



## Mapping Relational Database to OWL Ontology Based on MDE Settings

Benamar Bouougada<sup>1</sup>, Djelloul Bouchiha<sup>1</sup>, Redha Rebhi<sup>2,3</sup>, Ali Kidar<sup>4,5</sup>, Giulio Lorenzini<sup>6\*</sup>, Abdelghani Bouziane<sup>1</sup>, Hijaz Ahmad<sup>7</sup>, Younes Menni<sup>8</sup>

<sup>1</sup> Dept. Mathematics and Computer Science, EEDIS Lab., Inst. Sciences and Technologies, Ctr Univ Naama, UDL-SBA, Naama 45000, Algeria

<sup>2</sup> Department of Mechanical Engineering, University of Medea, Faculty of Technology, Medea 26000, Algeria

<sup>3</sup> LERM - Renewable Energy and Materials Laboratory, University of Medea, Medea 26000, Algeria

<sup>4</sup> Faculty of Exact Sciences and Informatics, Ziane Achour University of Djelfa, P.O. Box 3117, Djelfa, Algeria

<sup>5</sup> Faculty of Sciences and Technology, Ziane Achour University of Djelfa, P.O. Box 3117, Djelfa, Algeria

<sup>6</sup> Department of Engineering and Architecture, University of Parma, Parco Area delle Scienze, 181/A, Parma 43124, Italy

<sup>7</sup> Department of Basic Science, University of Engineering and Technology, Peshawar 25000, Pakistan

<sup>8</sup> Unit of Research on Materials and Renewable Energies, Faculty of Sciences, Department of Physics, Abou Bakr Belkaid University, P.O. Box 119, Tlemcen 13000, Algeria

Corresponding Author Email: [Giulio.lorenzini@unipr.it](mailto:Giulio.lorenzini@unipr.it)

<https://doi.org/10.18280/ria.350305>

### ABSTRACT

**Received:** 17 April 2021

**Accepted:** 4 June 2021

#### Keywords:

*knowledge representation, knowledge engineering, model-driven-engineering, atlas-transformation-language, relational database*

Ontology is an important aspect of the semantic web, which is why semantic web developers are interested in constructing ontology in various applications based on domain experts. By transforming an existing application database into ontology, we many construct ontologies without having to hire an expert in the field. Model-driven engineering is the foundation of the suggested strategy (MDE). In a nutshell, the technique is divided into two phases, the first of which attempts to prepare the data needed for the transformation in the form of a model with a database. A compliance relationship between this model and its meta-model is required. Phase (2) applies a set of rules written in the Atlas Transformational Language to change the model produced in the first phase into another model, which is an OWL ontology (ATL). We tested our solution using a set of databases created specifically for this purpose and built it in an eclipse environment using an EMF and ATL transform language. The acquired findings demonstrate the strength and efficacy of the recommended strategy.

## 1. INTRODUCTION

Data is often stored on the web in databases, despite the many advantages of databases such as the way information is stored in an organized manner, which makes it very easy for humans to access it, allowing it to be exploited, but this is not sufficient, especially for the machines. Even though the data in databases is organized, the computer cannot exploit the data on its own.

The birth of the semantic web allowed the machine to process data on its own, through the existing technology in the semantic web such as ontology, which gave us many advantages from databases, the most important of which is the effective sharing of information and its reuse by the machine, thanks to the ontology structure.

Gruber's and Borst's definitions of ontology are the most well-known and fundamental: 'explicit statement of a conceptualization' [1] and 'formal statement of a shared conception' [2].

Borst has included the notion and expression of sharing in the ontology definition, which the conceptualization should have and must be written in a machine-readable style. Studer et al. [3] combined these definitions in 1998, calling it 'a formal and clear specification of a shared understanding' (p.25).

Many papers in the topic of ontology development are available from a variety of sources. Those focused on constructing an ontology from a natural language or PHP scripts, such as references [4-9], some of which rely on domain experts, and others which rely on databases, such as references [10-16]. Despite this huge amount of existing works, there is a lack or special requirements in order to create ontology.

In this post, we'll show how to construct OWL ontology from a database using a two-phase strategy based on MDE Settings. Where we will prepare a model comprising a database to be translated into OWL ontology in the first stage. In the second phase, we obtain OWL ontology by applying sets of transformation rules to the model obtained in the first phase.

