An Approach for Merging Domain Ontologies Using Semantic and Orphan Mapping Technique

Muhammad Ahsan Raza[†], Binish Raza[††], Muhammad Akbar[†††], and Taiba Jabeen[††††]
ahsan_0136@yahoo.com
[†]Department of Information Technology, Bahauddin Zakariya University, Multan, Pakistan
[††]Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia
[†††]Institute of Computer Science and Information Technology, The Women University, Multan, Pakistan
[††††]Faculty of Education, Allama Iqbal Open University, Multan, Pakistan

Summary
Research on ontology merging has prompted numerous questions, such as, do existing studies consider domain semantics, do the constructs of different ontologies merge, and how can resultant ontologies be validated. We intend to devise an ontology-merging approach to answer these questions. We analyze the vocabulary of ontologies semantically and use the WordNet ontology for semantic orientation. The proposed approach presents ontology-merging algorithms whereby the concepts and properties in similar domain ontologies are mapped to merged ontology. In addition, the approach also performs orphan mappings to cover the unmapped vocabulary in ontologies. Our approach shows that the resultant ontology is error free, consistent, and mirrors the semantics of a domain completely. We conclude that addressing the semantic mapping of ontology vocabulary can achieve worthwhile improvements. Furthermore, we apply our methodology to the library domain and provide suggestions for future research.

Key words:
Knowledge Ontology Alignment, Semantic Web, Ontology Matching, Library Ontology.

1. Introduction

The purpose of the Semantic Web vision is to make the existing web meaningful, where data can be searched in an efficient manner. The Semantic Web relies on the use of ontology to represent meaningful data. Ontology provides a common understanding of terms in a domain. This process involves the explanation of basic concepts in a domain as well as their machine-interpretable definitions.
According to Gruber, "An ontology is an explicit specification of a conceptualization" [1]. In this definition, the term explicitly states that concepts should be presented in machine understandable formats. Vanitha et al. [2] provided another definition for ontology, describing it as a graph consisting of a finite set of nodes N. These nodes are linked with one another via edges, in which each edge provides a direct connection between two nodes of the graph. A node represents a specific domain class, whereas edges represent relationships among different classes.
Researchers have developed numerous ontologies [3] [4]; thus, more than one conceptualization (i.e., ontologies) may exist for the same domain. Therefore, the need to merge two or more ontologies to bridge the gap between different conceptualizations of the same domain has emerged. Ontology merging involves the creation of a common vocabulary for a domain using two or more existing domain ontologies [5]. A common ontology is useful, as it removes the inconsistencies of an individual ontology and supports interpretability among information services. Different techniques have been proposed for merging ontologies, in which the hierarchical relationship structure of an ontology, semantic (synonym) knowledge, or a combination of both are exploited to combine the vocabularies of ontologies [6] [8]. However, existing techniques lack in handling unmatched components of ontologies in the process of ontology merging [6] [13]. Furthermore, a concept's instances can better explain the semantics of that ontology concept. However, efforts focusing on the use of instances in the process of merging ontologies also remain limited [7] [14].
This study develops an approach for merging two same-domain ontologies while considering the semantics and instances of ontology concepts. Specifically, the proposed approach emphasizes on merging the orphan vocabulary (i.e., the unmatched concepts and properties) of ontologies in order to reduce the inconsistencies from the merged ontology. Furthermore, we are interested in evaluating merged ontology concepts and properties to determine the effectiveness of our approach. Thus, we use the results of reasoning test on the final merged ontology vocabulary.
The rest of this paper is organized as follows. Section 2 provides an overview of the literature on ontology merging. Section 3 briefly discusses the proposed approach and its components. Section 4 describes the experiments and results, and the final section concludes the work and presents suggestions for future directions.

Manuscript received April 5, 2020
Manuscript revised April 20, 2020

2. Literature Survey

Yuan and Tripathi [9] concluded that ontology is developed and employed in different fields of interest (i.e., software engineering, the Semantic Web, and natural language processing). Furthermore, numerous ontologies have been constructed for a single domain, especially in the field of requirement engineering. The authors in [9] argued that merging different ontologies into a common ontology can enhance requirement engineering procedures. Thus, the authors presented a methodology that summarized existing ontologies into a new ontology and showed an improvement in engineers' works using an ontology knowledge base.

