

Mapping of extensible markup language-to-ontology representation for effective data integration

Su-Cheng Haw¹, Lit-Jie Chew¹, Dana Sulisty Kusumo², Palanichamy Naveen¹, Kok-Why Ng¹

¹Faculty of Computing and Informatics, Multimedia University, Cyberjaya, Malaysia

²School of Computing, Telkom University, Bandung, Indonesia

Article Info

Article history:

Received Mar 3, 2022

Revised Aug 17, 2022

Accepted Sep 15, 2022

Keywords:

Mapping rules

Mapping scheme

Ontology representation

Semantic relationship

Extensible markup language to ontology

ABSTRACT

Extensible markup language (XML) is well-known as the standard for data exchange over the internet. It is flexible and has high expressibility to express the relationship between the data stored. Yet, the structural complexity and the semantic relationships are not well expressed. On the other hand, ontology models the structural, semantic and domain knowledge effectively. By combining ontology with visualization effect, one will be able to have a closer view based on respective user requirements. In this paper, we propose several mapping rules for the transformation of XML into ontology representation. Subsequently, we show how the ontology is constructed based on the proposed rules using the sample domain ontology in University of Wisconsin-Milwaukee (UWM) and mondial datasets. We also look at the schemas, query workload, and evaluation, to derive the extended knowledge from the existing ontology. The correctness of the ontology representation has been proven effective through supporting various types of complex queries in simple protocol and resource description framework query language (SPARQL) language.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Su-Cheng Haw

Faculty of Computing and Informatics, Multimedia University

Persiaran Multimedia, 63100 Cyberjaya, Selangor, Malaysia

Email: sucheng@mmu.edu.my

1. INTRODUCTION

Extensible markup language (XML) has been widely used as the data exchange format over the internet [1], [2]. Big data analytics is the trend in various industries to boost their industrial performance, and in fact, XML data format usually forms the basis of data streaming used in the analytical process [3], [4]. However, XML data represent the data only at the syntactic level. On the other hand, ontology is a knowledge representation that established a shared vocabulary, conceptualizations and model domain knowledge for various applications [5]–[8]. Ontology is often expressed in ontology web language (OWL) format.

Several ontology generation (also known as ontology mapping) techniques existed to transform the gap between the syntactical XML and semantical OWL representation [9]–[11]. Ontology enrichment is also an objective of the transformation [12]. It is to extend the ontology by adding the elements and constructor (class, object attributes, data types, concept relations, axioms, properties). In addition, the ontology population process adds individuals or attributes to available individuals from XML data to the ontology representation.

In general, the mapping approaches from XML to ontology representation can be grouped into two main categories: the instance approach and the validation approach [13]. The instance approach intends to convert XML documents directly to ontology representation without using schema knowledge. Most of these approaches generate new ontologies from only XML content by using XML path language (XPath) query

language based on path expression to navigate each path of the respective node in the XML. Klein presented the first mapping tool to translate XML documents directly to an ontology language such as resource description framework (RDF) or OWL [14]. He proposed a method to transform the ambiguous XML data into the RDF statements on a one-way mapping basis. Bohring and Auer [15] proposed a framework to map XML into OWL, which is built on top of the XML instance document to possibly generate XML schema definition (XSD), and finally transform it into OWL. O'Connor and Das [16] proposed a domain-specific language called XML master. It is developed by using OWL syntax and XPath query language.

The validation approach refers to the approach that generates ontology from the schema. Both XSD and document type definition (DTD) are the two major schemas that are being used today. However, DTDs are not in XML format, which DTDs do not support 'namespace' while XSD does provide more advanced features. These approaches make use of the advantage of XSD, which contains the defined elements and XML types of simple or complex data. Ferdinand *et al.* [17] proposed a semi-automated approach named ontology web language mapping (OWLMAP), which is constructed based on some mapping rules to handle complex type, simple type, attribute, element, element type, substitution group and so on. The XML schema to ontology web language (XS2OWL) [18] approach targets to support the interoperability between the XML and OWL environment. The tool automatically transforms the XSD as input into: i) main ontology which is directly reflected by the defined transformation rules and ii) mapping ontology. The mapping ontology is used to keep the radio-frequency identifications (rf:IDs) of the OWL constructs of the main ontology which cannot be generated directly from the main ontology. There are four classes of mapping ontology which are complex type info type, element info type, data type property info type and data type property info type. Bedini *et al.* [19] developed a prototype named Janus, which consists of 40 transformation rules to map the XSD constructs to ontology (OWL2-RL) constructs. Their approach managed to minimize the information loss during the transformation process. As an example, the construction 'restriction', derived from the restriction of a simple type, allows the creation of several simple types from the simple predefined types in XSD. However, the transformation is designed based on an application domain to compute statistical analysis of the business to business (B2B) domain, which becomes the constraint of this approach. An efficient XML to OWL converter (EXCO) [20] is a tool, which could manage both enrichment and population for XML documents into OWL by covering both the internal and external references. Thuy *et al.* [21] proposed s-trans, which transforms XML healthcare data into ontology based on extraction of the XML schema with added description of the semantic knowledge. Subsequently, in another research, Thuy *et al.* [22] proposed to reduce the redundancy of data resulting from duplicate elements in XML schema by measuring the similarity between these duplicates before the transformation process.

