Journal of Computing and Security

July 2017, Volume 4, Number 2 (pp. 81–100)

http://www.jcomsec.org

A Survey on Transformation Methods of XML Documents to Ontologies

Mohammadereza Pourkhani [a,*], Bita Shadgar [b], Alireza Osareh [b],

[a] Academic Center for Education, Culture and Research (ACECR), Khouzestan, Ahvaz, Iran.

[b] College of Engineering, Computer Department, Shahid Chamran University, Khouzestan, Ahvaz, Iran.

ARTICLE INFO.

Article history:
Received: 31 March 2017
Revised: 03 October 2018
Accepted: 22 October 2018
Published Online: 02 December 2018

Keywords:
Survey, Semantic Web, Transformation, XML Documents, Ontologies.

ABSTRACT

Nowadays, the great amount of information is stored in many data resources such as relational databases, XML documents and so on. Semantic Web is aimed to make current Web information semantic. The information of data resources should be accessed semantically, to reach this goal, it should be organized and stored under the ontology. Ontology is a key and important concept in Semantic Web. Generally, ontology is a common understanding of a domain. On the other hand, completely manually ontology construction is a time-consuming, labor intensive and prone to errors process. Therefore, an approach is required to at least semi-automatically extract ontology from data resources. Transformation approach is a convenient one to extract ontologies from various data resources. Hence, Interoperability among extracted ontologies resulted from transformation approach via mediation ontology approaches is possible and false interpretations of information with regard to different conceptualizations of a given domain is solved. In this paper, we definitely explain the transformation approach, gather the most notable methods in this field and classify them under two categories. Subsequently, we surveyed the transformation methods and finally select the best one in each category.

1 Introduction

The aim of the Semantic Web, also known as Web of data [1], is to integrate various data resources and to make information which is available on the web semantic. One important aspect in ensuring the success of the Semantic Web is to access Semantic data which should be generated ontologies to produce these kinds of data. By using ontology, meaning of information which existed in diverse data resources could formally and explicitly be expressed. As a result, content of these resources would be understandable for machine in addition to being understandable for human. Ontology in the Semantic Web expresses terms and their relations in a given domain [2]. The W3C consortium has proposed several formats for displaying data on the Semantic Web, including RDF, RDF Schema and OWL. Among these languages, OWL is standard language for the Web application programs. Generally, the information available on the current Web has been published by using XML as a standard format for data exchange. Therefore, an approach is needed to semantically represent and re-use of this information in the Semantic Web. On the other hand,

* Corresponding author.

Email addresses: m.pourkhani@acecr.ac.ir (M. Pourkhani), bita.shadgar@scu.ac.ir (B. Shadgar), alireza.osareh@scu.ac.ir (A. Osareh)

https://dx.doi.org/10.22108/jcs.2018.106872.0

ISSN: 2322-4460

unfortunately, completely handy ontology construction is a time-consuming, labor intensive and prone to errors process which needed to cooperate with domain experts. There are two general approaches *i.e.* mapping and transformation to represent XML documents semantically. The main reason which causes the transformation approach more important than the mapping approach is that many data resources have no ontology therefore the transformation approach could extract at least semi-automated local ontologies from these data resources. The transformation approach tries to use the similarities between ontology model and XML model by the definition of rules known as transformation rules to convert XML constructions to OWL ontology ones. The main purpose of this paper is to review the methods that have been proposed to transform XML documents to the ontologies and to determine what progress is made and what works should be done in the transformation process that transforming structures accompanied by transforming meanings. For example, transforming inverse, transitive and symmetric relations should be considered in the future researches.

In this paper, XML technology and ontology were briefly explained in Section 1 and then respectively in Sections 2 and 3, the transformation approach and points associated with it, as well as methods to transfer XML documents to ontologies reviewed in details. Finally in the Section 4, the reviewed methods were compared according to eight criteria and best methods were introduced in the terms of numbers of transformation rules as well as completeness and richness of extracted ontologies.

1.1 XML

In the late 90s, XML language was introduced by the W3C consortium as an easy and standard way to store self-description data. The self-description data are those that define both their own content and structure. This language is separated content from presentation and provided different view of the contents of the documents without need to copy contents [2]. In addition, the content of XML documents can be used in other applications that depend on the definition of appropriate vocabularies by user. Generally, there are two ways, namely data type definition or DTD that is old and limited way and XML Schema that provide greater range of capabilities for defining data types, to specify the structure of XML documents. The DTD constructions could be defined in a separated DTD file (external DTD) or in the same XML document (internal DTD). Usually, it is better to use an external DTD, because definition is applicable in several documents and otherwise repetition is inevitable, also maintain consistency over time is difficult. XML schema provided the most appropriate structure for the definition of XML documents. XML Schema syntax is XML-based. This design features improved readability and more importantly reusability. So that, unlike DTD, writing a separated parser and convenient editor are no longer needed, because XML does this work by itself. Reusable and recoverable patterns are more important features of XML. The ability to define new types by extending or restricting the values is provided in the XML Schema. Thus, constructing model from other model made possible and reduced workload. Also, XML Schema provided complex data types that could be used in other XML documents. XPath is a language for addressing parts of XML documents. This language was implemented on the tree model of XML data and had non-XML syntax. The tree view of a simple XML document is shown in Figure 1.

The main objective of the designing XQuery language was to provide a flexible query tool to extract data from real and virtual Web documents. Also, providing the required interaction between the Web world and the databases world and access to XML documents like database were among other purposes. XQuery language portion to XML is a closely relation like portion of SQL language to relational database. This query language was designed for XML data. XML data is not only XML data files but also anything that is expressed as XML form (like databases). This language is based on Xpath and is supported by most popular databases. XSL language contains both XSLT language and formatting language. However, both of them are a XML application. XSLT language defines the rules by which the XML document (input) converts to another XML document or document becomes an HTML or plain text (the output document may use a DTD or a similar pattern with the input document or use different vocabularies). XSLT language could be used independent from other formatting language. This language is able to transfer data and metadata from a XML document to another one and that's why one of the valuable tools in XML-based applications.

The first advantage of XML language is its expressing language because it uses natural language not computer one. So, this language is easy to read and understand. The second advantage is its expressing language compatible with common programming languages such as Java and C++, so any program that can process XML can also retrieve XML data. The third advantage is the flexibility of XML because it allows all to describe their arbitrary content simply by making relevant tags. This freedom of action could be resulted in lack of mutual understanding between document writer and user. Because of an object or a concept could be expressed with different vocabularies. So, the meaning would be difficult to understand

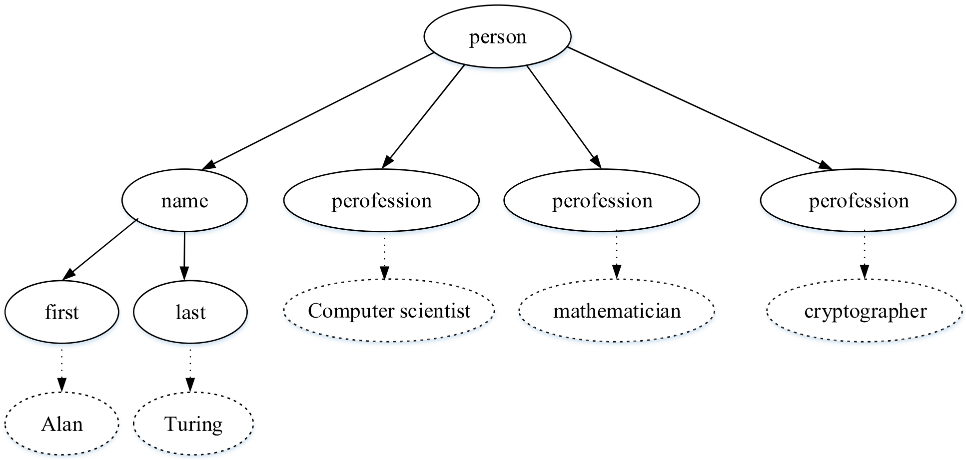

Figure 1. An XML Document Tree

for the computer as document user. For example, for user, generator concept could be expressed by creator or inventor vocabularies. In addition, XML has some semantic disadvantages. In general, XML focuses on grammar and syntax and does not provide any way for describing the meaning of the XML document. This problem occurs when an agent wants to understand and reason about XML data.

1.2 Ontology

Ontology in Semantic Web expresses terms and relations between them in a given domain [2]. In others word, the main components of ontology consist of concepts, properties and the relationships between them. Ontology can reveal relationships between the concepts in any structure like Web documents, so that machines would be able to understand them and then do various processing on these documents. In general view, ontology is as follows [2]:

- Vocabularies + Structure = Taxonomy
- Terms + Rules + Restrictions + Constraints = Ontology
- Ontology + Instances = Knowledge Base

Formally, Ontology is a quadruple $O(C, P^C, R, H)$. Here, O is ontology name with regard to domain of interest, C is a finite set of concepts or classes, P^C is a finite set of concept properties, R represents relation between concepts and H is Hierarchy of classes and properties. Navigli et al. [3] divided ontology into three levels *i.e.* fundamental ontologies (FOs), core ontologies (COs) and specific domain ontologies (SDOs) (Figure 2).