The purpose of this approach is not only to transform the database or create OWL ontology but also to exploit existing knowledge to create ontology without resorting to the domain expert. This exploitation lies in the use or transformation of the application databases that contain knowledge in a specific field.

The rest of the paper is structured as follows: the next section discusses some of the works that have contributed to the development of ontology. The third section describes in great detail how to link a relational database to an OWL using MDE Settings. In the fourth section, we will also discuss the experimental results obtained by applying the proposed

approach. Finally, in the fifth section, we will provide a conclusion.

## 2. RELATED WORKS

Many ontology development works rely on the database as a resource, among which we mention the following:

In 2019, Tissot et al. [14] proposed a tool that provides a whole transformation architecture, from relational to ontology. The architecture consists of three components: The first component defines all necessary metadata and mapping scripts that store all information related to the mapping process. This component is called ‘General Architecture’. While the second component enriches the terminology produced using the logical and the physical models. The third and final component applies the rules of mapping that encompass many cases of transformation, such as NxN relationships, hierarchical relationships, constraints, and improving the ontology elements and instances.

Liao et al. [13] created a model tool in the Java environment that allows us to generate knowledge ontologies using relational schema metadata. The proposed method transforms various data elements into OWL components using the semantic relationship that is present between the ontology and the relational scheme.

Bouougada et al. in 2017 reported an approach that allows generating ontology and linked data from Web applications. They are based on the component of the Web application either server components or client components such as server PHP files and HTML files. The approach is concretized by a tool that proves its effectiveness [4].

Despite the fact that all of the above-mentioned studies allow for direct mapping of relational databases to OWL ontologies via a set of conversion rules, this is insufficient. Our approach takes the written relational database as SQL files to create the OWL ontology elements using MDE settings.

## 3. MDE-SETTINGS-BASED MAPPING RDB-TO-OWL

MDE Settings are used to map a relational database to an OWL ontology, as shown in Figure 1. The generation of the systems according to MDE settings often consists of two phases, so the proposed approach has two phases, namely the preparation of the input model (pre-processing) and the mapping (transformation process).

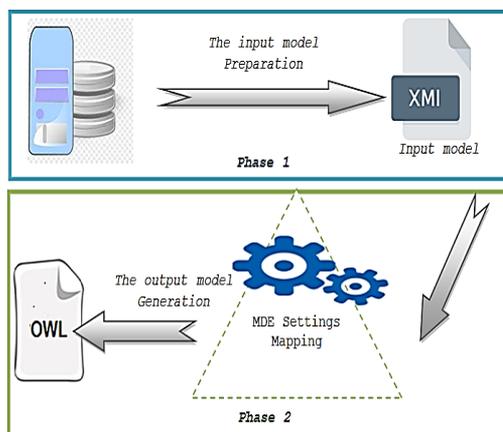


Figure 1. System architecture

## 3.1 Input model preparing phase

Listing 1 shows us an extract of the input model which created from the table "Customer". In this phase we will write an input model for a given relational database using a specific domain language.

The system automatically generates the input model from the database, which is a SQL file containing a set of tables and their contents (extensions). This ensures that the input model and the relational database meta-model are in sync. The input model from this stage was used in the mapping phase.

```

1. <?xml version="1.0" encoding="UTF-8"?>
2. <bdd:DATABASE xmi:version="2.0"
   xmlns:xmi="http://www.omg.org/XMI"
   xmlns:bdd="http://www.cuniv-naama.dz/BDD sample/"
3. name="http://www.semanticweb.org/ordinateur
   ologies/2018/11/Gestion_Customer">
4. <Table name="Customer">
5. <Attribute name="firstname" type="String"/>
6. <Attribute name="familyname" type="String"/>
7. <Attribute name="adress" type="String"/>
8. <Attribute name="age" type="Int">
9. <Constraint check_value="18" Op_compare=""/
10. </Attribute>
11. <Key name="IdCustomer" type="Int"/>
12. <Tuples type="Customer">
13. <Cell name="firstname" value="Ahmed"/>
14. <Cell name="familyname" value="BenAmed"/>
15. <Cell name="adress" value="NAAMA"/>
16. <Cell name="age" value="28"/>
17. <Cell name="IdCustomer" value="04500011"/>
18. </Tuples>
19. <Tuples type="Customer">
20. <Cell name="firstname" value="Amine"/>
21. <Cell name="familyname" value="Amine"/>
22. <Cell name="adress" value="NAAMA"/>
23. <Cell name="age" value="30"/>
24. <Cell name="IdCustomer" value="04500012"/>
25. </Tuples>
26. </Table>