More recently, Shapkin and Shumsky [23] proposed modularizing the transformation from XML to ontologies based on some designed templates, which are constructed based on class and property values. Singapogu *et al.* [24] proposed the mapping by looking at the XML schema elements to automatically structure and represent it in the first draft of ontology. Subsequently, some part-of-speech tagging method is employed to extract the domain knowledge to enrich the refinement of ontology. Jounaidi and Bahaj [25] formulated some rules for mapping the XML schema into ontology representation. This mapping also covers the relationships between the nodes to ensure the structure is maintained. They also proposed canonical data model (CDM) to transform XML Schema into OWL ontology [26]. Hacherouf *et al.* [27] proposed patterns identification for XSD conversion to OWL (PIXCO), a method based on formal concept analysis (FCA) to model the transformation patterns. There are several processes involved including the constructions of XML schema, the transformation patterns identified and the OWL modelling.

From the review, we observed that EXCO [20] tool is stable and has enriched information. Nevertheless, EXCO can be further improved to support some advanced operators and restrictions. Our proposed framework extended EXCO to add some new functionalities as described in the next section.

2. THE PROPOSED METHOD

Figure 1 depicts the overall framework of our proposed approach. In this method, a validation approach is used in the proposed solution. At first, if there is no XSD nor DTD available as input, it will be generated automatically from the XML documents. The generation of the target OWL is composed of a few stages as elaborated next.

2.1. Stage 1: initial transformation step

The trang application programming interface (API) is used in the transformation between an XML document and XML schema to define the restriction on the XML structure. Trang API is an open source API for working with XML files to convert the XML schema into XSD schema format. In addition, trang is also capable to infer a schema from an XML document itself if the schema is not present.

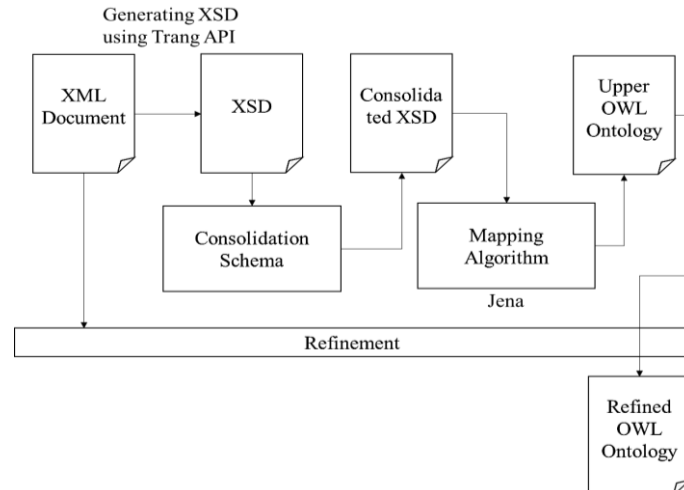


Figure 1. The overview of the framework

2.2. Stage 2: resolving the conflict

Next, to resolve the internal and external references, consolidation mapping is adopted from [19] method. First, i) collecting schema files, XSD schemas are collected to get the reference of their location into a hash table. The namespace and references are saved to avoid duplication; ii) merging schemas files, namespace prefixes of the saved schemas are unified to merge them into the main schema file; and iii) reorganizing schema, to reorganize the internal references within the main schema file. The referred elements will be simply appended into the node, and this is done hierarchically through the descendant. Finally, the useless element is eliminated.

2.3. Stage 3: automated transformation

The automatic transformation is handled through an algorithm developed based on the transformation model to map the consolidated output construct into the ontology web language description logics (OWL-DL) construct. The process is done without any user intervention. This process will ease the user with the initial mapping constructed.

2.4. Stage 4: refinement stage

Refinement of the generated ontology and mapping of bridges. The invalid mapping can be cleaned and reconstructed. Mapping of ontology which cannot be generated directly from the XML schema can be manually mapped using the mapping ontology that keeps the rdf:IDs of the OWL constructs.

3. METHOD

3.1. Translation on UWM dataset

The University of Wisconsin-Milwaukee (UWM) XML document from the University of Washington (UW) database group [28] is used as an example. UWM data are the course data derived from the UWM website. Figure 2(a) shows the partial view of UWM XML document, which contains the series of <course_listing> records and each of them contains the details of the record elements and value. The next step of the ontology generation process is the conversion of the XML document to XML Schema, XSD. The following generated XML-schema is depicted in Figure 2(b). The XSD generated having of the elements, sub elements, and property restrictions like the type of cardinality and also operators of class combinations (union of, complement of, intersection of).