Fundamental ontology or upper Ontology contains very few concepts and basic principles which are existed in all domains. Core ontology contains elements of any given domain. Specific domain ontology consists of ontology elements that should be conceptualize for a given domain. Domain ontology is referred to core ontology with concepts existed in a given domain. Ontology objectives and performance are based upon various ontology application types. The main objective of ontology construction can be outlined as follows:

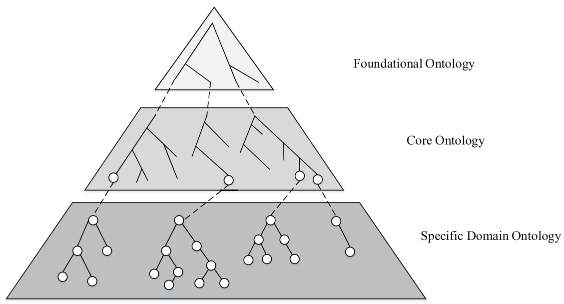

Figure 2. The General Three Levels of the Domain Ontologies [3]

- To agree on the understanding of the shared concepts of information structure among different communities
- To make a domain knowledge analyzable and reusable
- To specify domain assumption in an explicit manner
- To make a partition between operational knowledge and domain knowledge
- To analyze domain knowledge

Those objectives can only be applied on domain ontologies (*i.e.* global ontology), while application ontologies (*i.e.* local ontology) can benefit from these characteristic after mapping them to the global ontologies. Currently, the most important ontological languages for Semantic Web are as follows [2]:

- RDF [1] which is a data model for the resources and the relationships between them. RDF data model creates a simple meaning on data model. This data model is represented by XML syntax.
- DAML [2] + OIL [3] is a semantic markup language. This language is an extension of RDF language trying to alleviate some of its shortcomings through richer primitives. One of the main features of this language is that can formulate meanings of language terms by applying descriptive logic [4] theory.
- RDF Schema or RDFS is the language to describe properties and classes of RDF resources. It contains rules for producing the hierarchy of attributes and classes.
- OWL [5] is a richer language to describe properties and classes rather than RDFS, for example, relation between classes (disjoint classes), cardinality (*e.g.* exactly one), equality, richer types of properties (symmetric or transitive) and enumerative classes are easily expressible in this language. This language takes use of RDF vocabularies and XML syntax, but overcomes those deficits existed in other ontology languages like RDF and DAML+OIL. In the following, Table 1 shows a comparison between the three ontological languages and as well as limitations of RDF and DAML+OIL languages in contrast of OWL language.

As can be seen, OWL ontology language is the most powerful language in terms of expression power among other ontological languages. In logic, usually a profile called a sub-language. Selecting profile or sub-language depends upon ontology structure for program and also, the volume of reasoning operations that must be done on ontology. In general, the OWL 1 ontology language was categorized into three sub-languages *i.e.* OWL Lite, OWL DL and OWL Full where their expression power is increased from left to right respectively and computational cost would be increased proportional to increasing in expression power (Figure 3). In the OWL 1 language, there are two types of properties *i.e.* data type property and object property. Data type property shows relationships between classes and literals while object property shows relationships between instances of two classes.

OWL 2 language has been developed OWL 1 language by a small but useful set of features developed by the user to achieve efficient reasoning algorithms. Generally, OWL 2 ontology language is categorized

[1] Resource Description Framework
[2] DARPA Agent Markup Language (DAML)
[3] Ontology Inference Layer (OIL)
[4] Description Logic (DL)
[5] Web Ontology Language

Table 1. A Comparison Between the Capabilities of Ontological Languages [4]

The Expression	RDF/RDFS	DAML+OIL	OWL
Class	✓	✓	✓
RDF: Property	✓	✓	✓
RDFs: SubClassOf	✓	✓	✓
RDFs: SubPropertyOf	✓	✓	✓
RDFs: Domain	✓	✓	✓
RDFs: Range	×	✓	✓
Individual	×	✓	✓
SameClassAs	×	✓	✓
SamePropertyAs	×	✓	✓
SameIndividualAs	×	✓	✓
DifferentIndividualFrom	×	✓	✓
InverseOf	×	✓	✓
TransitiveProperty	×	✓	✓
SymmetricProperty	×	✓	✓
FunctionalProperty	×	✓	✓
InverseFunctionalProperty	×	✓	✓
AllValuesFrom	×	toClass	✓
SomeValuesFrom	×	hasClass	✓
MinCardinality	✓	✓	✓
MaxCardinality	✓	✓	✓
Cardinality	✓	✓	✓
OneOf	✓	×	✓
DisjointWith	×	✓	✓
ComplementOf	×	✓	✓

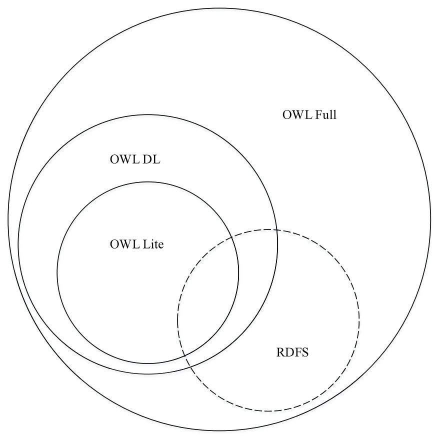

Figure 3. The Hierarchy of the OWL 1 Ontological Language Versions

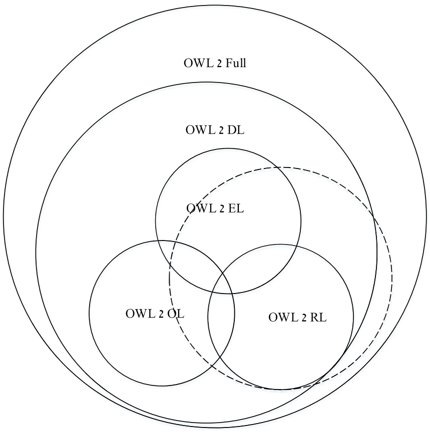

Figure 4. The Hierarchy of the OWL 2 Ontological Language Versions

into five sub-languages *i.e.* OWL 2-EL, OWL 2-QL, OWL 2-RL, OWL 2-DL AND OWL 2-Full [5] (Figure 4). OWL 2-EL sub-language is particularly useful in applications employing ontologies that contain very large numbers of properties and/or classes. It captures the expressive power used by many these ontologies and is a subset OWL 2 DL for which the basic reasoning problems can be performed in time that is polynomial with respect to the size of ontology [6]. OWL 2-QL sub-language is aimed at applications that use very large volumes of instance data and where query answering is the most important reasoning task. The expressive power of the sub-language is necessary quite limited, although it does include most of the main features of conceptual models such as UML class diagrams and ER diagrams [6]. OWL 2-RL sub-language is aimed at applications that require scalable reasoning without sacrificing too much expressive power. It is designed to accommodate both OWL 2 applications that can trade the full expressivity of the language for efficiency.

2 Transformation Process

Generally, ontology construction process consists of three steps *i.e.* namely generation, enrichment and population. In the case that there is no ontology, it should be generated directly by using predefined rules from the XML document. This process is called ontology generation process. This step must be performed automatically. In the ontology enrichment process by using the XML document, new constructions (such as classes, object and data type properties) would be added to existing ontology. This step should be performed at least semi-automatically. Other data resources such as relational database and so on could also be used in this process. Ontology population process is added instances from XML document data to

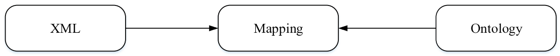

Figure 5. The Mapping of XML to Ontology

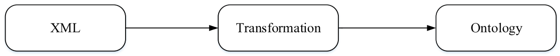

Figure 6. The Transformation of XML to Ontology

generated ontology. This step must be performed automatically. Generation and enrichment processes could be done by using XML document and XML schema or DTD. Ontology population process is usually done by using XML document data and ontology associated to XML document. In the terms of ontology generation and enrichment processes, approaches could be classified into two categories in order to semantic representation of XML document, *i.e.* mapping XML document to ontology and transformation of XML document to ontology.

2.1 Mapping of XML document to ontology

In the mapping approach, most efforts are associated with wrapper systems relating a XML document and its schema to domain ontology. Ontology may be global or local. In this approach, depending on the nature and objectives of each project, building wrapper consists of a range of manual programming to complex mapping languages and environments in which automatic reasoning is used to define mapping systems (Figure 5).

2.2 Transformation of XML Document to Ontology

In the transformation approach, XML document and its structure (DTD or XML schema) are used to convert XML constructions and its data to a semantic representation. In this approach, there is no local ontology and XML constructions are converted to corresponding ones in the ontology by using transformation rules (Figure 6).

Output of transformation process based methods is always ontology, but these methods could be categorized into two groups based upon their input:

- DTD-based methods
- XML schema-based methods

Therefore, in the transformation approach of XML document to OWL ontology, from DTD or XML Schema at least semi-automatically is extracted ontology and non-automatically (*i.e.* manually) ontology generation would be prevented. The transformation approach is more important than mapping approach for two reasons:

Table 2. A Comparison of the Mapping and Transformation Approaches

Method Name	Existence of Source Model	Existence of Goal Model	Complexity of Rules	Complexity of Implementation	Losing of Information and Meaning	Dependency to Domain
Mapping	Yes	Yes	Complex	Simple	No	Yes
Transformation	Yes	No	Simple	Complex	Yes	No

First, Source model *i.e.* XML documents or relational databases may be available but its destination model *i.e.* local ontologies may not be available. Second, developing a new ontology model for a domain is a lengthy process that involves identifying concepts, properties, relationships, constraints and restriction in a specific domain and validating the resulted model with domain experts. The ontology engineer needs expertise in the domain, knowledge in ontology modeling and skills necessary in using an ontology language or ontology editing tool [7]. Moreover, depending on the domain, developing an ontology model can take several months to several years and may require continuous involvement of domain experts. Since organizations tend to avoid engaging domain experts in long running endeavors and since some of this knowledge has already been encoded in computer applications and databases, several researchers have decided that it might be more practical to derive this knowledge from existing data models. This derived knowledge could then be used to generate a rudimentary ontological model which can then be revised or further augmented [7]. A comparison of the mapping and transformation approaches is shown in Table 2.