```

Listing 1. The input model sample

## 3.2 The mapping phase

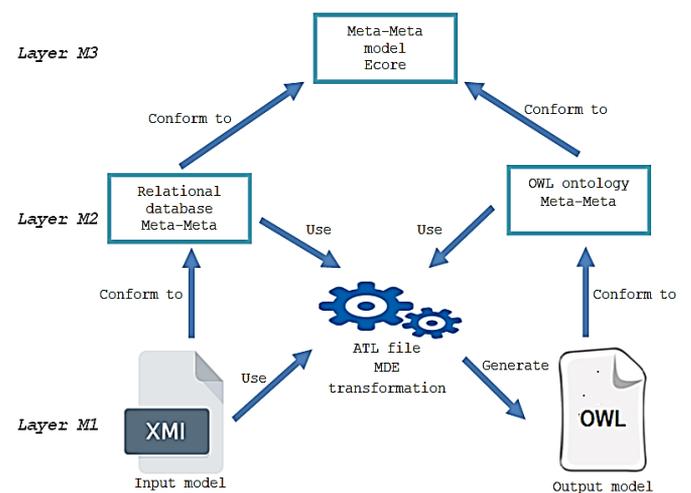


Figure 2. The mapping phase

Figure 2 shows the mapping phase or the transformation process phase that is based on MDE settings, so it consists of three layers where each layer contains models. The first layer

M1 contains the input model (obtained from the previous phase) and the OWL output model. The relational database meta-model and the OWL ontology meta-model are found in the second layer M2. The meta-meta-model (Ecore) is found at layer M3 [17]. In this phase, the system runs a set of transformation rules to produce an OWL output model.

In the next subsections, we will give a detailed description of these three layers.

### 3.2.1 Models layer (M1)

This layer consists of 2 models, the input model, and the output model; both of them are instance of the second layer models M2. The input model is the model obtained from the first phase (preparing phase), and the output model is the result of the mapping process of the input model. The output model represents the ontology as an OWL file.

### 3.2.2 Meta-models layer (M2)

In this layer, we are defining three components: two meta-models namely relational meta-model, OWL ontology meta-model, and the mapping that allows us to match between the elements of these meta-models.

The first component is the relational database meta-model is known as the source meta-model in the transformation based on MDE settings. Our proposed source meta-model is shown in Figure 3, it allows us a full representation of a given database schema and its extension, thus we can create an input model by using a specific domain language provided by this meta-model.

A relational database can be written as a set of ‘Table’ elements containing a collection of ‘Attribute’ components using this specific domain language. The extension (data) of the tables is represented by a set of ‘Tuple’ elements. Also, the relationships between tables are supported by the proposed meta-model through its representation by ‘Foreign Key’

elements. Additionally, constraints are also represented by this metadata model via ‘Constraint’ elements.

The second component is the OWL ontology meta-model or the target meta-model according to the MDE settings appellation is shown in Figure 4. In this meta-model, we just define the mapping elements that are required to generate an ontology from a database.

The ‘Class’ element, ‘DatatypeProperty’ element, ‘ObjectProperty’ element and ‘NamedIndividual’ element make up the structure. The mapping based on the MDE configuration or the transformation engine, which is a set of transformation ATL rules defined in Atlas Transformation Language (ATL) [18], is the third component. These rules allow elements from the source and destination meta-models to be matched.

The relational database meta-model, the OWL meta-model, the input model, and the mapping are sent to the transformation engine in order to generate an OWL ontology file (ATL file).

According to the number of the elements existed in source meta-model. We must define an ATL rule for each element, so we have to define six rules as shown in Table 1. Table 1 have five columns: the first column contains the rule number, the second and the third column contain the source and target element names, the fourth column discusses the rule, and the last column explains the mapping.

### 3.2.3 Meta-meta-models layer (M3)

The meta-meta-model layer, which is the MOF or the Ecore meta-meta-model language that existed in the EMF Framework [19], is the final layer in the MDE transformation.

After launching the transformation process, which takes all the components that we previously defined, we will get an OWL ontology file, which contains all the database information, whether schemas or data.