The rules of the generation of OWL constructs are shown in Figure 3. OWL class can be created from xsd: complex types; and xsd: elements which are independent identities. OWL data type property is created from the element which they are the only literal with no attributes as well as the XML attributes. The constraints properties from XML schema like min occurs or max occurs will map as the cardinality constraint in OWL. There is owl: minimum cardinality and owl: maximum cardinality. The inheritance that is shown is a relationship that is derived from XML Schema will be mapped to RDF schema (RDFS): sub class of in OWL. As a similar condition for elements will be mapped to RDFS: sub property of RDF. The compositors of combining elements sequence, all and choice will be mapped into owl: intersection of, owl: union of or

owl: complement of. Lastly, the model group definitions and attribute group definitions are specialisations of complex types since they only contain elements respectively attribute declarations are also mapped to OWL class. The summary of the transformation mapping rules is shown in Figure 3.

```

<root>
<course_listing>
<note>#</note>
<course>216-088</course>
<title>NEW STUDENT ORIENTATION</title>
<credits>0</credits>
<level>U</level>
<restrictions>; ; REQUIRED OF ALL NEW STUDENTS. PREREQ: NONE</restrictions>
<section_listing>
  <section_note></section_note>
  <section>Se 001</section>
  <days>W</days>
  <hours>
    <start>1:30pm</start>
    <end></end>
  </hours>
  <bldg_and_rm>
    <bldg>BUS</bldg>
    <rm>S230</rm>
  </bldg_and_rm>
  <instructor>Gusavac</instructor>
  <comments>9 WKS BEGINNING WEDNESDAY, 9/6/00 </comments>
</section_listing>
<section_listing>
  <section_note></section_note>
  <section>Se 002</section>
  <days>F</days>
  <hours>
    <start>11:30am</start>
    <end></end>
  </hours>
  <bldg_and_rm>
    <bldg>BUS</bldg>
    <rm>S171</rm>
  </bldg_and_rm>
  <instructor>Gusavac</instructor>
  <comments>9 WKS BEGINNING FRIDAY, 9/8/00 </comments>
</section_listing>

```

(a)

```

<xs:schema id="root" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:sche
<xs:element name="root" msdata:IsDataSet="true" msdata:Locale="en-US">
  <xs:complexType>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="course_listing">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="note" type="xs:string" minOccurs="0" />
            <xs:element name="course" type="xs:string" minOccurs="0" />
            <xs:element name="title" type="xs:string" minOccurs="0" />
            <xs:element name="credits" type="xs:string" minOccurs="0" />
            <xs:element name="level" type="xs:string" minOccurs="0" />
            <xs:element name="restrictions" minOccurs="0" maxOccurs="unbounded">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="A" nillable="true" minOccurs="0" maxOccurs="unbounded">
                    <xs:complexType>
                      <xs:simpleContent msdata:ColumnName="A_Text" msdata:Ordinal="1">
                        <xs:extension base="xs:string">
                          <xs:attribute name="HREF" type="xs:string" />
                        </xs:extension>
                      </xs:simpleContent>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="section_listing" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="section_note" type="xs:string" minOccurs="0" />
            <xs:element name="section" type="xs:string" minOccurs="0" />
            <xs:element name="days" type="xs:string" minOccurs="0" />
            <xs:element name="instructor" type="xs:string" minOccurs="0" />
            <xs:element name="comments" type="xs:string" minOccurs="0" />
            <xs:element name="hours" minOccurs="0" maxOccurs="unbounded">

```

(b)

Figure 2. Partial view of the, (a) UWM XML document and the corresponding and (b) generated XSD