Some notes about the transformation process should be considered:

- The transformation rules should not be too complicated.
- The required time for the transformation process should be considered.
- User involvement level during the transformation process should not be exceeded from a certain threshold.
- Information and meaning losses should not be occurred as a result of the transformation process.
- The transformation rules should be able to convert almost all constructions of source model into destination model as much as possible.
- The transformation rules should not be dependent on certain domain that is, local ontology could be extracted from data resources with uniform format by using same transformation rules.

All the XML to ontology transformation methods should be satisfy the following requirements [8]:

- Data integrity: The output of the transformation process should describe the original data as enough as possible.
- Semantic integrity: In some circumstances, the transformation process output *i.e.* ontology should describe accurately the information contained in the XML document, so the quality of the transformation process should be analyzed.
- Preserving the original structures: Along with mapping XML structures such as elements and attributes, transformation procedures should also support relationships between these constructions.
- Mapping data types: Each attribute should be mapped along with its data type.
- Using common syntax: output ontology should be expressed by using common syntax such as OWL ontology language otherwise it could not interact with other Semantic Web applications and ontologies in a given domain. In this case, the process suffers from the applicability problem.
- Automation of transformation process: The transformation process should be performed at least semi-automatically and fully manual operation should be avoided.

Generally, some notes should be considered to achieve an efficient and complete method for constructing ontology from XML document [9]:

First, the method should be based on DTD or XML Schema instead of the XML document, because XML schema or DTD could be used for several XML document. Therefore, this prevents generating several ontologies from several XML documents based on a DTD or XML Schema. Second, the method should present mapping of ontology constructions and corresponding XML ones in a file. This file could be used to retrieve and translate answers related to user queries. Third, the method should be relied on declarations existed in DTD or XML Schema instead of declarations of elements existing in the XML document to benefit from reusability of types by several elements within the schema. Relying on elements declarations leads to generating redundant OWL terms from multiple elements of same type. Fourth, XML document could be modeled using different styles. Some of them only have a global element as the root element, while others have multiple global elements. Some of XML

Table 3. The Mapping of the DTD Constraints to the OWL Constraints

DTD Constraints	OWL Constraints
#FIXED value <!ENTITY entity-name “entity-value”>	OWL:HasValue
#REQUIRED	OWL:MinCardinality (=1)
#IMPLIED	OWL:Cardinality (=0)
NOTATION	RDFs:Comment
+	OWL:MinCardinality (=1)
?	OWL:MinCardinality (=0)
*	OWL:MinCardinality (=0) OWL:MaxCardinality (=unbounded)

documents used types defined locally and others used types defined globally. As a result, the method should be able to extract ontology from any XML documents with any designs. Fifth, the method should have a step to evaluate and refine structure of built ontology and mapping file. This step could be done manually by domain experts.

3 Transformation Methods

In this section, transformation methods are discussed to construct OWL ontology form the DTD and then the XML Schema. It is necessary to note that the methods are described chronologically.

3.1 Transformation From DTD to OWL

Pam et al. [8] presented an automatic method to extract OWL ontology from DTD. In this method, first all of the DTD constructions would be converted into OWL ones by using transformation rules. Also, in the case that DTD of XML document is not available, DTD would be created from related XML document making use of HiTSoftware[6] tool. Then, ontology would be populated by using XML document data together with made OWL ontology. In this step, population procedure traverses from open-tag of root element and ends when it meets close-tag of root element. Meanwhile, if an element in XML document is matched with a node in OWL ontology, then an ontology instance corresponding to a XML individual would be generated. The authors mentioned that one of drawbacks of other methods is during the ontology population phase. When in the transformation process of XML document data, the next element has the same name with the previous one, no solution to deal with this problem exist. In this method, element renaming procedure would be used. The proposed method framework is shown in Figure 7.

[6] `http://www.hitsw.com/xml_utilites`

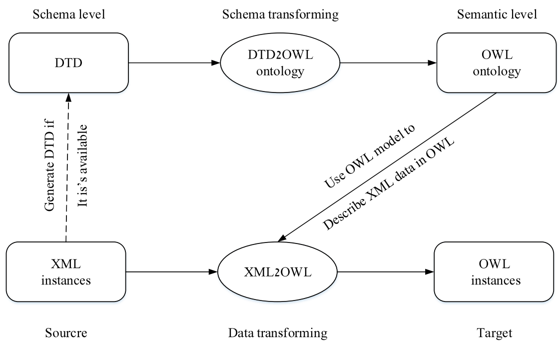

Figure 7. The General Framework of the DTD2OWL System [8]

Element defined by <!DOCTYPE> in DTD is mapped to the root class of OWL ontology which is the first class declared by owl:class. For element definition <!ELEMENT element-name (element-content)>, if the element-content contains sequences of children, this element considered as OWL class. Each owl:class is represented by a unique identifier, rdf:ID and disjoints with other classes. For the case, an element in DTD is described by <!ELEMENT> tag but it only contains data type (#PCDATA or #CDATA), it is mapped to an OWL data type property. The mappings of DTD constraints are listed in Table 3.

For nested elements, the rdfs:subclassof declaration was not used. The reason is because some nested elements in a DTD are not actually the sub-class of their parent element. This method is automatic, as result a middle course was considered. A new object property described by owl:objectproperty to establish the relationship between child node and parent node. The name of object property is derived for ”has” concatenating with child node name. It should be noted that this method is a simple and incomplete for transformation of DTD to OWL ontology which does not consider all DTD constructions in transformation process. For example, attribute types *i.e.* enumeration, ID, IDREF; different entity types *i.e.* general, parametric, external and internal; the hierarchical relationship between the declarations of elements *i.e.* sequences,

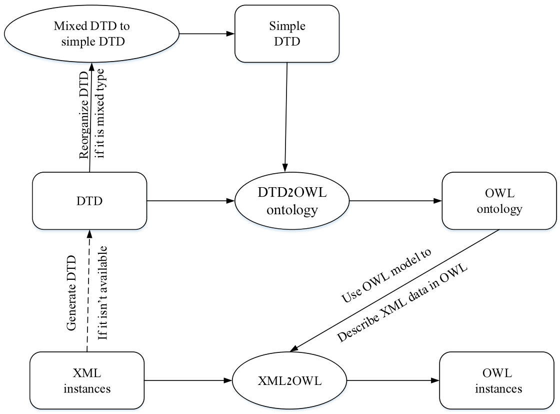

Figure 8. The General Framework of the Extended DTD2OWL System [12]

choice and mixed DTD. These constructions are the most commonly used in the DTDs. Also, the Authors in their two previous works [10, 11] transferred DTD and XML schema to RDF schema. In these two works, extracting classes and sub-classes form related XML document is based on transformation rules and also user opinion (semi-automatic). But in this method, sub-classes were considered as object property for expressing relationship between parent and child classes and as result transformation process is automatic and user opinion is no longer needed. Another disadvantage of these two works is to use RDF Schema language, because the language is less expressive power than OWL language.

Hacherouf et al. [12] developed the DTD2OWL system which is the best method for automatically transformation of XML document into OWL ontology in this category. In this method, like the DTD2OWL system, OWL ontology is generated from DTD file related to XML document. If DTD file is not available, first it is created from XML document and then, ontology would be populated by using XML document data together with made OWL ontology. The authors described difference of their method with method of Pam et al. [8] that, in proposed method, first it is checked that if DTD file is simple, then ontology would be generated by using transformation rules. But if DTD file is mixed one, DTD file and other related DTD files which are referenced each other would be merged and a simple DTD file would be formed and then OWL ontology would be generated. Also, the merging algorithm consisted of three phases. The proposed method framework is shown in Figure 8.

It could be said that such an idea to solve the problem of references in the construction of the ontology of more than one XML Schema belonged to Lacoste et al.'s method [13]. In fact, it could be said that this method is a development of Pam et al.'s method [8] and as well as the main idea of Lacoste et al.'s method [13]. In this method, more transformation rules than Pam et al.'s method [8] were provided that converted more DTD constructions to OWL ones, as a result ontology generated by the extended DTD2OWL system is more complete than the primary DTD2OWL system. The transformation rules with examples are listed in Table 4.

3.2 Transformation from XML Schema to OWL

Klein [14] presented a method for automatically transforming XML document data to RDF instances. In this method, the ontology that describes the XML document is already existed in RDF Schema therefore the aim of extracting RDF triples is to populate the ontology. All elements of the XML document tree are traversed in a depth-first order, starting from the root element. Only elements of the XML document are translated into RDF triples that are classes or properties in the ontology, so all meanings of the XML document are not extracted. For example, name and value of tag properties are not converted to RDF triples. An ontology which is in the RDFS language cannot express all meanings of a given domain, which regarding the publication date of the work, common languages for the expression of the ontology were RDFS and DAML + OIL.

Ferdinand et al. [15] presented a method for automatic transformation of XML schema into OWL ontology. In this method, XML document data are mapped to the RDF graph. The authors cited that the XML mapping into RDF can do efficient inference and semantic validation of XML document data. Once translated into RDF, the data can be classified with an OWL reasoner. As a result, the classification could lead to discover implicit class membership or implicit relationships between objects. For naming all new constructions in the ontology, the compound of namespace with the # character and attribute names used in the XML Schema was used. The transformation rules of XML schema to OWL ontology are listed in Table 5. Also, the transformation rules of XML schema to RDF are briefly listed in Table 5 and Table 6 respectively.

