the construction industry," Department of Civil & Building Engineering, Loughborough University, Licestershire UK, 2010.

[4] L. Davis, "Designing out Waste: A design team guide for buildings," 2012. [Online]. Available: http://www.modular.org/marketing/documents/DesigningoutWaste.pdf. [Accessed 07 12 2014].

[5] Z. Liu, M. Osmani and P. Demian, "The potential use of BIM to aid construction waste minimisation," Sophia Antipolis, France, 2011.

[6] O. O. Faniran and G. Caban, "Minimizing waste on construction project sites," *Engineering, Construction and Architectural Management,* vol. 5, no. 2, pp. 182-188, 1998.

[7] M. Osmani, J. Glass and A. Price, "Architect and contractor attitudes to waste minimisation," in *Proceedings of the Institution of. Civil Engineers : Waste and Resource Managament*, London. UK, 2006.

[8] W. Lu and H. Yuan, "Investigating waste reduction potential in the upstream," *Renewable and Sustainable,* pp. 804-811, 2013.

[9] S. Azhar, J. Brown and R. Farooqui, "BIM-based sustainability analysis: An evaluation of building performance analysis software," *In Proceedings of the 45th ASC Annual Conference,* vol. 4, no. 1, pp. 30-40, 2009.

[10] V. Bazjanac, "Early lessons from deployment of IFC compatible software," California, 2002.

[11] C. Eastman, P. Teicholz, R. Sacks and K. Liston, BIM Handbook: A Guide to Building

Information Modeling for Owners, Managers, Designers, Engineers and Contractors, 2 ed., New York: John Wiley & Sons, Inc, 2011.

[12] A. Grilo and R. Jardim-Goncalves, “Value proposition on interoperability of BIM and collaborative working environments,” *Automation in Construction,* pp. 522-530, 2010.

[13] D. B. Lenat and E. A. Feigenbaum, “On the Threshold of Knowledge,” *International Joint Conference on Artificial Intelligence,* vol. 87, no. 2, pp. 1173-1182, 1991.

[14] A. Radinger, B. Rodriguez-Castro, A. Stolz and M. Hepp, “BauDataWeb: the Austrian building and construction materials market as linked data,” in *Proceedings of the 9th International Conference on Semantic Systems*, 2013.

[15] K. Karam, K. Gunwoo, Y. Donghee and Y. Jungho, “Semantic material name matching system for building energy analysis,” *Automation in Construction,* vol. 30, no. 0, pp. 242 - 255, 2013.

[16] M. Hepp, “GoodRelations: An Ontology for Describing Products and Services Offers on the Web,” in *Proceedings of the 16th International Conference on Knowledge Engineering and Knowledge Management (EKAW2008)*, Acitrezza, Italy, 2008.

[17] M. Hepp and J. de Bruijn, “GenTax: A generic methodology for deriving OWL and RDF-S ontologies from hierarchical classifications, thesauri, and inconsistent taxonomies,” in *The Semantic Web: Research and Applications*, Springer, 2007, pp. 129-144.

[18] M. Hepp, “ Products and Services Ontologies: A Methodology for Deriving OWL Ontologies from Industrial Categorization Standards,” *Int'l Journal on Semantic Web & Information Systems (IJSWIS),* vol. 2, no. 1, pp. 72-99, 2006.

[19] M. Hepp, “eClassOWL: A fully-fledged products and services ontology in OWL,” in *Poster Proceedings of ISWC2005. Galway*, Galway, 2005.

[20] M. M. Farag, “Quantitative methods of materials selection,” in *Handbook of materials selection*, M. Kutz, Ed., United States of America, John wiley & Sons, 2002, pp. 1-24.

[21] S. J. Rysavy, B. Dennis and D. Valerie, “DIVE: A Graph-Based Visual-Analytics Framework for Big Data,” *IEEE Computer Graphics and Applications,* vol. 34, no. 2, pp. 26-37, 2014.

[22] I. Horrocks, P. F. Patel-Schneider and F. Van Harmelen, “From SHIQ and RDF to OWL: The making of a web ontology language,” *Web semantics: science, services and agents on the World Wide Web,* vol. 1, no. 1, pp. 7-26, 2003.

[23] I. Horrocks, “Ontologies and the semantic web,” *Communications of the ACM,* vol. 51, no. 12, pp. 58-67, 2008.

[24] A. S. Sidhu, T. S. Dillon, E. Chang and B. S. Sidhu, “Protein Ontology Development using OWL,” *OWLED,* vol. 188, 2005.

[25] P. Shvaiko and J. Euzenat, “A survey of schema-based matching approaches,” *In Journal on Data Semantics,* vol. 4, no. 1, pp. 146-171, 2005.

[26] T. R. Gruber, “Toward principles for the design of ontologies used for knowledge sharing,” *International journal of human-computer studies,* vol. 43, no. 5, pp. 907-928, 1993.

[27] R. Cyganiak, D. Wood and M. Lanthaler, “RDF 1.1 Concepts and Abstract Syntax,” 25 02 2004. [Online]. Available: http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/Overview.html. [Accessed 27 12 2014].

[28] R. W. Group, “RDF - Resource Description Framework,” 2004. [Online]. Available: http://www.w3.org/RDF/. [Accessed 15 May 2014].

[29] S. Weibel, J. Kunze, C. Lagoze and M. Wolf, “Dublin Core Metadata for Resource Discovery,” 1998. [Online]. Available: http://www.hjp.at/doc/rfc/rfc2413.html. [Accessed 1 December 2014].