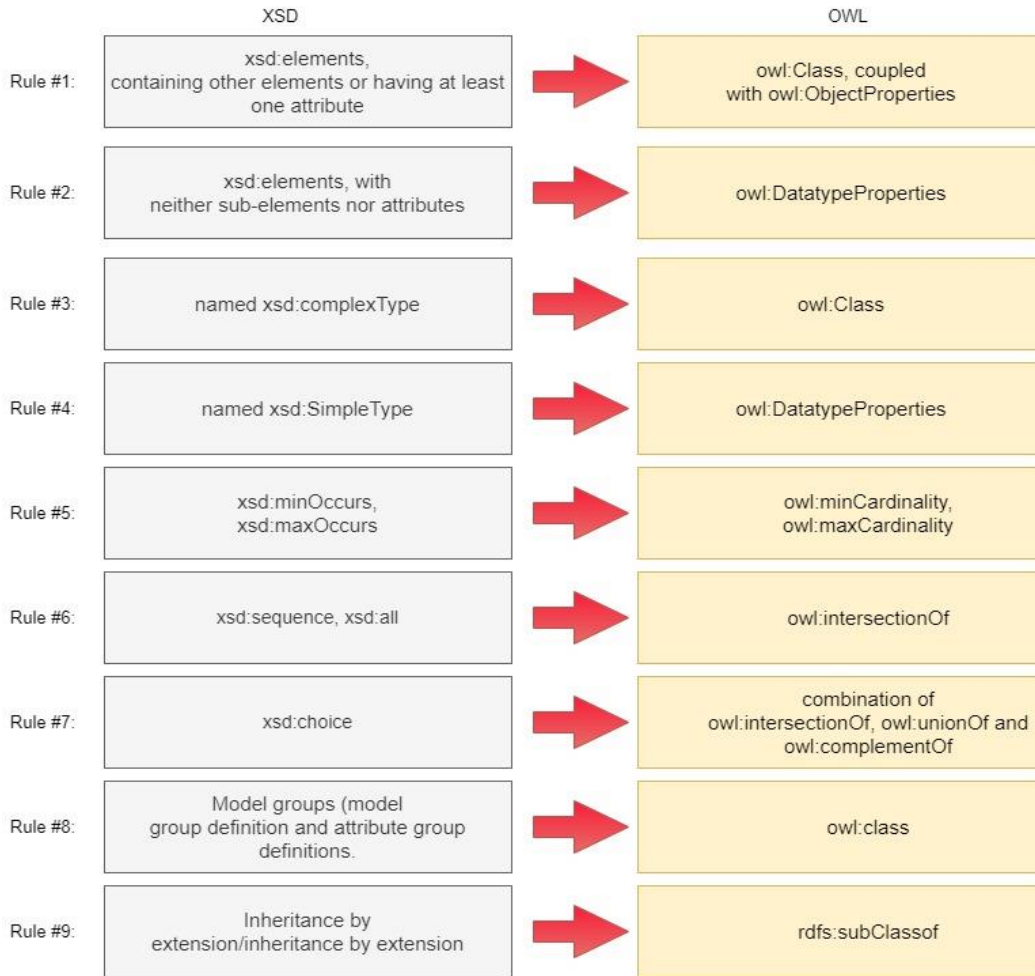


Figure 3. Transformation rules from XSD to OWL

Figure 4 and Table 1 show the OWL classes and properties generated respectively, where by the Table 1(a) lists the object type property while Table 1(b) lists the data type property. During the transformation, two elements with the same name, but on a different level, the property will add “has” prefix for owl: object properties. In addition, rdf:ID will be generated for each instance of the class in order. The generated OWL ontology is shown in Figure 5. This ontology is comprised of seven complex types since seven OWL classes, root, course_listing, restriction, A, section_listing, hours, and bldg_and_rm are created. The couse_listing, section_listing, hours and bldg_and_rm further contains respective properties as shown in Figure 5.

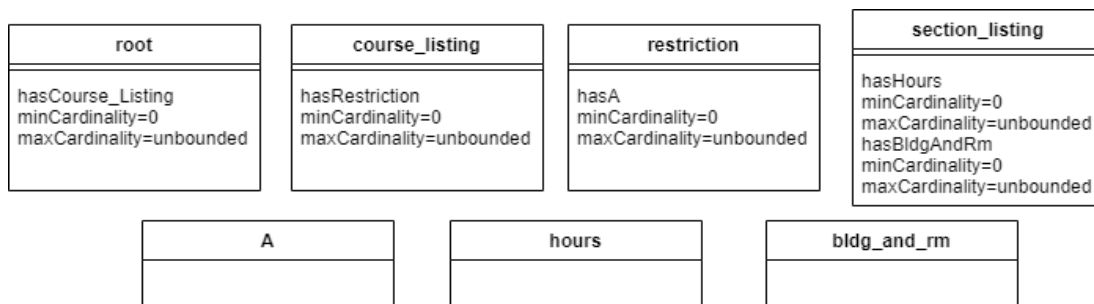


Figure 4. The classes of UWM OWL

Table 1. The properties of UWM OWL (a) object type property and (b) data type property

(a)		
Name	Domain	Range
Has course	Root	Course_listing
Course	Course_listing	Restrictions
Has A	Restrictions	A
Has section listing	Course_listing	Section_listing
Has hours	Section_listing	Hours
Has bldg and rm	Section_listing	Bldg_and_rm

(b)		
Name	Domain	Range
Note	Course_listing	Xs: string
Course	Course_listing	Xs: string
Title	Course_listing	Xs: string
Credits	Course_listing	Xs: string
Level	Course_listing	Xs: string
HREF	A	Xs: string
Section_note	Section_listing	Xs: string
Section	Section_listing	Xs: string
Days	Section_listing	Xs: string
Instructors	Section_listing	Xs: string
Comments	Section_listing	Xs: string
Start	Hours	Xs: string
End	Hours	Xs: string
Bldg.	Bldg_and_rm	Xs: string
Rm	Bldg_and_rm	Xs: string