Pavel and Euzenat [10] presented an alternative ontology-merging technique to support the Semantic Web vision. Through logical and analytical comparisons of different applications using individual and merged ontologies, the authors noticed that the application using merged ontology yields better results.

Leung et al. [11] provided another research vein on ontology merging. The authors' proposed technique adopts a three-layer framework for merging domain ontologies. The input layer (i.e., the first layer) parses the input into elementary (i.e., keywords) and structural (i.e., taxonomic relationships) inputs. The second layer, namely, the basic layer, provides a guideline for the executive layer (i.e., the third layer) using seven ontology-matching methods. Finally, the executive layer constructs a common conceptual ontology using different reasoning techniques. The technique is also useful for identifying appropriate ontology-matching methods for merging ontologies.

A common information format is necessary in the domain of Internet of things (IoT) for communication among IoT devices. To support the vision of global IoT in which Iot devices at different levels can interoperate, another facet of ontology-merging focused on the semantic exchange of data among IoT devices [12]. The authors adopted an ontology-merging technique and showed improved interoperability among IoT devices using a common merged ontology.

Lambrix and Kaliyaperumal [15] presented an alignment process that merges ontologies in different sessions, including mapping, filtration, and suggestion sessions. Furthermore, the authors realized that the process of mapping and generating high-quality alignments is impossible without user involvement. User judgments are necessary during the alignment of ontology vocabularies. Overall, the proposed session-based ontology-merging framework demonstrates better performance compared with classical ontology-merging techniques.

3. Ontology-merging Framework

Our model consists mainly of two main components, namely, a concept merger and a properties merger, and performs merging without user intervention. Fig. 1 illustrates the functions of the three main components of the proposed approach. The method proceeds with the input of two ontologies. First, it merges the two ontology concepts. Next, it maps the relationships (i.e., data properties and object properties) among the concepts. Finally, a merged ontology is obtained as the output of the method. The main steps of each component of the proposed approach are described as follows.

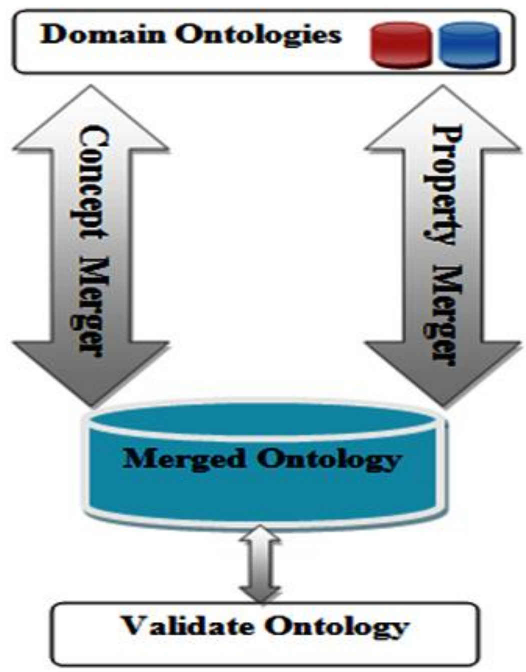

Fig. 1 Main components of ontology-merging framework

3.1 Concept Merger

This component merges the concepts of the two domain ontologies based on semantic information obtained from a knowledge base following the two simple steps. The details of these steps are as follows (see Fig. 2).

1) Concept Normalization

The first step of the concept merger module normalizes the ontology concepts. After the normalization, the individual concepts are divided into two types of tokens, namely, (1) word tokens and (2) non-word tokens (e.g., punctuations). The two main tasks of this phase are as follows.