[30] D. Brickley and R. Guha, “RDF Schema 1.1,” 2004. [Online]. Available: http://www.w3.org/TR/rdf-schema/. [Accessed 1 December 2014].

[31] J. Heflin, J. Hendler and S. Luke, “SHOE: A knowledge representation language for internet applications,” 1999.

[32] D. Fensel, F. Van Harmelen, I. Horrocks, D. L. McGuinness and P. F. Patel-Schneider, “OIL: An ontology infrastructure for the semantic web,” *IEEE Intelligent Systems,* vol. 16, no. 2, pp. 38-45, 2001.

[33] D. L. McGuinness, R. Fikes, J. Hendler and L. A. Stein, “DAML+ OIL: an ontology language for the Semantic Web,” *IEEE Intelligent Systems,* vol. 17, no. 5, pp. 72-80, 2002.

[34] S. Bechhofer, “OWL: Web ontology language.,” *In Encyclopedia of Database Systems,* pp. 2008-2009, 2009.

[35] B. Motik, F. P. Peter, B. P. Conrad Bock, F. Achille and H. Peter, “Owl 2 web ontology language: Structural specification and functional-style syntax.,” *W3C recommendation,* vol. 27, no. 65, p. 159, 2009.

[36] D. Tsarkov and I. Horrocks, “FaCT++ description logic reasoner: System description.,” *In Automated reasoning,* pp. 292-297, 2006.

[37] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur and Y. Katz, “Pellet: A practical owl-dl reasoner.,” *Web Semantics: science, services and agents on the World Wide Web,* vol. 5, no. 2, pp. 51-53, 2007.

[38] V. Haarslev and R. Möller, “Racer: A Core Inference Engine for the Semantic Web.,” *EON,* vol. 87, 2003.

[39] R. Stevens, P. Baker, S. Bechhofer, G. Ng, A. Jacoby, N. W. Paton, G. C. A. and A. Brass, “TAMBIS: transparent access to multiple bioinformatics information sources.,” *Bioinformatics,* vol. 16, no. 2, pp. 184-186, 2000.

[40] C. David, C. Olivier and B. Guillaume, “A survey of RDF storage approaches,” *ARIMA Journal,* vol. 15, pp. 11-35, 2012.

[41] P. Cudre-Mauroux, I. Enchev, S. Fundatureanu, P. Groth, A. Haque, A. Harth, F. L. Keppmann, D. Miranker, J. F. Sequeda and M. Wylot, “NoSQL databases for RDF: An empirical evaluation,” *The Semantic Web - ISWC 2013,* pp. 310-325, 2013.

[42] M. Stonebraker, “SQL databases v. NoSQL databases,” *Communications of the ACM,* vol. 53, no. 4, pp. 10-11, 2010.

[43] J. Pokorny, “NoSQL databases: a step to database scalability in web environment,” *International Journal of Web Information Systems,* vol. 9, no. 1, pp. 69-82, 2013.

[44] M. Buerli and C. P. S. L. Obispo, “The Current State of Graph Databases,” *Department of Computer Science, Cal Poly San Luis Obispo, mbuerli@ calpoly. edu,* vol. 32, no. 3, pp. 67-83, 2012.

[45] N. Leavitt, “Will NoSQL databases live up to their promise?,” *Computer,* vol. 43, no. IEEE, pp. 12-14, 2010.

[46] I. MongoDB, “MongoDB - Agile and Scalable Document Base,” 2010. [Online]. Available: http://www.mongodb.org/. [Accessed 15 May 2014].

[47] Oracle, “Oracle NoSQL Database,” 2010. [Online]. Available: http://www.oracle.com/uk/products/database/nosql/overview/index.html. [Accessed 15 May 2014].

[48] Yahoo, “Apache Cassandra,” 2011. [Online]. Available: http://cassandra.apache.org/. [Accessed 17 May 2014].

[49] Voldemort, “Project Voldemort - A distributed database,” 2009. [Online]. Available: http://www.project-voldemort.com/voldemort/. [Accessed 17 May 2014].

[50] K. Panos and D. I. Danai, “Survey Regarding Control and Reduction of Construction Waste,” Lima, Peru, 2012.

[51] C. Batini, M. Lenzerini and S. B. Navathe, “A comparative analysis of methodologies for database schema integration.,” *ACM computing surveys (CSUR),* vol. 18, no. 4, pp. 323-364, 1986.

[52] A. Doan, N. F. Noy and A. Y. Halevy, "Introduction to the special issue on semantic integration.," *ACM SIGMOD Record,* vol. 33, no. 4, pp. 11-13, 2004.

[53] A. Halevy, A. Rajaraman and J. Ordille, "Data integration: the teenage years.," *In Proceedings of the 32nd international conference on Very large data bases,* pp. 9-16, 2006.

[54] C. Batini, M. Lenzerini and S. B. Navathe, "A comparative analysis of methodologies for database schema integration," *ACM computing surveys (CSUR),* vol. 18, no. 4, pp. 323-364, 1986.

[55] A. Doan, N. F. Noy and A. Y. Halevy, "Introduction to the special issue on semantic integration," *ACM SIGMOD Record,* vol. 33, no. 4, pp. 11-13, 2004.

[56] A. Halevy, A. Rajaraman and J. Ordille, "Data integration: the teenage years," *In Proceedings of the 32nd international conference on Very large data bases,* pp. 9-16, 2006.

[57] J. Kim, S. Kim and L. Tang, "Case Study on the Determination of Building Materials Using a Support Vector Machine," *Journal of Computing in Civil Engineering,* vol. 28, no. 2, pp. 315-326, 2012.