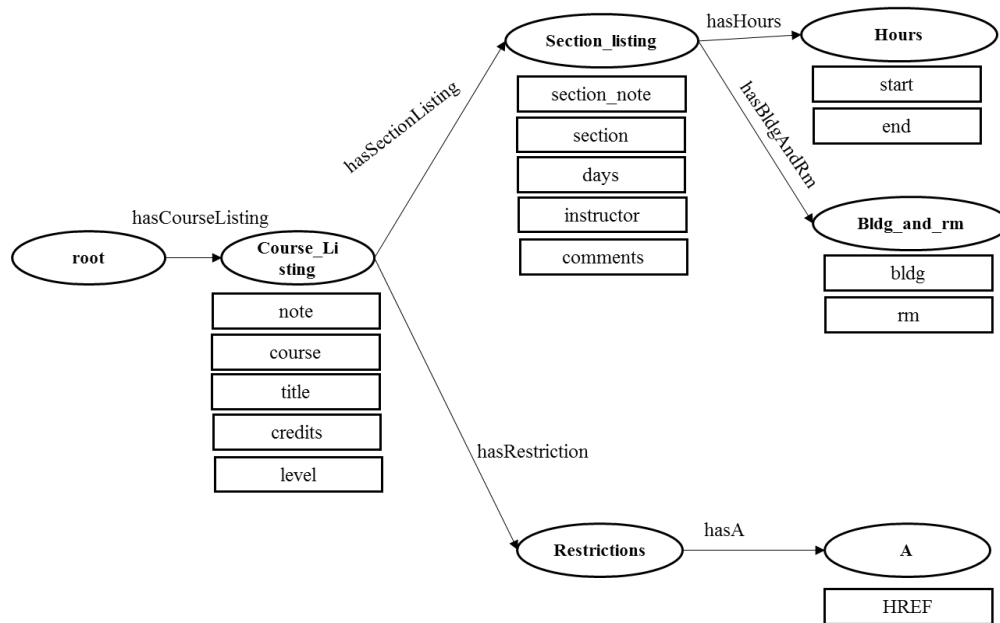


Figure 5. The generated UWM OWL ontology

In the next section, the method of ontology population that we adopted is by looking at the XML instances and XSD files as inputs. XSD documents are acting as the reference to translate the XML instance to the OWL ontology. The snippet as follows shows an example of the transformation of XML elements to instances according to the OWL model. The data type property is represented as follows. The extracted model of OWL ontology constructed composed of: i) classes for concept definition; ii) object properties for object relationship; and iii) data type properties for the relationship between object and data values.

```

<course_listing rdf:id= " id11234544 " >
  <hasSectionListing rdf:id= " #id2213444 " />
</ course_listing >

<section_listing rdf:id="id2213444">

```

```

<section_note rdf:datatype="&xs:string"> </name>
<section rdf:datatype="&xs:string">Se 001</name>
<days rdf:datatype="&xs:string">R</name>
<instructor rdf:datatype="&xs:string">Silberg</name>
<comments rdf:datatype="&xs:string"></name>
</ section_listing >

```

3.2. Translation on mondial dataset

In addition, the same XML-OWL transformation is applied to the mondial XML document. Tables 2 and Table 3 show the OWL classes and properties generated. Table 3(a) lists the object type property while Table 3(b) lists the data type property. The generated mondial OWL ontology is depicted in Figure 6.

Table 2. The classes of mondial OWL

Name	Constraints
Mondial	Has continent, min cardinality=0, max cardinality=unbounded
Continent	Has country, min cardinality=0, max cardinality=unbounded
Country	Has city, min cardinality=0, max cardinality=unbounded Has ethnicgroups, min cardinality=0, max cardinality=unbounded Has religions, min cardinality=0, max cardinality=unbounded Has encompassed, min cardinality=0, max cardinality=unbounded Has border, min cardinality=0, max cardinality=unbounded
City	Has population, min cardinality=0, max cardinality=unbounded
Population	-
Ethnicgroups	-
Religions	-
Encompassed	-
Border	-

Table 3. The properties of mondial OWL, (a) object type property and (b) data type property

(a)

Name	Domain	Range
Has continent	Mondial	Continent
Has country	Mondial	Country
Has city	Country	City
Has population	City	Population
Has ethnicgroups	Country	Ethnicgroups
Has religions	Country	Religions
Has encompassed	Country	Encompassed
Has border	Country	Border

(b)

Name	Domain	Range
Name	City	Xs: string
	Continent	Xs: string
	Country	Xs: string
Id	Continent	Xs: string
	City	Xs: string
Population	City	Xs: int
	Country	Xs: int
Latitude	City	Xs: double
Longitude	City	Xs: double
Percentage	Ethnicgroups	Xs: int
	Religions	Xs: int
	Encompassed	Xs: int
Length	Length	Xs: int
Continent	Encompassed	Xs: int
Country	Border	Xs: string
Gdp_total	Country	Xs: int
Datacode	Country	Xs: string
Population_growth	Country	Xs: double
Car_code	Country	Xs: string
Indep_date	Country	Xs: string
Infant_mortality	Country	Xs: double
Government	Country	Xs: string
Inflation	Country	Xs: int
Gdp_agri	Country	Xs: int
Total_area	Country	Xs: int