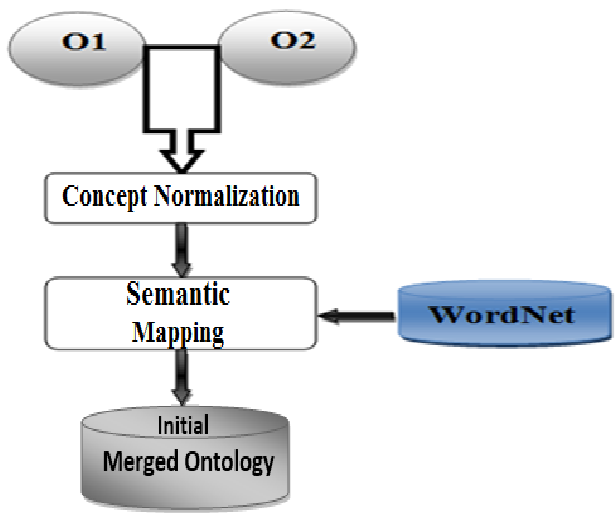

Fig. 2 Steps of concept merger module

a) Removing stop words

Unwanted words, such as “an,” “are,” “the,” and so on, introduce noise to the concepts. Such words must be filtered from the individual concepts to extract only useful word tokens. Next, these tokens are used to map the concepts of the two dissimilar ontologies.

b) Stemming word tokens

This task normalizes the ontology concepts further by removing suffixes or affixes from the concepts. After stemming, we obtain root words for concepts, thereby making it easy to match concepts in the ontologies. For instance, the ontology concept “Books” is stemmed to the concept “Book” after preprocessing. Similarly, the concept “conference proceedings” is transformed into “conference proceeding.”

2) Semantic mapping

Semantic mapping is the main phase of the concept merger component, which involves a string comparison of concept names between the ontologies. For a meaningful comparison of the concepts of the two ontologies, we rely on concept instances and an ontology called WordNet, which is a widely used English lexical database. Fig. 3 briefly describes the steps of the algorithm with which the entire semantic mapping procedure is executed. Each concept is inputted into WordNet to obtain the synonymy relations of concepts (synsets). Once all the concepts’ synsets are identified, similarities among them along with the concepts’ instances are computed based on the synsets lexical analysis (called as semantic similarity).

We take two concepts, namely, “PHD thesis” and “Doctorate thesis,” from Ontologies 1 and 2, respectively, to illustrate the function of the semantic analysis submodule. We obtain a synonym list of the two concepts using WordNet, as shown in Table 1. Using the concept synonyms, we calculate the similarity score using Eq. 1. Finally, the high score concepts are merged in the resultant ontology.

$$Sim\,(c1, c2) = path\_similarity\,(C1_{Syn\_list}\,,\, C2_{Syn\_list}), \qquad (1)$$

where C1Syn_list and C2Syn_list represents the synonyms list for concept 1 and concept 2, respectively.

For orphan concepts, superclass is computed for each orphan concept. Finally, these concepts are inserted as child concepts of their corresponding base class.

Algorithm: Semantic Analysis

Input: first ontology (O1), second ontology (O2)

Output: initial merged ontology (M)

```
FOR each concept Ci in Ok // where k = 1 or 2
    Compute Instances and Synset from WordNet
END-FOR
FOR each concept Ci of O1
    FOR each concept Tj of O2
        Compute semantic_similarity(Ci, Tj)
        IF similarity found THEN
            Merge (Ci, Tj) in M
            Break
        END-IF
    END-FOR
END-FOR
// for unmatched concepts
FOR each unmatched concept  Ci
    Extract base class
    Insert Ci as child class of base class in M
END-FOR
RETURN    M
```

Fig. 3 Algorithm for semantic analysis

Table 1: Synonym list of concepts using WordNet

Sr. No.	*Concept*	*Synonyms*
1.	PHD thesis	PhD, doctor, doctorate, postgraduate, degree
2.	Doctorate thesis	PhD, advanced degree, doctoral degree, doctorate, master, master degree, postgraduate degree

3.2 Properties Merger

To merge the properties of two same-domain ontologies, first, the properties merger component divides the properties of each ontology into data and object properties. Next, properties merging occurs in two phases, namely, semantic mapping and orphan mapping, as shown in Fig. 4.