Cruz et al. [16] proposed a method for using an ontology-based mediation framework to integrate heterogeneous XML sources. The framework comprises of two phases; first, transforming XML schema to local ontologies and second merging the extracted ontologies into a global ontology. During the process of merging, a mapping file is also created to maintain information between general and local ontologies. The transformation rules are divided into two categories:

Table 4. The Transformation Rules of the Extended DTD2OWL System

DTD Constraints	**OWL Constraints**
Complex Type	Class
Simple Type/Attribute	DataType Property
#FIXED value <!ENTITY entity-name "entity-value">	OWL:HasValue
#REQUIRED	OWL:MinCardinality (=1)
#IMPLIED	OWL:Cardinality (=0)
NOTATION	RDFs:Comment
+	OWL:MinCardinality (=1)
?	OWL:MinCardinality (=0)
*	OWL:MinCardinality (=0) OWL:MaxCardinality (=unbounded)
Enumeration Type	OWL:OnOf
Attribute ID Type	ObjectProperty
Attribute IDREF Type	ObjectProperty
SEQUENCE	Owl:IntersectionOf
Choice	Owl:UnionOf

Table 5. The Transformation Rules of the OWLMAP System (XML to OWL)

XML Schema Constructions	**OWL Constructions**
Complex Type/ Element Type/ Attribute Group	OWL:Class
Simple Type/ Attribute	OWL:DataTypeProperty
Element Of Complex Type	OWL:ObjectProperty
Inheritance Restriction/Inheritance Extension	RDFs:SubClassOf
Substitution Group	OWL:SubPropertyOf
Element Type Local/ Attribute Type Local	OWL:AllValuefrom
Min Occurs/ Max Occurs	OWL:MinCardinality/ MaxCardinality/ Cardinality
Sequence/ All	OWL:IntersectionOf
Choice	Boolean Expression With OWL:IntersectionOf, OWL:UnionOf, OWL:ComplementOf

Table 6. The Transformation Rules of the OWLMAP System (XML to RDF)

XML Schema Constructions	**RDF Constructions**
Composed Element	Resource
Sub-Element/ Attribute	RDF Property
Simple Element	RDF Literal

the element level and the structure level (Table 7).

To achieve the reversible query translation process between the structural query and the semantic query in the framework, the authors extend the RDF vocabulary to represent not only the semantics but also the structure of the data. The user queries should be applied onto the global ontology. A Query is rewritten into sub-queries and applied on each XML documents

Table 7. The Transformation Rules of the Cruz et al. [16]

XML Schema Constructions	**RDF Schema Constructions**
Element Level Transformation Rules	
Attribute	Property
Simple Type Element	Property
Complex Type Element	Class
Structure Level Transformation Rules	
Element Attribute Relationship	Class to Literal Relationship
Element Sub-Element Relationship	Class to Class Relationship

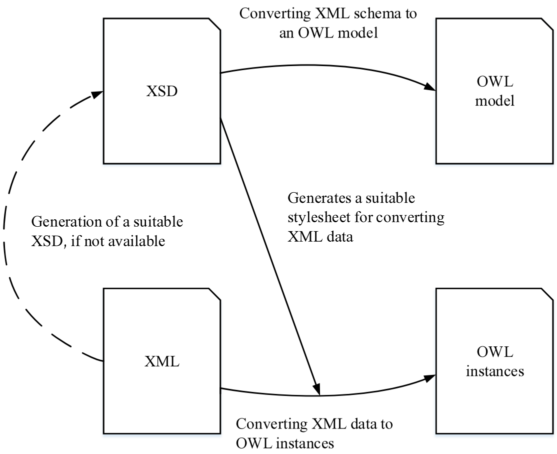

Figure 9. The general framework of the XML2OWL system [17]

then the results are aggregated and returned. The use of ontology in RDFS language because of the less expressive power than the OWL ontology language and therefore the generation of primitive ontology is the main point of weakness of the method.

Bohering et al. [17] presented a method to transfer XML data model into OWL ontological model. In the case that XML schema is available, it is used to generate ontology otherwise, XML schema is generated from related XML document. Also in this method, the used XML documents were made from relational databases. XML document and XSL style sheet were used to generate ontology instances. Halpern 's algorithm [18] was used to create XSL style sheet from XML document. A XSL Style Sheet file which converts the XML individuals into the OWL ontology instances is created simultaneously. This style sheet will be configured automatically to adjust the transformation process of the instances to the OWL model. To support the separation of model and data, the OWL model will be stored separately from the OWL instances. The OWL instances will connected to their model using owl:import property, therefore each OWL instance which references the OWL model obtains an adjustable namespace prefix. The general framework of the proposed method is shown in Figure 9.

The authors noted that user does not have any control over the transformation process. The transformation algorithm is based upon heuristic method, so that possibly no optimal solution will be reached. As a result, the generated ontology will be a simple and elementary one that required handy work to refine and enrich and the instances generation is almost primitive because duplicate instances are created with distinct IDs. The transformation rules are listed in Table 8.

Tsinaraki et al. [19, 20] presented an algorithm to automatically transformation of XML schema into OWL ontology (Figure 10) as well as an algorithm to automatically transformation of OWL ontology into XML schema (Figure 11).

In this method, OWL ontology was made only from XML Schema and ontology instances were not made. Also like Bohering et al.'s method [17], a XSL style sheet was used for implementation. The transformation rules are almost similar to Bohering et al.'s method [17] and these rules are listed in Table 9.

Table 8. The Transformation Rules of the XML2OWL System

XML Schema Constructions	OWL Constructions
Elements, containing other elements or having at least one attribute	Owl:Class, coupled with Owl:Object Properties
Elements, with neither sub-elements nor attributes	Owl:Datatype Properties
Named complex Type	Owl:Class
Named Simple Type	Owl:Datatype Properties
MinOccurs, MaxOccurs	Owl:MinCardinality, Owl:MaxCardinality
Sequence, All	Owl:intersectionOf
Choice	Combination of Owl:intersectionOf, Owl:unionOf and Owl:complementOf

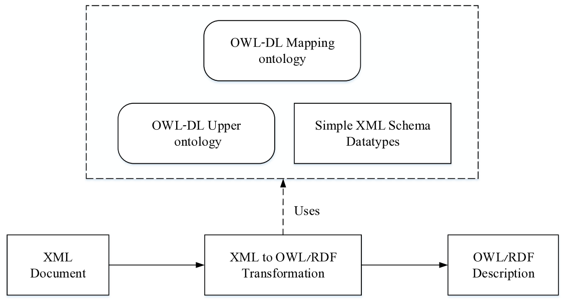

Figure 10. The Transformation of the XML Document Into OWL Ontology [19, 20]

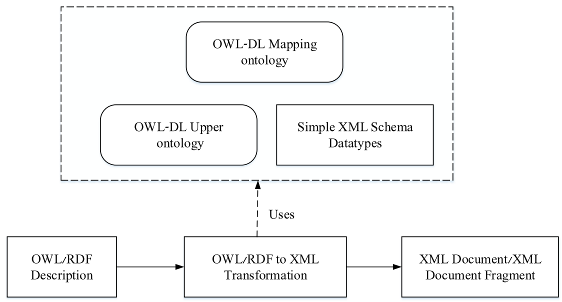

Figure 11. The Transformation of the OWL Ontology Into XML Document [19, 20]

Table 9. The Transformation Rules of the XML2OWL System

XML Schema Constructions	OWL Constructions
Complex Type	Class
Simple Type	Datatype Property
Element	Datatype or Object Property
Attribute	Datatype Property
Sequence	Unnamed Class-Intersection
Choice	Unnamed Class-Union
Annotation	Comment

In this method, mapping of OWL ontology constructions corresponding to XML Schema ones would be stored in a file called mapping file. Because, this file is used to restore OWL ontology to XML schema. This method is composed of three parts. A main OWL ontology or upper ontology that directly captures XML schema semantics. A data types XML schema which contains the simple XML schema data types defined in the source XML schema and are used in the main ontology. A mapping OWL ontology that keeps OWL ontology constructions corresponding to XML schema along with a unique identifier. In a valid OWL ontology, the different constructions should have unique identifier while the XML schema allows different constructions to have the same name. Also, the mapping OWL ontology systematically captures the semantics of the XML schema constructions that cannot be directly captured in the main ontology, since they cannot be represented by corresponding OWL constructions. Other classes in the mapping ontology are as follows [19, 20]:

- The DataTypePropertyInfoType class which maintains the mapping of the OWL data type properties with the corresponding XML schema constructions and specifies the possible default value and type of the XML schema constructions represented by a data type property.
- The ElementInfoType class which captures information about the XML schema elements that cannot be directly represented by OWL constructions and maintains the mapping of the XML schema elements with the corresponding OWL constructions.
- The ComplexTypeInfoType class which captures information about the complex XML schema types that cannot be directly represented by OWL constructions and maintains the mapping of the complex XML schema types with the corresponding OWL constructions.
- The ChoiceType and SequenceType classes which captures information about the complex XML schema sequences and choices that cannot be directly represented by OWL constructions.

Xu et al. [21] presented a method to generate OWL ontology from XML document by using relational model. In this method, the relational model is used as an intermediate model in the process of ontology generation. They used two algorithms referred to as XTR [7] to map XML document to relational database and RTO [8] to map relational database OWL ontology. According to XTR algorithm to create relational database from XML document, the XML document is considered in the form of a tree with labeled nodes. In this tree, if the element is leaf it includes a literal value and has not attributes, such as an element is regarded as simple type. Each simple type is regarded as an attributes and it is mapped to table columns. Also, if an element has sub-element or attribute, such an element is regarded as complex type. Each complex type would be mapped to a class and its related sub-elements and attributes are regarded as properties. Each class would be mapped to a table and its properties are regarded as table columns. The table name and table column name are respectively equaled to class name and class property name. It should be noted that the primary keys, foreign keys and other constraints of tables are defined manually. The authors mentioned that adding primary keys, foreign keys and other constraints of tables in a completely handy form, is not a good and correct idea. Also, no specific algorithm is cited for the RTO section, but the terms and constraints were defined manually for ontology. Obviously, the methods without intermediate model which directly generating ontology are faster and have less overhead and avoid complexities of converting models into each other and losing information and semantics during these conversions.