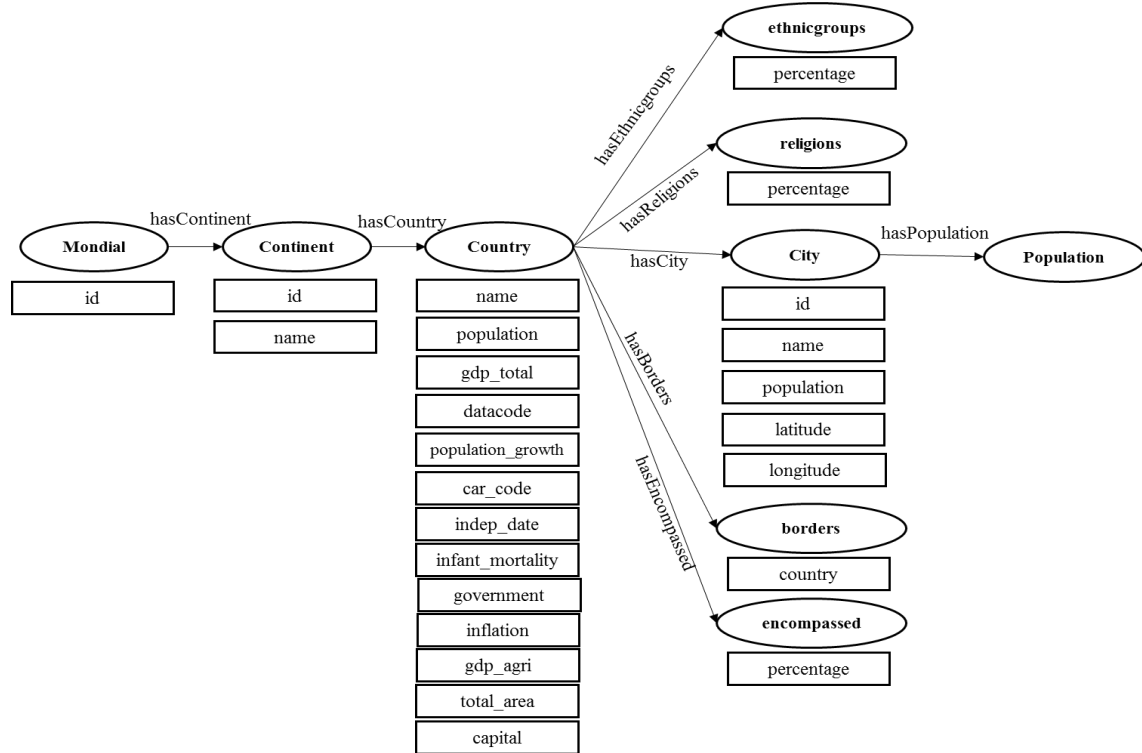


Figure 6. The generated UWM OWL ontology

4. RESULTS AND DISCUSSION

The implementation of the proposed approach has been applied to several XML datasets from different domains including UWM and mondial datasets. The evaluation of the final ontology is done by comparing the semantics captured in the defined ontologies with the semantics captured in the automatic generation. Based on the comparison shown, the semantics captured manually is shown to be the same as the automatic transformation result. The correctness of the ontology representation has been proven by the reflection of the query result with manual verification. Some examples designed query tests were executed towards the constructed UWM and mondial ontology by using simple protocol and resource description framework query language (SPARQL) playground (standalone multi-platform web application) [29].

4.1. Query results on UWM dataset

Two queries were executed to check the correctness of the number of returned results on UWM ontology representation as compared with the query retrieved from the XML dataset itself. Figure 7 shows the first query, query 1, which list the course_listing with credit of 7. Figure 8 depicts query 2, which list the number of sections of each course group by credit. From the number of returned results, it shows that the ontology constructed via our mapping scheme is correct.

Query 1: Select course_listing with credit of 7.

```
select ?course_listing ?credit where {
  ?course mdlo:credit ?credit .
  FILTER (?credit = "7")
}
```

Query Result:

course	credit
367-221	7
367-412	7
367-433	7
367-411	7
367-901	7
410-222	7
410-234	7
410-295	7

Figure 7. Test case on Query 1 on UWM dataset

Query 2: Get the number of sections of each course group by credit

```
select ?course_listing (COUNT(?course_listing) as ?courseCount) where {
  ?course_listing rdf:type uwmo:course_listing .
  ?course_listing uwmo:credit ?credit .
}
GROUP BY ?credit
```

Query Result:

	courseCount
	20
	15
	25
	37
	34
	23

Figure 8. Test case on Query 2 on UWM dataset

4.2. Query results on mondial dataset

Figure 9 shows the first query, query 1, which list the countries latitude at 50.3 with their respective religion. Figure 10 depicts query 2, which show the city of more than 10000 population. From the number of returned results, it shows that the ontology constructed via our mapping scheme is correct.

Query 1: Show the countries latitude at 50.3 with their name and the religion in this country if any.

```
select * where {
  select ?country ?latitude ?name where {
    VALUES ?longitude { "50.3"^^xsd:string }
    ?country mdlo:name ?name .
  }
  OPTIONAL { ?country mdlo:hasReligious ?religious } .
}
```

Query Result:

country	latitude	religions
Belgium	50.3	Muslim
Belgium	50.3	Protestant
Belgium	50.3	Orthodox
Belgium	50.3	Catholic

Figure 9. Test case on Query 1 on Mondial dataset

Query 2: Select city population more than 0000.

```
select ?city ?population where {
  ?city mdlo:population ?population .
  FILTER (?population > 10000)
}order by ?population
```

Query Result:

city	population
Dortmund	600918
Duisburg	536106
Bochum	401129
Wuppertal	383776
Bielefeld	324067
Gelsenkirchen	293542
Bonn	293072
Monchengladbach	266073
Munster	264887

Figure 10. Test case on Query 1 on Mondial dataset

5. CONCLUSION

In this paper, we proposed a set of transformation rules to translate XML documents into OWL ontology representation. The generated ontology is found accurately defined the semantics of the XML document through the evaluation of comparison to the manual transformation approach. In future work, we will focus on the generation of OWL for the unsupported constructs of the validation schema.