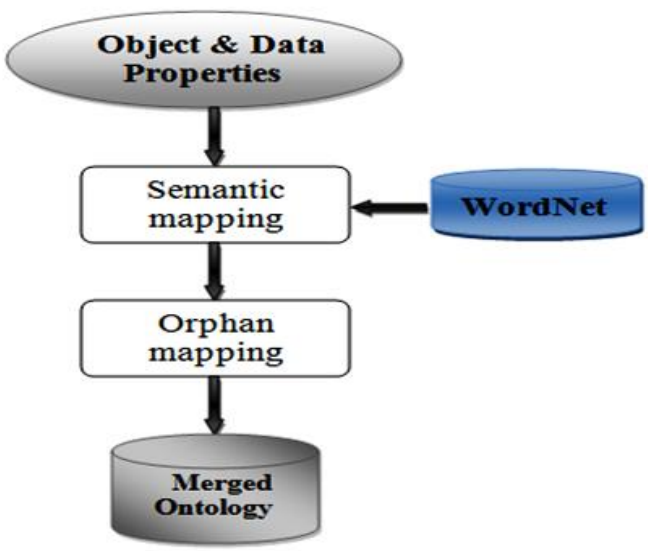

Fig. 4 Phases of properties merger module

1) Semantic mapping

This phase involves the mapping of two properties based on their meanings (semantic knowledge). If two properties are matched, then they are merged in the resultant ontology.
First, synsets are extracted from WordNet for each property of a concept. Next, lexical matching is performed using the synsets of two properties (see Fig. 5). The same procedure is repeated for other merged concepts of the ontology.

Algorithm: Property merger

Input: first ontology (O1), second ontology (O2)

Output: merged ontology (M)

```
1  FOR each concept Ci in Ok // where k = 1 or 2
2      Extract properties Pi of Ci
3      Compute Synset (Pi) from WordNet
4      DO semantic-mapping (Pi ∈ O1, Pj ∈ O2)
5      Merge Pi in M
6  END-FOR
7  FOR each unmapped properties Pi
8      DO orphan-mapping ( Pi, M)
9  END-FOR
```

Fig. 5 Algorithm for merging properties

2) Orphan mapping

This phase covers the merging of orphan properties (i.e., no match is obtained via semantic mapping). In this phase, the properties of the concepts (either belonging to Ontology 1 or Ontology 2) are added in the resultant merged ontology (see line 7−9 in Fig. 5). At the end of this phase, all the unmapped properties of Ontology 1 or Ontology 2 are inserted in the merged ontology.

3.3 Ontology Validation

Once the two ontologies are merged into a new ontology, the consistency of the new merged ontology (i.e., duplicate concepts, unused concepts, and missing edges) must be verified. Thus, we perform a simple reasoning test on the resultant merged ontology using the Fact++ reasoner [16]. This reasoner can help guarantee that the ontology concepts are valid and connected sensibly to one another.

4. Results and Discussion

To illustrate the proposed framework, we use two domain-specific ontologies that describe the vocabulary of the library domain. Fig. 6 demonstrates the portions of two ontologies, in which the first ontology comprises 13 concepts, and the second ontology consists of 09 concepts. The “is–a” link connects the different concepts of each ontology. Each of the given input ontologies is processed through the different steps of our framework to merge the valid concepts of the input ontologies as well as their semantic relationships into the new merged ontology.
To obtain a merged ontology, first, we create two different library ontologies in Ontology Web Language (OWL) using the Protégé ontology editor. We save these

ontologies as separate files named CS_Library.owl and Library.owl. Second, we merge these two ontologies using the components of our methodology. In the concept merger module, we proceed with the algorithm and perform semantic analysis on the concepts of the two ontologies. This analysis can facilitate the extraction of semantically similar concepts. First, we normalize the concepts into simple forms then use the WordNet ontology to compute semantics (synonyms) related to each concept. Second, we merge similar concepts based on concept synonyms.