Rodrigues et al. [22] presented a semi-automatic tool called JXML2OWL to transform or map XML documents to OWL ontologies. The purpose of the tool is to develop a user-friendly and interactive tool that allows a user to transfer data in XML format to OWL ontology or to map to any existing ontology defined in OWL language with the purpose of easing and automating the semantic data integration process. The authors used some rules of Ferdinand et al. [15] and Bohering et al. [17], therefore the input of proposed tool can be either XML schema or DTD. Also, if an XML schema is not available, it is possible to load an XML document. So, the tool extracts a possible schema. XSLT is the used standard to transform XML documents. The XML document generated by the tool can be used by any XSL processor to automatically transform XML data into instances of the ontology. In

[7] XML to Relational Database (XTR)
[8] Relational Database to Ontology (RTO)

Table 10. The Transformation Rules of the JXML2OWL System With Related Notations

XML Constructions	OWL Constructions	Notations
node A	Class	(OWL Class URI, expression XPath) (OWL Class URI, expression XPath, expression XPath ID)
A node	Data type Property	(OWL Data type Property URI, Domain Class Mapping, Expression XPath)
A relation between two node	Object Property	(OWL Object Property URI, Domain Class Mapping, Range Class Mapping)

this method, Xpath expression is used to distinguish multiple nodes with same name but with different tags. Also, if there is an existing ontology for the given domain which is richer than extracted ontology from XML document, the XML data is mapped to instances of existing ontology. The user has no control during transformation process but in the mapping process, the user has full control over process. The transformation rules of the method with notations are as below (Table 10).

Ghawi et al. [9] presented a tool called X2OWL for automatic extraction of OWL ontology from XML document. In this method, made ontology only includes concepts and properties but not instances. Individuals would be retrieved from XML document when answering user queries and then translated into ontology instances. The reason for this is that the proposed tool is part of the framework to integrate data resources such as relational databases by using ontologies. In this method, XML schema was used to generate ontology and if XML schema related to XML document is not available, XML schema is generated by using the Trang[9] software. In the following, XML Schema would be evaluated and then its graph would be obtained by using the Xsom[10] and Jung[11] software. In the next step, OWL ontology would be extracted by using Jena Framework[12] and transformation rules. Finally, generated OWL ontology as well as mapping file would be evaluated and refined. The authors noted that to insure independence of the schema design style, algorithms is based on an XML schema graph that describes the schema in the same way whatever its design style is. The Transformation rules and the general framework of the proposed method are shown in Table 11 and Figure 12 respectively.

The aim of the refinement step is to detect and remove invalid mappings. Invalid mappings should be deleted because if they will be used in query resolution, they will lead to invalid queries that return no results. A mapping is invalid, if it contains an invalid

Table 11. The Transformation Rules of the X2OWL System

XML Schema Constructions	OWL Constructions
Global Named Complex Type	Class
Local Anonymous Complex Type	Class
Inheritance: Extension/ Restriction	Class Inheritance: Rdfs:subClassOf
Element Group	Class
Attribute Group	Class
Element-Sub Element Relationship	Object Property
Simple Type	Datatype Property
Attribute	Datatype Property

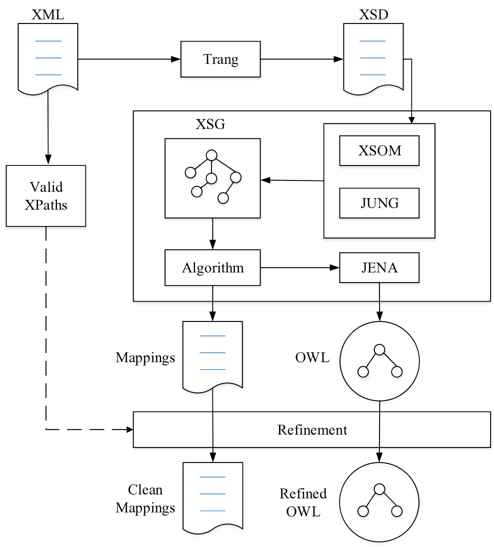

Figure 12. The General Framework of the X2OWL System [9]

Xpath expression for a given XML document or it references another invalid mapping. Detecting invalid mappings can be automatically, if an XML document is provided which is considered typical of all XML documents conforming to the used XML schema. In this case, all possible Xpath expressions of the mappings are compared with those extracted from the typical XML document any mapping that contains an Xpath expression non-belonging to the typical XML docu-

[9] http://www.thaiopensource.com/relaxng/trang.html
[10] https://xsom.dev.java.net
[11] http://jung.sourceforge.net
[12] http://jena.sourceforge.net

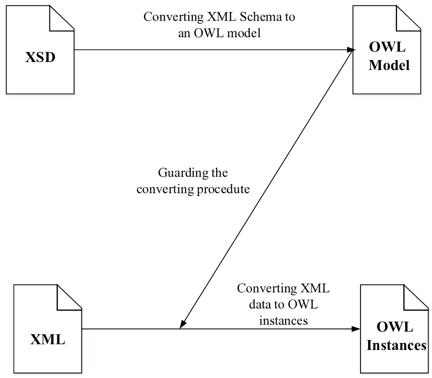

Figure 13. The General Framework of the Jeiping et al. [24]

ment is considered invalid. Furthermore, the mappings are rescanned to detect mappings that reference invalid mappings. Those mapping are also considered invalid. If no typical XML document is provided, the process can be done by a human expert manually.

Breitling et al. [23] presented an automatic tool to translate XML data into RDF via XSLT. This method has three features. First, empty nodes in the RDF are avoided, therefore for each node, a unique identifier URI is considered. The reason is that such nodes are inaccessible in RDF database and decrease the performance of queries in SPARQL language. Second, the suggested transforming method is a bidirectional, so that XML schema transformed into RDF could be reversed to primitive schema but inverse transformation is not yet implemented. Third, independence of XML schema *i.e.* Elements in some construction of XML documents have order such as Sequence, this order may be ignored in the reversed transformation process. So in the tool, an option to keep order of elements for such constructs is considered. The transformation rules are written in complicated XSLT language which requires the users to have sufficient knowledge of this language. If an Xpath expression is not suitable as URI for one node, then user manually inserted a unique URI.

Jeiping et al. [24] presented an automatic method to generate OWL ontology from XML schema. The authors based their proposed method on the Tsinaraki et al.'s method [19, 20]. The general framework of proposed method is shown in Figure 13.

After extraction of ontology, it would be populated by using the XML document data. Also, the transformation rules of proposed method are listed in Table 12.

Generally, XML schema does not provide Information about the structure and elements of a valid XML document. This information could be used to define Xpath expressions for selecting elements and attributes of the XML document. When the elements

Table 12. The Transformation Rules of the Jeiping et al. [24]

XML Schema Constructions	OWL Constructions
XML Schema DataTypes	XML Schema DataTypes
Attributes	DataType Properties
Simple Type Elements	DataType Properties
Complex Type Elements	Object Properties
Complex Types	Classes
Sequence	Unnamed Class-Intersection Of
Choice	Unnamed Class-Union Of
Attributes, Attribute Groups, Element Groups are referenced in Complex Type definitions	If they are contain Attributes or Simple type Elements DataType Properties
	If they are contain Complexes type Elements Object Properties

or attributes of the XML document is selected, their content is mapped to corresponding classes and properties in the OWL ontology. In some cases, additional information is required to select an element or attribute by the Xpath expression. This occurs when some elements are in a same level in an XML document. Therefore, the variables used in the Xpath expressions, and then at run time, the real values would be replaced variables.

Bedini et al. [25] presented an automatic method to extract OWL ontology from XML schema by using pattern recognition method. The generated ontology is in the Turtle format. This method is used 40 patterns that by using it, almost all the constructions of XML Schema transferred to the related constructions of OWL ontology. Also, no pattern is mentioned for generating ontology instances. The authors cited that availability of proposed method as one of their privileges to other methods in this field. They noted that since the OWL ontology is generally more expressive than XML schema, it is not possible to define a direct transformation pattern for each OWL ontological constructions. As an example, it is not possible to define automatically a pattern from XML schema for describing binary relations such as inverse, transitive and symmetric properties. The same is true for other constructions like owl:differentFrom, owl:NegativepropertyAssertion and owl:PropertyChainAxion. However, neither is it possible to convert all XML schema integrity constraints onto OWL ontology, as there are areas where XML schema is more expressive than OWL ontology, For example pattern and length constraints on data values *i.e.* <pattern value="[a-z] [a-z][0-

9]”/>, <xs:length value=”8”/>, <xs:minlength value=”5”/>, <xs:maxlength value=”5”/>, have no direct mapping into OWL ontology. Also in this method, the WordNet dictionary is used for normalizing tags which extracted from XML Schema to produce semantically well-structured ontology.