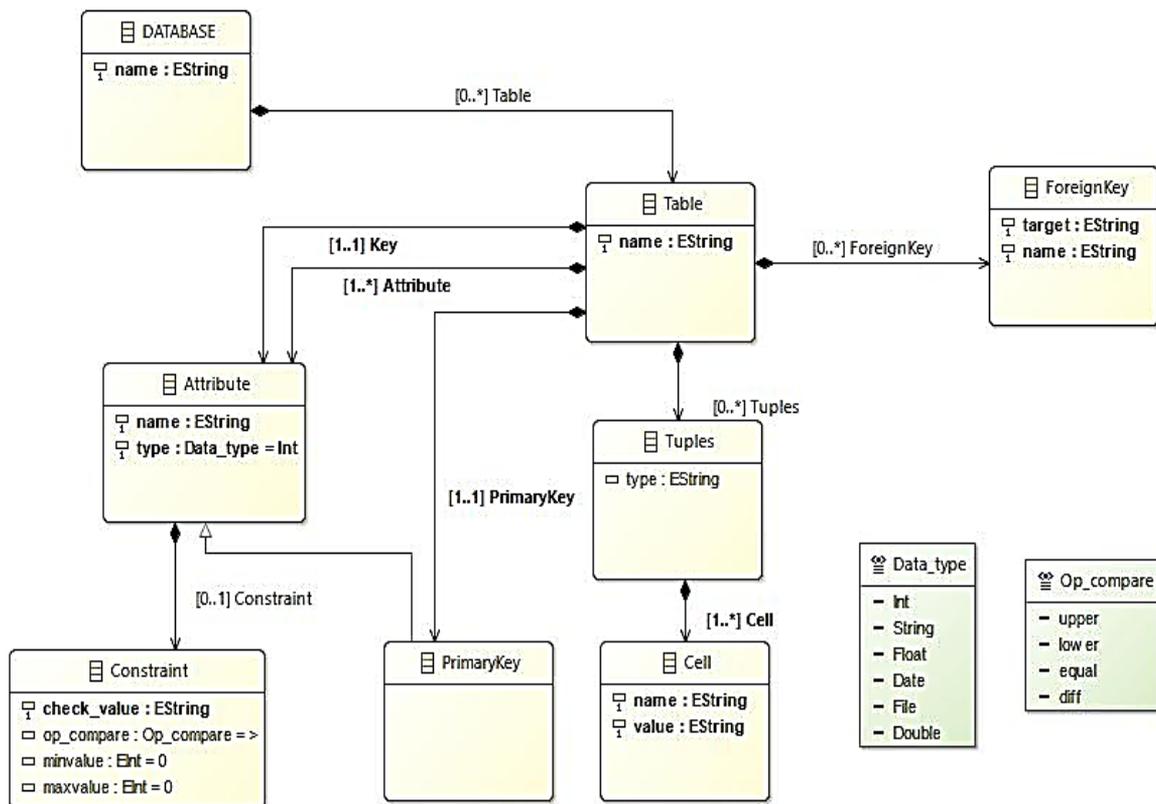


Figure 3. The source meta-model

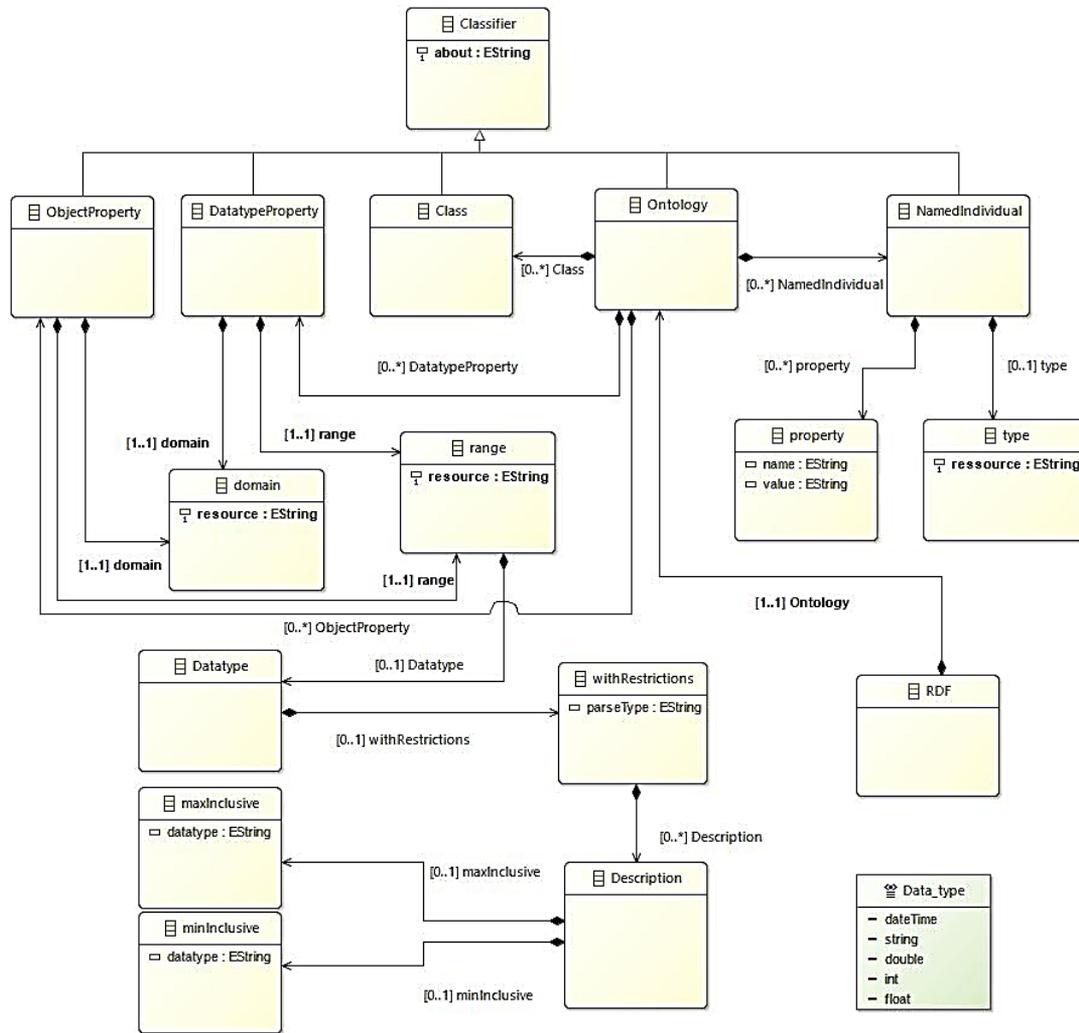


Figure 4. OWL ontology met-model

Table 1. The mapping

Rule number	Relational database meta-model elements (source element)	OWL meta-model elements (target element)	Rule description	Mapping
Rule 1	"Database", "name"	"Ontology", "about"	Produce an "Ontology" element through the "Database" element	"Database"."about" = "Ontology"."name"
Rule 2	"Table", "name"	"Class"	generating an OWL "Class" from the "Table" element	"class"."about"="Database"."Name"+"Table"."Name".
Rule 3	"Attribute", "name", "type"	"Data type property", "domain" and "range"	create a Data type property from the "Attribute" of the table	<ul style="list-style-type: none"> <li>"Data type property"."about" = "Ontology"."name"+"Attribute"."name".</li> <li>"domain"."resource" = "Ontology"."name"+"Table"."name".</li> <li>"range"."resource"="Ontology"."name"+"attribute"."type"</li> <li>"Object property"."about" = "Ontology"."name"+"Attribute"."Name".</li> </ul>