Table 2 presents a list of merged concepts between the ontologies, where “articles” (concept in Ontology 1) is merged with “publication” (concept in Ontology 2) as the “article” concept in the new merged ontology. Furthermore, we add the term “publication” as a synonym relationship with the merged “article” in the new ontology for semantic orientation. Similarly, we insert a synonym relationship between the concepts “doctoral thesis” and “PHD thesis” to provide all the semantics of the library domain.

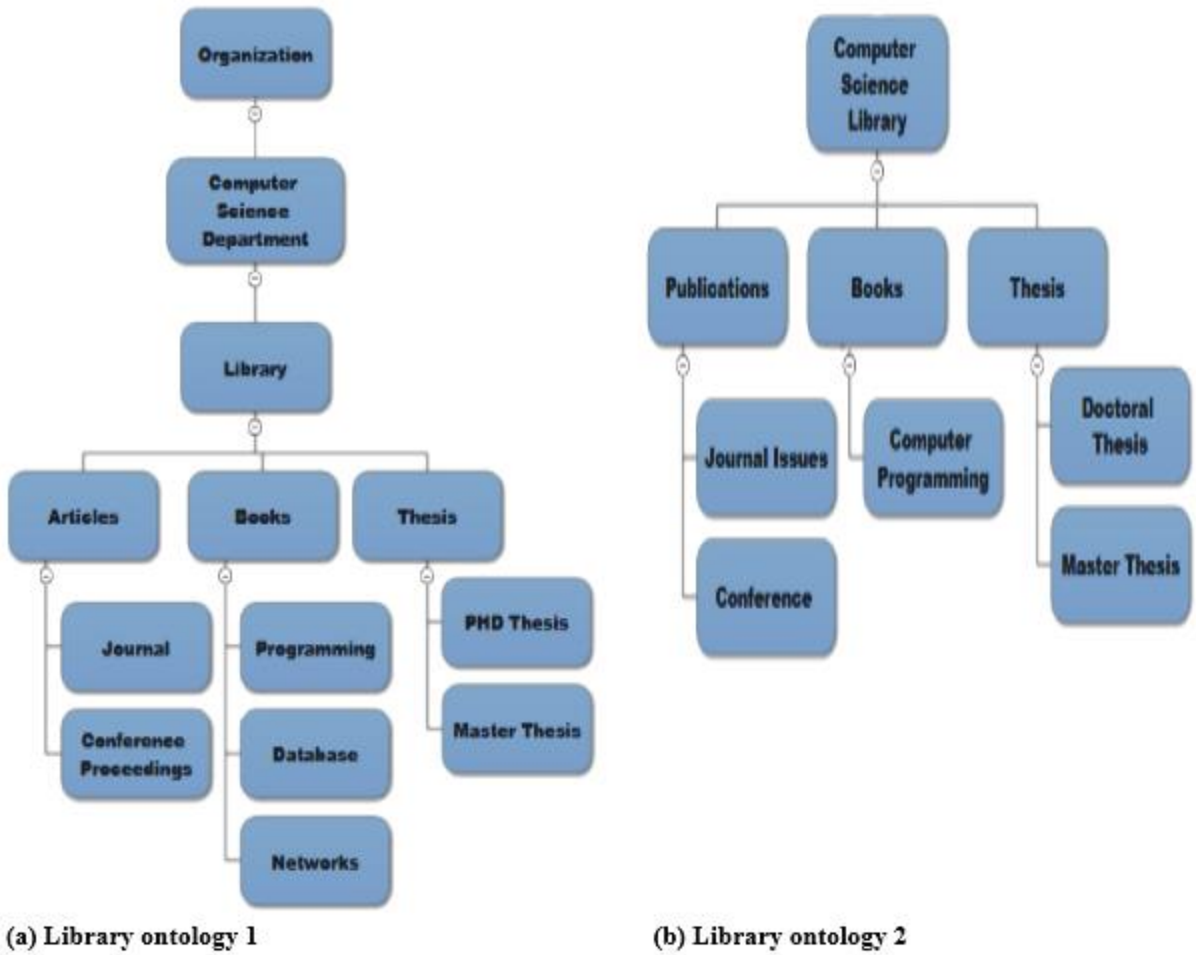

Fig. 6 Input library ontologies

Table 2: List of merged concepts

Ontology 1 Concepts	***Ontology 2 Concepts***	***Merged Concepts***
Library	Computer science library	Computer science library
Articles	Publication	Article
Books	Books	Book
PHD thesis	Doctorate thesis	PHD thesis