Lacoste et al. [13] presented a fully automated method to generate OWL ontology from XML Schema. The name of purposed method is EXCO *i.e.* an Efficient XML to OWL Converter. In this method, the OWL ontology would be generated by using one or more XML Schema files that are associated with each other. Ontology extraction phase and ontology population phase could be performed in parallel. Before the ontology extraction phase, there is a phase called schema consolidation. In this phase, Internal and external references problem would be solved. Internal references are those references referred to elements or data types in the same file but external references are those references referred to elements or data types belonged to another XML schema. The external references are made using either the ”include” or ”import” mechanisms. The ”include” mechanism of XML schema is similar to the ”import” mechanism, the only difference being that in the case of ”include” the namespace of the calling document is preserved, whereas in ”import” the child element has its own namespace. The schema consolidation phase is composed of three parts. In the first step, all of the files that are included and imported by the main XML schema are collected. Hash tables are used for this purpose. In this table, such information as file name, file location, namespace, namespace pre-fix and so on would be stored. Also, in this step main schema could be specified. For example, main schema could be the schema having the most number of references. The first step also resolved potential issues such as resolving circular and multiple references. For example, it is possible for a schema to be referenced many times and to import schemas from which it was imported, which could lead to an infinite loop. The second step merges the collected schemas into one schema document. The second step also handles the namespaces. In the third step, the merged schema document is reorganized to remove all internal references. The transformation rules and ontology population phase of this method are similar to that of Bohering et al.'s method [17]. The ontology instances are generated from related XML documents, XML schema and also by help of a XSL style sheet file. The transformation rules of proposed method are listed in Table 13 and also, the general framework of proposed method is shown in Figure 14.

Yahia et al. [26] presented a method to automatically generate ontology from XML schema. The proposed method is approximately similar to Ghawi et

Table 13. The Transformation Rules of the Lacosti et al. [13] .

XML Schema Constructions	**OWL Constructions**
Elements, Containing other elements or having at least on attribute	Owl:Class, Coupled with Owl:Object Properties
Elements, with neither sub-elements nor attributes	Owl:DataType Properties
Named Complex Type	Owl:Class
Named Simple Type	Owl:DataType Properties
MinOccurs, MaxOccurs	Owl:MinCardinality, Owl:MaxCardinality
Sequence, All	Owl:IntersectionOf

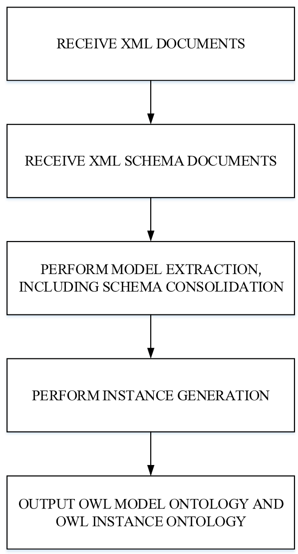

Figure 14. The General Framework of the Lacosti et al. [13].

al.'s method [9] with this difference that this method has no phase to evaluate and refine generated ontology and revise mapping file through valid XPath expression. The general framework of proposed method is shown in Figure 15.

Shenoy et al. [27] presented a method to integrate data resources that are structurally and semantically heterogeneous. The suggested data resource used for the integration system is XML documents that have different schemes. In this method, first two different local ontologies are obtained by using transformation rules and then a global shared ontology is obtained making use of mapping rules from two local ontologies. Also, XML document data are not to be converted into ontology instances but user queries in SPARQL format first are applied on the global shared ontology and then would be applied by each of two XML document by using mapping rules written in XQuery format. Finally, answers after merging and translating would

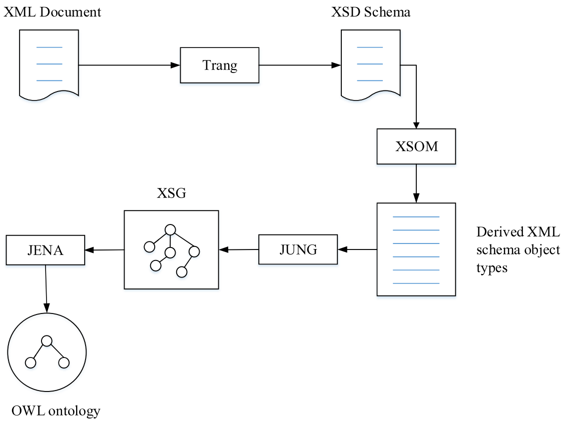

Figure 15. The General Framework of the Yahia et al. [26].

Table 14. The Transformation Rules of the Shenoy et al. [27].

XML Schema Constructions	OWL Constructions
Owl:Class	Complex type
Owl:DataType Properties	Simple type
Owl:DataType Properties	Attribute

be return to user. The transformation rules of XML schema to OWL ontology are so simple and are as follow; each complex type is mapped to a concept, each simple type is mapped to a property and each attribute is mapped to a property, too. The authors defined no rule to create object property in the target ontology and only considered to create data type property. The transformation rules of proposed method are briefly listed in Table 14.

Tan et al. [28] presented a method to interact simulation systems in WISE1 platform. The simulation systems use XML format to exchange information. XML format is a common and standard representation to exchange information but in a domain for two XML documents, this may lead to false interpretation of information duo to different conceptualization. That is why it is better to exchange information among such systems in the platform via OWL ontology. It should also be noted that the architecture of the method is similar to methods of Ghawi et al. [9] and Yahia et al. [26]. The framework and transformation rules of the method mentioned as below (Table 15) (Figure 16):

In the method, a set of mappings are defined to generate stylesheet for translating a schema to equivalent ontology. This stylesheet is used to translate XML documents that have the same schema. It is obvious that for different schema, different stylesheet is used. Like Ghawi et al.'s method, there is a configuration file to maintain all valid Xpath expressions for any kind of OWL ontology. If an Xpath expression is detected as valid, by using predefined mappings, the related

Table 15. The Transformation Rules of the Tan et al. [28] .

XML Schema Constructions	OWL FULL Constructions
Named XSD:Elements (not within the elements "datatypes")	Class
<...XSD:Element name="Attribute" type="X">	ObjectProperty, DatatypeProperty or AnnotationProperty (Detemined by the value of the type)
Named <XSD: Attribute type="X" ...>	ObjectProperty, DatatypeProperty or AnnotationProperty (Detemined by the value of the type)
Named XSD: ComplexType on top level	RDFS:DataType
Named XS: Element within the Element "DataType"	RDFS:DataType
XSD:MinOcurrs, XSD:MaxOcurrs	OWL: MinCardinality OWL: MaxCardinality

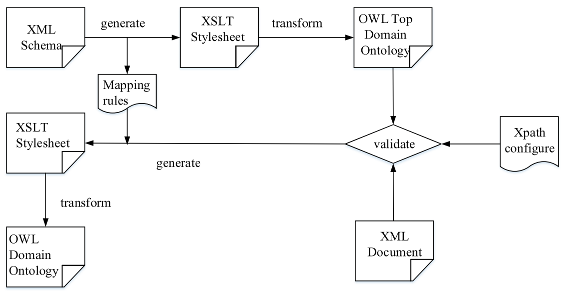

Figure 16. The General Framework of the Tan et al. [28].

stylesheet is created and then the OWL ontology is generated.

El hajjamy et al. [29] presented an automated method with 23 rules for transformation of XML schema to OWL2 ontology. The proposed method consists of three sections; XML schema is analyzed by using DOM1 parser and schema constructions are extracted. Then, transformation process is done with the use of defined rules. The framework and rules of the method are as follows (Table 16) (Figure 17):

4 Discussion and Conclusion

Almost all existing methods to transfer XML documents into ontologies reviewed in this paper. It has been tried to mention the framework and the rules of each reviewed method, so that the similarities and differences of them can be easily compared. Generally, methods according to their input were divided into two categories, namely methods based on XML Schema and DTD-based methods. In general, there is

Table 16. The Transformation Rules of the El hajjamy et al. [29] .

XML Schema Constructions	OWL2 Constructions
Global Named Complex Type	Class
Element/Attribute Group	Class
Local Anonymous Complex Type	DataType Property
SimpleType/Attribue	DataType Property
Restriction on XML Schema DataType Content using xsd:pattern	Restriction on DataType Property using xsd:pattern
Restriction on XML Schema DataType Value using xsd:min(max) Inclusive	Restriction on DataType Property using xsd:min(max) Inclusive
Restriction on XML Schema DataType Value using xsd:min(max) Exlusive	Restriction on DataType Property using xsd:min(max) Exlusive
Restriction on XML Schema DataType Length using xsd:min(max) Length	Restriction on DataType Property using xsd:min(max) Exlusive
Restriction on Set of Values (Enumeration)	DataOneOf
Union Elements	A New DataType using DataUnionOf
Transitive Relation	TransitiveObjectProperty
Xsd:Key	HasKey
Xsd:KeyRef	FunctionalObjectProperty
Binary Relation	Two Pairs of ObjectProperty
N-ary Relation	Chaing Axiom Propert and Self Restriction ObjectHas-Self
Element with a unique Constraint	Set DataMaxCardinality to 1
use="required"	Set DataMinCardinality to 1
use="optional"	Set DataMinCardinality to 0
minOccurs	DataMinCardinality
maxOccurs	DataMaxCardinality
minOccurs	ObjectMinCardinality
maxOccurs	ObjectMaxCardinality

no efficient and completely automatic solution that can transform all the constructs of XML model to the equivalent constructs of ontology by preserving semantic. The comparisons of the reviewed methods are difficult to specify the right transformation process. Each method overcomes the drawbacks of the others and each tried to solve a problem in this field. The generated ontology in the methods using RDF language to describe the ontology has less constructs and meanings than those describe by OWL language,