Rule 4	"Foreign Key", "target", "name"	"Object property", "Domain" and "Range"	Create an Object property from the "Foreign Key" element of the table	<ul style="list-style-type: none"> <li>"domain"."resource"="Ontology"."name"+"table"."name".</li> <li>"range"."resource" = "Ontology"."name" + "ForeignKey"."target"</li> </ul>
Rule 5	"Tuple", "type"	"NamedIndividual", "type"	instances generating as "NamedIndividual" elements from the extension of table (Tuple)	<ul style="list-style-type: none"> <li>"Namedindividual"."type" = "Ontology"."name"+"Tuple"."type".</li> <li>"properties"."name"="Cell"."name"</li> <li>"properties"."value"="Cell"."value"</li> </ul>
Rule 6	"Constraint", "minvalue", "maxvalue"	"Datatype", "withrestriction", "minInclusive", and "maxInclusive"	Create a restriction values on data type propriety	The element "minInclusive" takes the value if the attribute "minvalue", and the "maxInclusive" takes the value of the attribute "maxvalue"

#### 4. DISCUSSIONS OF EXPERIMENTS RESULTS

The proposed technique was implemented as a tool in the Eclipse environment, which included two plug-ins: The Eclipse Modeling Framework (EMF) and the Atlas transformation language (ATL).