ACKNOWLEDGEMENTS





This work is supported by the funding of MMU-Tel U Joint Research Grant (MMUE/210062).

REFERENCES





- [1] S.-C. Haw, A. Amin, C.-O. Wong, and S. Subramaniam, "Improving the support for XML dynamic updates using a hybridization labeling scheme (ORD-GAP)," *F1000Research*, vol. 10, pp. 1–14, Sep. 2021, doi: 10.12688/f1000research.69108.1.
- [2] L. Bao, J. Yang, C. Q. Wu, H. Qi, X. Zhang, and S. Cai, "XML2HBase: storing and querying large collections of XML documents using a NoSQL database system," *Journal of Parallel and Distributed Computing*, vol. 161, pp. 83–99, Mar. 2022, doi: 10.1016/j.jpdc.2021.11.003.
- [3] X. Jin, B. W. Wah, X. Cheng, and Y. Wang, "Significance and challenges of big data research," *Big Data Research*, vol. 2, no. 2, pp. 59–64, Jun. 2015, doi: 10.1016/j.bdr.2015.01.006.
- [4] J. Yang *et al.*, "Intelligent bridge management via big data knowledge engineering," *Automation in Construction*, vol. 135, pp. 1–14, Mar. 2022, doi: 10.1016/j.autcon.2021.104118.
- [5] N. Luo, M. Pritoni, and T. Hong, "An overview of data tools for representing and managing building information and performance data," *Renewable and Sustainable Energy Reviews*, vol. 147, pp. 1–14, Sep. 2021, doi: 10.1016/j.rser.2021.111224.
- [6] A. C. Khadir, H. Aliane, and A. Guessoum, "Ontology learning: grand tour and challenges," *Computer Science Review*, vol. 39, pp. 1–14, Feb. 2021, doi: 10.1016/j.cosrev.2020.100339.
- [7] H. Q. Dung, L. T. Q. Le, N. H. H. Tho, T. Q. Truong, and C. H. Nguyen-Dinh, "A novel ontology framework supporting model-based tourism recommender," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 10, no. 4, pp. 1060–1068, Dec. 2021, doi: 10.11591/ijai.v10.i4.pp1060-1068.
- [8] D. Tomaszuk and D. Hyland-Wood, "RDF 1.1: knowledge representation and data integration language for the web," *Symmetry*, vol. 12, no. 1, p. 84, Jan. 2020, doi: 10.3390/sym12010084.
- [9] S. Elnagar, V. Yoon, and M. A. Thomas, "An automatic ontology generation framework with an organizational perspective," in *Proceedings of the Annual Hawaii International Conference on System Sciences*, Jan. 2020, vol. 2020-January, pp. 4860–4869, doi: 10.24251/hicss.2020.597.
- [10] O. Corcho, F. Priyatna, and D. Chaves-Fraga, "Towards a new generation of ontology based data access," *Semantic Web*, vol. 11, no. 1, pp. 153–160, Jan. 2020, doi: 10.3233/SW-190384.
- [11] C. Blankenberg, B. Gebel-Sauer, and P. Schubert, "Using a graph database for the ontology-based information integration of business objects from heterogenous business information systems," *Procedia Computer Science*, vol. 196, pp. 314–323, 2022, doi: 10.1016/j.procs.2021.12.019.
- [12] M. Ali, S. Fathalla, S. Ibrahim, M. Kholief, and Y. Hassan, "Cross-lingual ontology enrichment based on multi-agent architecture," *Procedia Computer Science*, vol. 137, pp. 127–138, 2018, doi: 10.1016/j.procs.2018.09.013.
- [13] M. Hacherouf, S. N. Bahloul, and C. Cruz, "Transforming XML documents to OWL ontologies: a survey," *Journal of Information Science*, vol. 41, no. 2, pp. 242–259, Apr. 2015, doi: 10.1177/0165551514565972.
- [14] M. Klein, "Interpreting XML documents via an RDF schema ontology," in *Proceedings. 13th International Workshop on Database and Expert Systems Applications*, 2002, pp. 889–893, doi: 10.1109/DEXA.2002.1046008.
- [15] H. Bohring and O. Auer, "Mapping XML to OWL ontologies," in *Lecture Notes in Informatics (LNI), Proceedings - Series of the Gesellschaft für Informatik (GI)*, Bonn: Gesellschaft für Informatik, 2005, pp. 147–156.
- [16] M. J. O'Connor and A. Das, "Acquiring OWL ontologies from XML documents," in *Proceedings of the sixth international conference on Knowledge capture - K-CAP '11*, 2011, pp. 17–24, doi: 10.1145/1999676.1999681.
- [17] M. Ferdinand, C. Zircins, and D. Trastour, "Lifting XML schema to OWL," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Berlin: Springer, 2004, pp. 354–358.
- [18] C. Tsinarakis and S. Christodoulakis, "Interoperability of XML schema applications with OWL domain knowledge and semantic web tools," in *On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS*, Berlin, Heidelberg: Springer, 2007, pp. 850–869.
- [19] I. Bedini, C. Matheus, P. F. Patel-Schneider, A. Boran, and B. Nguyen, "Transforming XML schema to OWL using patterns," in *2011 IEEE Fifth International Conference on Semantic Computing*, Sep. 2011, pp. 102–109, doi: 10.1109/ICSC.2011.77.
- [20] D. Lacoste, K. P. Sawant, and S. Roy, "An efficient XML to OWL converter," in *Proceedings of the 4th India Software Engineering Conference on - ISEC '11*, 2011, pp. 145–154, doi: 10.1145/1953355.1953376.
- [21] P. T. T. Thuy, Y.-K. Lee, and S. Lee, "S-Trans: semantic transformation of XML healthcare data into OWL ontology," *Knowledge-Based Systems*, vol. 35, pp. 349–356, Nov. 2012, doi: 10.1016/j.knosys.2012.04.009.
- [22] P. T. T. Thuy, Y. K. Lee, and S. Lee, "A semantic approach for transforming XML data into RDF ontology," *Wireless Personal Communications*, vol. 73, no. 4, pp. 1387–1402, Dec. 2013, doi: 10.1007/s11277-013-1256-z.
- [23] P. Shapkin and L. Shumsky, "A language for transforming the RDF data on the basis of ontologies," in *WEBIST 2015 - 11th International Conference on Web Information Systems and Technologies, Proceedings*, 2015, pp. 504–511, doi: 10.5220/0005479505040511.
- [24] S. S. Singapogu, P. C. G. Costa, and J. M. Pullen, "Automated ontology creation using XML schema elements," *CEUR Workshop Proceedings*, vol. 1523, pp. 10–17, 2015.
- [25] A. Jounaidi and M. Bahaj, "Designing and implementing XML schema inside OWL ontology," in *2017 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, Nov. 2017, pp. 1–7, doi: 10.1109/WINCOM.2017.8238166.
- [26] A. Jounaidi and M. Bahaj, "Converting of an xml schema to an owl ontology using a canonical data model," *Journal of Theoretical and Applied Information Technology*, vol. 96, no. 5, pp. 1422–1435, 2018.
- [27] M. Hacherouf, S. Nait-Bahloul, and C. Cruz, "Transforming XML schemas into OWL ontologies using formal concept analysis," *Software & Systems Modeling*, vol. 18, no. 3, pp. 2093–2110, Jun. 2019, doi: 10.1007/s10270-017-0651-4.
- [28] University of Washington, "XML repository UWM and mondial datasets." <http://aiweb.cs.washington.edu/research/projects/xmltk/xmldata/www/repository.html#dblp> (accessed May 15, 2018).
- [29] S. Bischof, S. Decker, T. Krennwallner, N. Lopes, and A. Polleres, "Mapping between RDF and XML with XSPARQL," *Journal on Data Semantics*, vol. 1, no. 3, pp. 147–185, Sep. 2012, doi: 10.1007/s13740-012-0008-7.