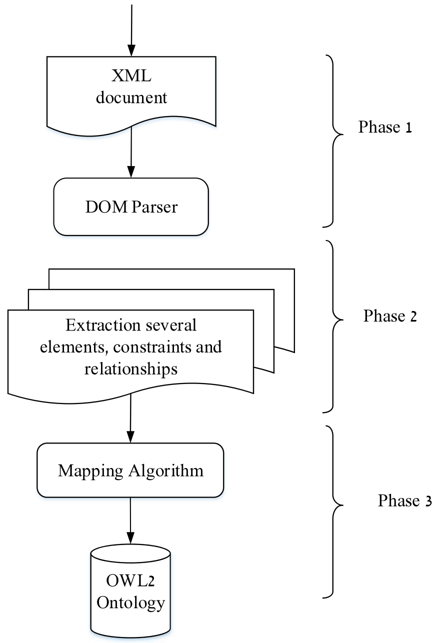

Figure 17. The General Framework of the El hajjamy et al. [29].

due to expressive power of RDF language. As a result, extracted OWL ontologies are richer than RDF ontologies. The output of the transformation process was often OWL1 ontological language except in El hajjamy et al. [29] method, target ontology is in OWL2 ontological language. In methods of Klein [14], Ferdinand et al. [15] and Breitling [23], the aim is that RDF triples generated and stored under ontology. Only in the method of Rodirues et al. [22], the input could be DTD, XML Schema or both. It should be noted that the two methods *i.e.* Tsinaraki et al. [19, 20] and Lacosti et al. [13] were inspired from bohering et al. [17] and also a method *i.e.* Jeiping et al. [24] was inspired from Tsinaraki et al. [19, 20]. Meanwhile, method of Hachouref et al. [12] completed the Pam et al. proposed method [8] and Yahia et al. method [26] is nearly similar to Ghawi et al. method [9]. Shenoy et al. method [27] resulted into generation of a very simple and elementary ontology and also, no rule for creation of the object property in the target ontology are considered. In the all reviewed papers, all transformation rules are expressed by using examples in a domain. To meet general purposed rules is better to use a language that independent from domains. In the methods of Tsinaraki et al. [19, 20] and Breitling et al. [23], the inverse transformation process is also considered. Generally, in the transformation process to evaluate the transformation system, the inverse transformation rules are defined and using the numbers of constructions of source and reversed source model, the precision and recall of the system are calculated. In this paper, eight criteria *i.e.* input type, output type, OWL ontology language type, transformation process type (automatic or semi-automatic), existence of instances generation phase, generation of ontology and instances, existence of integrity constraints and hierarchy in the target ontology. The ontology generated from XML instances contains primary structures like classes, data properties and object properties. In addition, the ontology generated from schemas is more structured than that generated from XML instances, for example it contains restrictions, constrains and hierarchy in addition to the primary structures. As results, the ontology generated from schemas is suitable for reasoning application. Also, there is no information loss in the methods transformed both schema and instances. Being automatic or semi-automatic of the reviewed methods was cited by the authors or inferred by the use of these two words from the main text. It should be noted that there is no formal factor(s) to closely determine boundary between being automatic or semi-automatic for the methods and the authors considered methods as automatic or semi-automatic depending upon extracted ontology needs how much handy work to be called complete. Also, it could be clearly understood that ontology extraction methods do not making use of intermediate model have better efficiency due to less overhead and also avoid complexity of turning models into each other and losing information and semantics during these conversions. Also, it should be finally noted that method of Hachouref et al. [12] is the best one in transformation methods based on DTD and method of bedini et al. [25] is the best one among transformation methods based on XML schema. In the XML schema category, Bedini et al. (2011) used 40 rules for their system while El hajjamy et al. [29] used only 23 rules, but method of El hajjamy et al. [29] has rules for conversion of transitive, binary and N-ary relations. In these three methods, it has been tried to translate maximum numbers of XML constructions to the equivalent ontology constructions with preserving semantic and minimum interaction of users. The selection measures for the best method in the two categories are number of transformation rules, richness of the extracted ontology which is related to proper transformation of XML constructions to equivalent ontology constructions with preserving semantic, automation of transformation process and also, lack of using intermediate model of that method. In Table 17, the reviewed methods with eight criteria are listed in chronological order from old to new. It is necessary noted that in the case that the name of proposed method was not mentioned in the paper, the authors names were used. The aim of each method and the constructions transformed by each method are mentioned in the Tables 18 and 19 respectively. In this paper, the transformation methods of XML documents to ontologies were expressed. Also in another article, the transformation methods of relational database to ontology were expressed [30]. In the future, the authors will aim to review ontology mediation methods so that extracted ontologies from data resources such as relational databases and XML documents could be interacted with each other.

References

[1] T. B. LEE, J. HENDLER, and O. LASSILA. The Semantic Web. *Scientific American*, 284(5):34–43, 2001. doi:https://www.jstor.org/stable/26059207.

[2] B. Shâdgâr, A. Osâreh, and A. H. Nezhâdi. Semantic Web: Concepts and Techniques. *Tehran, Armaghân Pub*, 2010.

[3] R. Navigli and P. Velardi. Learning Domain Ontologies for Document Warehouses and Dedicated Web Sites. *Computational Linguistics*, 30(2):151–179, 2004. ISSN 0891-2017. doi:10.1162/089120104323093276.

[4] M. M. Taye. Web-Based Ontology Languages and its Based Description Logics. *The Research Bulletin of Jordan ACM*, 2:1–9, 2011. ISSN 2078 - 7952.

[5] Markus Krötzsch, Bijan Parsia, Pascal Hitzler, Sebastian Rudolph, and Peter Patel-Schneider. OWL 2 Web Ontology Language Primer (Second Edition). W3C recommendation, W3C, December 2012. `http://www.w3.org/TR/2012/REC-owl2-primer-20121211/`.

[6] B. Glimm. Using SPARQL with RDFS and OWL Entailment. In *Reasoning Web. Semantic Technologies for the Web of Data*, pages 137–201. Springer, Berlin, Heidelberg, 2011. ISBN 978-3-642-23032-5. doi:10.1007/978-3-642-23032-5_3.

[7] K. M. Albarrak and E. H. Sibley. A survey of methods that transform data models into Ontology models. In *2011 IEEE International Conference on Information Reuse & Integration*, pages 58–65. IEEE, 2011. ISBN 978-1-4577-0965-4. doi:10.1109/IRI.2011.6009521.

[8] T. Pham, Y. K. Lee, and S. Lee. DTD2OWL: Automatic Transforming XML Documents into OWL Ontology. In *2nd International Conference on Interaction Sciences: Information Technology, Culture and Human*, pages 125–131. ACM, 2009. ISBN 978-1-60558-710-3. doi:10.1145/1655925.1655949.

[9] R. Ghawi and N. Cullot. Building Ontologies from XML Data Sources. In *20th International Workshop on Database and Expert Systems Application*, pages 480–484. IEEE, 2009. ISBN 978-0-7695-3763-4. doi:10.1109/DEXA.2009.68.

[10] P. T. T. Thuy, Y. K. Lee, S. Lee, and B. S.

Table 17. The Overall Comparison of the Reviewed Methods

Method Name	System Name	Input			Output	OWL Language Type	Rules	Existence of Instances Generation Phase	Generation of Ontology and Instances	Intermediate Model	Existence of Integrity Constraints	Existence of Hierarchy
		XML Schema	XML Document	DTD								
Klein [14]	Klein	✓	✓	×	RDF	-	Automatic	✓	Sequential	×	×	
Ferdinand et al. [15]	OWLMAP	✓	✓	×	OWL	DL	Automatic	RDF Graph	Sequential	×	✓	✓
Cruz et al. [16]	Cruz et al.	✓	✓	×	RDFS	-	Semi-Automatic	×	Not Available	×	✓	✓
Bohering et al. [17]	XML2OWL	✓	✓	×	OWL	DL	Automatic	✓	Sequential	×	✓	
Tsinaraki et al. [19, 20]	XS2OWL	✓	×	×	OWL	DL	Automatic	×	Not Available	×	✓	
Xu et al. [21]	Xu et al.	✓	×	×	OWL	DL	Semi-Automatic	×	Not Available	Relational Model	×	
Rodrigues et al. [22]	JXML2OWL	✓	✓	×	OWL	DL	Semi-Automatic	✓	Sequential	×	✓	✓
Pham et al. [8]	DTD2OWL	×	✓	✓	OWL	DL	Automatic	✓	Sequential	×	✓	
Ghawi et al. [9]	X2OWL	✓	×	×	OWL	DL	Automatic	×	Not Available	×	×	✓
Breitling [23]	XML2RDF	✓	✓	×	RDF	-	Semi-Automatic	×	Sequential	×	×	✓
Jeiping et al. [24]	Jeiping et al.	✓	✓	×	OWL	DL	Automatic	✓	Sequential	×	×	
Bedini et al. [25]	Janus	✓	×	×	OWL	OWL 2 RL	Automatic	×	Not Available	×	✓	✓
Lacoste et al. [13]	EXCO	✓	✓	×	OWL	DL	Automatic	✓	Parallel	×	✓	✓
Yahia et al. [26]	Yahia et al.	✓	×	×	OWL	DL	Automatic	✓	Not Available	×	×	✓
Shenoy et al. [27]	Shenoy et al.	✓	×	×	OWL	DL	Automatic	×	Not Available	×	×	
Hacherouf et al. [12]	Extended DTD2OWL	×	✓	✓	OWL	DL	Automatic	✓	Sequential	×	✓	✓
Tan et al. [28]	Tan et al.	✓	×	×	OWL	FULL	Semi-Automatic	×	Not Available	×	×	✓
El hajjamy et al. [29]	XSD2OWL2	✓	×	×	OWL	OWL 2	Automatic	✓	Not Available	×	✓	✓