Programming	Computer programming	Programming
Journal	Journal issues	Journal issue
Thesis	Thesis	Thesis
Master thesis	Master thesis	Master thesis

Certain concepts (e.g., "organization," "database," and "computer networks") remain unmapped after the semantic analysis. For these concepts, corresponding base concepts (ancestors) are extracted from their respective ontologies (Ontology 1 and Ontology 2) and inserted in the merged ontology as a subclass relationship on the basis of the base concepts. For example, "organization" is mapped as a subclass of the "Thing" concept (i.e., root class of OWL ontology).

Subsequently, we proceed with merging the properties of the two source ontologies. The algorithm continues with the semantic mapping of each property of Ontology 1 with that of Ontology 2. Once again, we use the WordNet ontology to perform this task. Table 3 presents the merged and unmerged properties (i.e., orphan properties) of the thesis concept. For the unmerged concepts (i.e., not mapped to any property), we simply insert them as a new property in the new ontology.

After merging the concepts and properties of the two source ontologies, we verify the resultant merged ontology in terms of concept duplication, property duplication, and consistency. We use the Fact++ semantic reasoner to test the newly merged ontology and find that the merged library ontology obtained as the output of our framework is 100% consistent.

Table 3: List of properties of thesis concept

Merged Property	***Orphan Property***
Title	Department
Submission data	Main supervisor
	Co supervisor

The final merged ontology for the library domain comprises 155 concepts organized via is–a and synonym relationships. Furthermore, it consists of 310 data type properties (e.g., five properties of the thesis concept as listed in Table 3). Fig. 7 shows a snapshot of the resultant ontology from the OntoGraph tool.

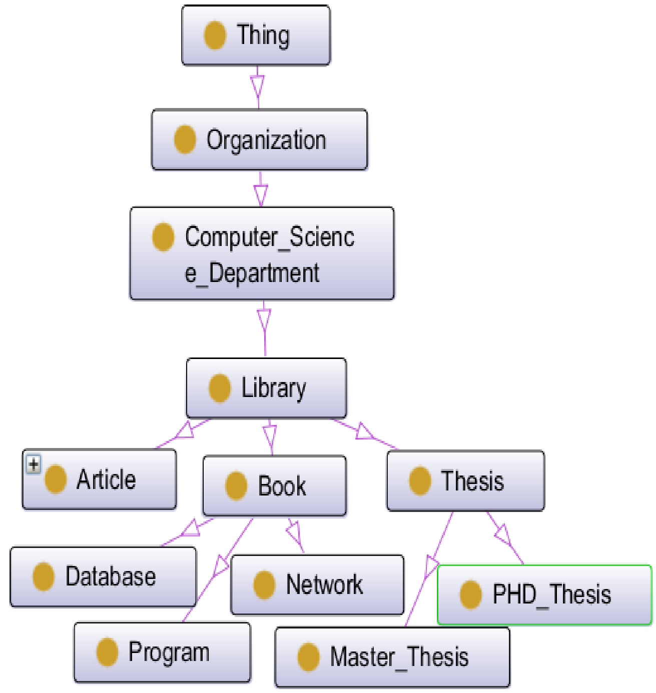

Fig. 7 OntoGraph view of resultant merged ontology

5. Conclusion and Future Work

The Semantic Web vision is based on the use of ontology describing the knowledge of a domain in terms of concepts as well as relationships among concepts. When various ontologies reflect the semantics of the same domain, misunderstandings, differences, and inconsistencies arise. As a result, combining same-domain ontologies into a common ontology to enhance domain semantics spanning multiple ontologies is necessary.