BIOGRAPHIES OF AUTHORS







Su-Cheng Haw     is Professor at Faculty of Computing and Informatics, Multimedia University, where she leads several funded researches on the XML databases. Her research interests include XML databases, query optimization, data modeling, semantic web, and recommender system. She is also the chairperson for Center for Web Engineering (CWE) and chief editor for Journal of Informatics and Web Engineering (JIWE). She can be contacted at email: sucheng@mmu.edu.my.







Lit-Jie Chew     holds a Bachelor of Computer Science (Hons.) major in data science from Multimedia University, Malaysia and is currently pursuing Master of Science (M.Sc.) in Information Technology at Multimedia University, Malaysia. His research areas of interest include recommender system, ontology, and IoT. He can be contacted at chewljie@gmail.com.







Dana Sulistiyo Kusumo     is a senior lecturer in the School of Computing, Telkom University, Indonesia. He received BEng in Informatics from the Telkom University, Indonesia, MEng in Information Technology from Bandung Institute of Technology, Indonesia and a Ph.D. degree in Computer Science and Engineering from the School of Computer Science and Engineering, The University of New South Wales, Australia. His research interests include Software Engineering and Information Architecture. He is the Head of Information Science and Engineering Lab and also a member of the Software Engineering Research Group at the School of Computing, Telkom University, Indonesia. He can be contacted at email: danakusumo@telkomuniversity.ac.id.



Palanichamy Naveen     joined the Faculty of Computing and Informatics, Multimedia University after receiving Ph.D from Curtin University, Malaysia. She received her Bachelor of Engineering (CSE) and Master of Engineering (CSE) from Anna University, India. Her research interest includes smart grid, cloud computing, machine learning, deep learning and recommender system. She is involved in multiple research projects funded by Multimedia University. She can be contacted at email: p.naveen@mmu.edu.my.



Kok-Why Ng     is a senior lecturer in the Faculty of Computing and Informatics (FCI) in Multimedia University (MMU), Malaysia. He did his B.Sc. (Math) in USM, Penang, and his M.Sc (IT) and Ph.D (IT) in MMU, Malaysia. His research interests are in Recommender System, 3D Geometric Modeling and Animation. He is also active in some research projects related to artificial intelligence, deep learning and human blood cells. He can be contacted at email: kwng@mmu.edu.my.