Table 18. The Aim of the Reviewed Methods

Method Name	System Name	Purpose
Klein [14]	Klein	Ontology generation to extract knowledge from XML data sources
Ferdinand et al. [15]	OWLMAP	Ontology generation to produce semantic data
Cruz et al. [16]	Cruz et al.	Semantic integration of heterogeneous XML data sources
Bohering et al. [17]	XML2OWL	Bridge the gap between XML and OWL
Tsinaraki et al. [19, 20]	XS2OWL	Ontology generation for interoperability among XML-based application
Xu et al. [21]	Xu et al.	Ontology generation to produce semantic data
Rodrigues et al. [22]	JXML2OWL	Move from syntactic to semantic data organization
Pham et al. [8]	DTD2OWL	Ontology generation to achieve semantic interoperability
Ghawi et al. [9]	X2OWL	Ontology generation for information integration
Breitling et al. [23]	XML2RDF	Ontology generation for semantic computing
Jeiping et al. [24]	Jeiping et al.	Ontology generation to obtain semantic mapping
Bedini et al. [25]	Janus	Ontology generation to create a reference knowledge base to simplify B2B application integration
Lacoste et al. (2011)	EXCO	Ontology generation to achieve semantic interoperability
Yahia et al. [13]	Yahia et al.	Ontology generation for information integration
Shenoy et al. [27]	Shenoy et al.	Ontology generation to Resolve heterogeneity for data integration
Hacherouf et al. [12]	Extended DTD2OWL	Ontology generation to enhance quality of the search on the Web
Tan et al. [28]	Tan et al.	Ontology generation to exchange data in semantically uniform format
El hajjamy et al. [29]	XSD2OWL2	Ontology generation for annotation of Web resources

Table 19. The Constructions Transformed by Each Method

XML Constructions Declaration		The Reviewed Methods																	
XML Schema	DTD	Klein et al.	OWLMAP	Cruz et al.	XML2OWL	XS2OWL	Xu et al.	JXML2OWL	DTD2OWL	X2OWL	Breitling	Jeiping et al.	Janus	EXCO	Yahia et al.	Shenoy et al.	Extended DTD2OWL	Tan et al.	El hajjamy et al.
Element	<!ELEMENT x...	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Unique Element																			✓
Attribute	<!ATTLIST x att ...	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Attribute Group			✓							✓		✓	✓		✓				✓
Element Group												✓							✓
Enumeration																	✓		✓
Attribute ID Type																	✓		✓
Attribute IDREF Type																	✓		✓
Substitution Group			✓	✓									✓						
Complex Type	<!ELEMENT x (y...	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Simple Type	<!ELEMENT x (#CDATA...	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Group			✓							✓			✓		✓				
Sequence			✓		✓	✓						✓	✓	✓			✓		
Choice			✓		✓	✓						✓	✓	✓			✓		
All			✓		✓								✓	✓					
Union													✓						✓
Any													✓						
Extension			✓							✓			✓		✓				✓
Restriction			✓							✓			✓		✓				✓
Simple Content													✓						
Complex Content													✓						
Import													✓	✓					
Include													✓	✓					
Max/Min Occurs	REQUIRED/IMPLID/+/?/*		✓		✓				✓				✓	✓			✓	✓	✓
Mixed										✓					✓		✓		
Annotation	NOTATION					✓			✓								✓		
Namespace			✓																
	#FIXED value <!ENTITY entity-name ”entity-value”>								✓								✓		
Transitive Relation																			✓
N-ary Relation																			✓
Binary Relation																			✓

Jeong. Exploiting XML Schema for Interpreting XML Documents as RDF. In *International Conference on Services Computing*, pages 555–558. IEEE, 2008. ISBN 978-0-7695-3283-7. doi:10.1109/SCC.2008.93.

[11] P. T. T. Thuy, Y. K. Lee, S. Lee, and B. S. Jeong. Transforming Valid XML Documents into RDF via RDF Schema. In *Third International Conference on Next Generation Web Services Practices (NWeSP'07)*, pages 35–40. IEEE, 2007. ISBN 978-0-7695-3022-2. doi:10.1109/NWESP.2007.23.

[12] M. Hacherouf and S. N. Bahloul. DTD2OWL: A New Approach for the Transformation of the DTD to OWL. *Procedia Computer Science*, 62:457–466, 2015. ISSN 1877-0509. doi:10.1016/j.procs.2015.08.506.

[13] D. Lacoste, K. P. Sawant, and S. Roy. An Efficient XML to OWL Convertor. In *4th India Software Engineering Conference*, pages 145–154. ACM, 2011. ISBN 978-1-4503-0559-4. doi:10.1145/1953355.1953376.

[14] M. Klein. Interpreting XML documents via an

RDF schema ontology. In *13th International Workshop on Database and Expert Systems Applications*, pages 889–893. IEEE, 2002. ISBN 0-7695-1668-8. doi:10.1109/DEXA.2002.1046008.

[15] M. Ferdinand, C. Zirpins, and D. Trastour. Lifting XML Schema to OWL. In *International Conference on Web Engineering*, pages 354–358. Springer, Berlin, Heidelberg, 2004. ISBN 978-3-540-27834-4. doi:10.1007/978-3-540-27834-4_44.

[16] I. F. Cruz, H. Xiao, and F. Hsu. An ontology-based framework for XML semantic integration. In *International Database Engineering and Applications Symposium*, pages 217–226. IEEE, 2004. ISBN 0-7695-2168-1. doi:10.1109/IDEAS.2004.1319794.

[17] H. Bohring and S. Auer. Mapping XML to OWL Ontologies. In *Leipziger Informatik-Tage*, pages 147–156. GI, 2005.

[18] Charlie Halpern-Hamu. Transform a Sample Instance to a Schema. `http://incrementaldevelopment.com/papers/xsltrick/#instance-to-schema`. Date Accessed: 1 July 2015.

[19] C. Tsinaraki and S. Christodoulakis. Interoperability of XML Schema Applications with OWL Domain, Knowledge and Semantic Web Tools. In *OTM Confederated International Conferences On the Move to Meaningful Internet Systems*, pages 850–869. Springer, Berlin, Heidelberg, 2007. ISBN 978-3-540-76848-7. doi:10.1007/978-3-540-76848-7_57.

[20] C. Tsinaraki and S. Christodoulakis. XS2OWL: A Formal Model and a System for enabling XML Schema Applications to interoperate with OWL-DL Domain Knowledge and Semantic Web Tools. In *Digital Libraries: Research and Development*, pages 124–136. Springer, Berlin, Heidelberg, 2007. ISBN 978-3-540-77088-6. doi:10.1007/978-3-540-77088-6_12.

[21] J. Xu and W. Li. Using Relational Database to Build OWL Ontology from XML Data Sources. In *International Conference on Computational Intelligence and Security Workshops (CISW 2007)*, pages 124–127. IEEE, 2007. ISBN 978-0-7695-3073-4. doi:10.1109/CISW.2007.4425461.

[22] T. Rodrigues, P. Rosa, and J. Cardoso. Moving from Syntactic to Semantic Organizations using JXML2OWL. *Computers in Industry*, 59(8):808–819, 2008. doi:10.1016/j.compind.2008.06.002.

[23] F. Breitling. A Standard Transformation from XML to RDF via XSLT. *Astronomische Nachrichten(Astronomical Notes)*, 330(7):755–760, 2009. ISSN 2008-1123. doi:10.1002/asna.200811233.

[24] J. Tan and Z. Huang. Discovering OWL Ontologies from XML. In *3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*. IEEE, 2010. ISBN 978-1-4244-6542-2. doi:10.1109/ICACTE.2010.5579194.

[25] I. Bedini, C. Matheus, P. P. Schneider, A. Boran, and B. Nguyen. Transforming XML Schema to OWL using Patterns. In *ICSC 2011 - 5th IEEE International Conference on Semantic Computing*, pages 1–8. IEEE, 2011. doi:10.1109/ICSC.2011.77.

[26] N. Yahia, S. A. Mokhtar, and A. Ahmed. Automatic Generation of OWL Ontology from XML Data Source. *Computer Science*, 9(2), 2012.

[27] M. Shenoy, K. Shet, and D. U. Acharya. OWL based XML Data Integration. *Computer Applications*, 68(2):15–19, 2013. doi:10.5120/11550-6827.

[28] H. Tan, G. Barakat, and V. Tarasov. Translating XML models into OWL Ontologies for Interoperability of Simulation Systems. In *3rd Workshop on Ontologies and Information Systems (WOIS 2015)*, pages 116–123. CEUR-WS.org, 2015.

[29] O. E. Hajjamy, L. Alaoui, and M. Bahaj. XSD2OWL2: Automatic Mapping from XML Schema into OWL2 Ontology. *Theoretical and Applied Information Technology)*, 95(8):1781–1796, 2017.

[30] M. Pourkhani, B. Shadgar, and A. Osareh. A Study on the Methods of Relational Databases Transfer to Ontology. *Journal of Studies in Library and Information Science*, 21(13):51–68, 2014.

Mohammadereza Pourkhani graduated in M.S. of software computer engineering from ACECR of Khouzestan. His areas of interest are Semantic Web, Database Technology and Networking. He currently cooperate in control and maintenance of a data center in a governmental department and also deal with data analysis and statistics.

Bita Shadgar graduated in PhD of Web Engineering from Bristol university of UK. Her areas of interest are Semantic Web, Database Technology and Machine Learning.

Alireza Osareh graduated in PhD of Computer Science from Bristol university of UK. His areas of interest are Semantic Web, Machine Learning and Image Processing.