This research proposes an ontology-merging approach with the aim of combining two same-domain ontologies while maintaining all the semantics of the domain. The methodology outlines two semantic modules, namely, the concept merger and properties merger, in which semantic analysis is performed on the vocabulary of two ontologies to create a common ontology vocabulary. The approach uses the WordNet knowledge base to resolve vocabulary ambiguities and relies on a semantic reasoning engine to test the resultant ontology. This ontology-merging framework merges two ontologies from the library domain and achieves 100% consistency in the vocabulary of the resultant merged ontology.

Our proposed technique suggests general algorithms for ontology vocabulary merging (i.e., concept mapping and properties mapping) such that it can be applied to other domains. In the future, we will attempt to extend our framework by proposing mappings for the cardinalities of ontology relationships.

References

[1] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing?," International Journal of Human-Computer Studies, vol. 43, no. 5, pp. 907-928, 1995/11/01/, 1995.

[2] K. Vanitha, M. S. Venkatesh, K.Ravindra, and S. V. Lakshmi, "The Development Process of the Semantic Web and Web Ontology," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 2, no. 7, 2011.

[3] S. Mishra, and S. Jain, "Ontologies as a semantic model in IoT," International Journal of Computers and Applications, vol. 42, no. 3, pp. 233-243, 2020/04/02, 2020.

[4] M. A. Raza, M. Rahmah, S. Raza, A. Noraziah, and R. A. Hamid, "A Methodology for Engineering Domain Ontology using Entity Relationship Model," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 10, no. 8, 2019.

[5] J. Euzenat, and P. Shvaiko, Ontology Matching: Springer, Berlin, Heidelberg, 2013.

[6] M. Priya, and K. Ch. Aswani, "A novel method for merging academic social network ontologies using formal concept analysis and hybrid semantic similarity measure," Library Hi Tech, 2019.

[7] S. Amirian, and S.Mohammadi, "Ontology Alignment using WordNet Method," International Journal of Computer Science and Network Security, vol. 17, no. 7, pp. 161-167, 2017.

[8] O. Verhodubs, "Merging of Ontologies Through Merging of Their Rules," arXiv preprint arXiv:2001.04326, 2020.

[9] X. Yuan, and S. Tripathi, "Combining ontologies for requirements elicitation." in 2015 IEEE International Model-Driven Requirements Engineering Workshop (MoDRE), 2015, pp. 1-5.

[10] S. Pavel, and J. Euzenat, "Ontology Matching: State of the Art and Future Challenges," IEEE Trans. on Knowl. and Data Eng., vol. 25, no. 1, pp. 158–176, 2013.

[11] N. K. Y. Leung, S. Kang, S. Lau, and J. Fan, "Ontology Matching Techniques: a 3-Tier Classification Framework," International Journal of the Computer, the Internet and Management, vol. 17, no. SP1, pp. 22.1-22.7, 01/01, 2009.

[12] M. Ganzha, M. Paprzycki, W. Pawłowski, P. Szmeja, and K. Wasielewska, "Towards Semantic Interoperability Between Internet of Things Platforms," Integration, Interconnection, and Interoperability of IoT Systems, R. Gravina, C. E. Palau, M. Manso, A. Liotta and G. Fortino, eds., pp. 103-127, Cham: Springer International Publishing, 2018.

[13] N. Chatterjee, N. Kaushik, D. Gupta, and R. Bhatia, "Ontology Merging: A Practical Perspective," In: Satapathy S., Joshi A. (eds) Information and Communication Technology for Intelligent Systems (ICTIS 2017) - Volume 2. ICTIS 2017. Smart Innovation, Systems and Technologies, vol 84, 2018, Springer, Cham

[14] G. S. Narula, R. Wason, V. Jain, and A. Baliyan, "Ontology mapping and merging aspects in semantic web," International Robotics and Automation Journal, vol. 4, no. 1, 2018.

[15] P. Lambrix, and R. Kaliyaperumal, "A Session-based Ontology Alignment Approach enabling User Involvement," Semantic Web Journal, vol. 8, pp. 225-251, 01/01, 2017.

[16] D. Tsarkov, and I. Horrocks, "FaCT++ Description Logic Reasoner: System Description," in Automated Reasoning, Berlin, Heidelberg, 2006, pp. 292-297.