In order to evaluate the implemented tool, we have created a set of databases so that we can prove the effectiveness of the tool. Note that in Table 2 we present only the five database results due to the convergence and similarity of the results across all databases.

First, we enter the database into the tool in the form of a SQL file, and then we launch the preprocessing phase to create an input file automatically. Second, we'll start the mapping process based on MDE parameters, which will result in an OWL ontology file that describes the database that was previously inputted.

We generate some criteria for each database conversion outcome that allow us to evaluate the results. These results are presented in Table 2 using the evaluation metrics Precision (1), Recall (2), and F-score (3) [20], which are stated in the equations:

$$\text{Precision} = \frac{\text{Relevant OWL elements}}{\text{Retrieved OWL elements}} \quad (1)$$

$$\text{Recall} = \frac{\text{Relevant OWL elements}}{\text{Database elements}} \quad (2)$$

$$\text{F - score} = \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

The number of retrieved OWL elements (classes, properties, and instances) contained in the OWL file obtained through the tool is the retrieved elements (OWL classes, data type properties, object properties, and individual).

The number of correct elements (OWL classes, data type properties, object properties, and individuals) among the recovered OWL elements is known as relevant OWL elements (classes, properties, and instances).

Database elements (tables, constraints, attributes, and extensions): represents the number of database elements (tables, constraints, attributes, and extensions) that the tool should return.

From the values of the experimental results, we can mention that the tool provides us a complete transformation, that is, it maps all the database elements of the input model and this explains why we got 1 as precision for all databases results. In addition, the mean F score is 0.96, which means that the proposed approach gave us good results that are very encouraging.

When we want to convert several databases with the same domain, we transform each one separately, resulting in a set of ontologies defining the same domain. In this case, we must use some ontology alignment techniques, such as the Refs. [21, 22], to integrate all of these ontologies and ensure the consensual aspect of the global ontology.

**Table 2.** Experimental results

		BDD 1	BDD 2	BDD 3	BDD 4	BDD 5
<b>computational Parameters</b>	<b>Retrieved OWL elements:</b> classes, properties, and individual	90	155	70	125	180
	<b>Relevant OWL elements:</b> classes, properties, and individual	90	155	70	125	180
	<b>Database elements:</b> tables, attributes, constraints, and extensions	100	160	75	140	190
<b>Evaluation Metrics</b>	<b>Precision</b>	1.00	1.00	1.00	1.00	1.00
	<b>Recall</b>	0.90	0.97	0.93	0.89	0.95
	<b>F-score</b>	0.95	0.98	0.97	0.94	0.97

#### 5. CONCLUSION AND PERSPECTIVES

In the classic web, web application data often reside in databases, which make them not automatically usable by devices. While semantic web technologies such as ontology give us the ability to make data accessible and usable by machines automatically, then our goal is to contribute to the link between classic web and semantic web by converting application databases that existing in the classic web into ontologies.

Through this article, we propose a model-based approach that allows us convert classic web applications to semantic web applications. The proposed approach consists of two important and complementary phases, which are the preparation of an input model and the transformation or mapping phase. The first part tries to generate an input model based on a text file written in SQL that contains the database we want to convert, while the second step completes the first phase by converting the input file to ontology using MDE settings.

The proposed method was supplemented by a tool that has proven to be quite effective, as evidenced by the promising results gained during the conversion of a set of databases to

OWL ontology. We will conduct a comparative analysis with related publications in the future.

#### REFERENCES

- [1] Gruber, T.R. (1993). A translation approach to portable ontologies specifications. *Knowledge Acquisition*, 5(2): 199-220.
- [2] Borst, W.N. (1997). Construction of engineering ontologies for knowledge sharing and reuse. Faculty of Electrical Engineering, Mathematics & Computer Science, University of Twente, Enschede the Netherlands.
- [3] Studer, R., Benjamins, V.R., Fensel, D. (1998). Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, 25(1-2): 161-197. [https://doi.org/10.1016/S0169-023X\(97\)00056-6](https://doi.org/10.1016/S0169-023X(97)00056-6)
- [4] Bouougada, B., Bouchiha, D., Bouziane, A., Malki, M. (2017). Ontology authoring and linked data generation from web applications. *International Journal of Strategic Information Technology and Applications (IJSITA)*, 8(4): 52-66. <https://doi.org/10.4018/IJSITA.2017100105>

- [5] Denaux, R., Dimitrova, V., Cohn, A.G. (2013). Interacting with ontologies and linked data through controlled natural languages and dialogues. Do-Form: Enabling Domain Experts to Use Formalised Reasoning-AISB Convention 2013, pp. 18-20.
- [6] Kaljurand, K. (2008). ACE view-An ontology and rule editor based on attempto controlled English. In Proceedings of the 5th OWL Experiences and Directions Workshop (OWLED 2008), Karlsruhe, Germany, pp. 1-12.
- [7] Kuhn, T. (2008). Acewiki: A natural and expressive semantic wiki. arXiv preprint arXiv:0807.4618. <https://doi.org/10.5167/uzh-26227>
- [8] Kuhn, T. (2013). A principled approach to grammars for controlled natural languages and predictive editors. *Journal of Logic, Language and Information*, 22(1): 33-70. <https://doi.org/10.1007/s10849-012-9167-z>
- [9] Kuhn, T. (2013). The understandability of OWL statements in controlled English. *Journal of Semantic Web - Linked Data for Science and Education*, 4(1): 101-115. <https://doi.org/10.3233/SW-2012-0063>
- [10] Dadjoo, M., Kheirkhah, E. (2015). An approach for transforming of relational databases to OWL ontology. arXiv preprint arXiv:1502.05844. <https://doi.org/10.5121/ijwest.2015.6102>
- [11] Farid, H., Khan, S., Javed, M.Y. (2016). DSont: DSpace to ontology transformation. *Journal of Information Science*, 42(2): 179-199. <https://doi.org/10.1177/01655515155591406>
- [12] Hazber, M.A.G., Li, R., Gu, X., Xu, G.D. (2016). Integration mapping rules: Transforming relational database to semantic web ontology. *Applied Mathematics Information Sciences an International Journal*, 10(3): 1-21. <https://doi.org/10.18576/amis/100307>
- [13] Liao, C.H., Wu, Y.F., King, G.H. (2019). Research on learning owl ontology from relational database. In *Journal of Physics: Conference Series*, 1176(2): 022031. <https://doi.org/10.1088/1742-6596/1176/2/022031>
- [14] Tissot, H., Huve, C.A.G., Peres, L.M., Fabro, M.D.D. (2019). Exploring logical and hierarchical information to map relational databases into ontologies. *International Journal of Metadata, Semantics and Ontologies*, 13(3): 191-208. <https://doi.org/10.1504/IJMSO.2019.099834>
- [15] YeePLY. Types de développement d'applications web. 2018. Available from: <https://fr.yeeply.com/blog/6-types-developpement-d-applications-web/>, accessed on May 27, 2021.
- [16] Al Khuzayem, L., McBrien, P. (2016). Owlrel: Learning rich ontologies from relational databases. *Baltic Journal of Modern Computing*, 4(3): 466-482.
- [17] OMG, MOF: Meta Object facility (MOF™) 2.0. OMG Specification in Object Mangment Group, 2006.
- [18] Massimo, T. ATL/User Guide - The ATL Language. 2016. Available from: [http://wiki.eclipse.org/ATL/User\\_Guide\\_-\\_The\\_ATL\\_Language](http://wiki.eclipse.org/ATL/User_Guide_-_The_ATL_Language), accessed on 15/09/2018.
- [19] Steinberg, D., Budinsky, F., Paternostro, M. (2009). Eclipse Modeling Framework (EMF), ed. E. Gamma, L. Nackman, and J. Wiegand: Addison-Wesley Professional.
- [20] Powers, D.M.W. (2008). Evaluation: from precision, recall and f-factor to roc, informedness, markedness and amp; correlation. *J Mach Learn Technol*, 2: 2229-3981. <https://doi.org/10.9735/2229-3981>
- [21] Otero-Cerdeira, L., Rodríguez-Martínez, F.J., Gómez-Rodríguez, A. (2015). Ontology matching: A literature review. *Expert Systems with Applications*, 42(2): 949-971. <https://doi.org/10.1016/j.eswa.2014.08.032>
- [22] Ardjani, F., Bouchiha, D., Malki, M. (2015). Ontology-alignment techniques: survey and analysis. *International Journal of Modern Education & Computer Science*, 7(11): 67-78. <https://doi.org/10.5815/ijmecs.2015.11.08>