Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666

Volume 6 Issue 22 (September 2021) PP. 148-157
DOI: 10.35631/JISTM.622012

JOURNAL OF INFORMATION SYSTEM AND TECHNOLOGY MANAGEMENT (JISTM)
www.jistm.com

ONTOLOGY MERGING USING PROTÉGÉ – A CASE STUDY

Zailan Arabee Abdul Salam[1*], Rabiah Abdul Kadir[2], Azreen Azman[3]

[1] Institute of IR4.0, Universiti Kebangsaan Malaysia, Malaysia
Email: zailanz@gmail.com
[2] Institute of IR4.0, Universiti Kebangsaan Malaysia, Malaysia
Email: rabiahivi@ukm.edu.my
[3] Department of Multimedia, Universiti Putra Malaysia, Malaysia
Email: azreenazman@upm.edu.my
[*] Corresponding Author

Article Info:

Article history:
Received date: 10.06.2021
Revised date: 15.07.2021
Accepted date: 20.08.2021
Published date: 01.09.2021

To cite this document:

Abdul Salam, Z. A., Kadir, R. A., & Azman, A. (2021). Ontology Merging Using Protégé – A Case Study. *Journal of Information System and Technology Management, 6* (22), 148-157.

DOI: 10.35631/JISTM.622012

Abstract:

The exponential growth of data and the boom of online businesses necessitates the need for data to be machine-readable, as humans are no longer able to manually manage the vast amounts of data. Ontologies can define concepts and relations that are amenable to processing by machines. Ontologies are created in silos and pockets of domains, and the need to merge these resources is key to universal access to multi-domain knowledge. Merging of ontologies has been explored to an extent over the last two decades, and this paper explores the extent of the tools and techniques available with a case study of merging two ontologies which are publicly available, the Person ontology and Institutional ontology, using the latest tools available on the most popular ontology editor, Protégé. It is found that automated merging tools have not been improved much over the last two decades, and the most current merging tools provided combine the two ontologies into one but do not unite or merge any of the classes or axioms which are equivalent. This can be seen in the axiom count, which does not decrease in the merged ontology, showing that no similar classes or actual axioms were merged. Protégé plugins which used to provide the semi-automatic mapping of similar classes to assist the merging process were found to be no longer available, and manual mapping by the knowledge engineer was required. This supports further research in automated ontology merging techniques.

Keywords:

Ontology, Merging, Protégé, Similarity, Integration

Introduction

Data growth has been exponential over the last few decades as transactions are recorded by information systems which have been prevalent in all industries with no exceptions. With the advent of enterprise information systems and the boom of e-commerce of late, knowledge representation methods such as controlled vocabularies, thesauri, taxonomies, and ontologies have been increasingly used to enhance the capabilities of these systems (Pawełoszek & Korczak, 2018). The web has been transforming from isolated data repositories to linked sources of information. As content presented on the Web is more often created by its users, the scale and heterogeneity of information on the Web is ever increasing, where user collaboration and information sharing is becoming essential for building web communities (Koprowska, 2008). Using a formal ontology makes the specification independent from what it describes and helps to avoid mistakes while modelling large domains (Guarino, 1998). The ability to reuse existing ontologies is what makes it particularly powerful, as each ontology can be applied to specify different aspects of the domain it models. Reusing depends on efficient methods of ontologies aligning and merging (Koprowska, 2008). The need for businesses to be discoverable on the internet is undeniable, and a product or service needs to be discoverable by machines, and these machines need to be able to make some sense of the data that is needed by the user, so that they can search for the relevant data on the web. This matching process between what the user wants and what is available on the web is what Semantic Web 2.0 aims to improve by establishing relationships between data which is understood by machines, so that the desired information can be extracted efficiently (Negi & Malik, 2018). Ontologies are a formal way of representing concepts in a domain and their relationships to other concepts in a machine-readable form. Machines would be able to access domain specific data from these ontologies with a certain level of semantics defined within them. Ontologies have been developed in many domains, but remain largely as virtual silos which prevent these resources from being queried, browsed, or leveraged in a truly uniform way (Cheatham, Cruz, Euzenat, & Pesquita, 2017). Merging of these ontologies which exist disjointedly will lead to a growth of interconnected, multi domain knowledge bases, which will enhance access to knowledge, which is machine readable, making information more accessible to both machines and their human users. This paper aims to review recent tools that are available to perform ontology merging. A case study of merging two ontologies which have similar concepts using Protégé ontology editor as well as OWL API will be presented. An overview of ontology merging concepts will be discussed, followed by a case study of merging two ontologies using the Protégé merging tool, where the results of the merging will be discussed.

Literature Review

For the scope of this paper, three main areas will be discussed which are directly related to Ontology Merging. These are Ontologies, Merging Tools, and Protégé Ontology editor.

Ontologies

Ontologies are an abstraction of a conceptualization containing explicitly defined concepts and associated relations (Gruber, 1993). As ontologies are seen as a means of sharing and reusing knowledge (Hitzler, Krotzsch, Ehrig, & Sure, 2005), consolidating the knowledge from multiple ontologies could potentially assist in enhancing the definition of terms semantically. Specific ontologies are created for specific purposes and would not be comprehensive in linking terms for general domains. By combining available ontologies, the pool of terms or resources would be widened, hence increasing the probability of equivalent terms being available and sufficiently described. Domain knowledge can be made explicit by using ontologies, but ontology development is largely done in silos, and there is no unified effort for a single ontology. These individual efforts each contain their own strengths and weaknesses, and merging these resources could prove beneficial in providing a rich base of knowledge of a particular domain. The semantic web will attain its goal only after every associated data source is presented within a common platform which is dependent on the efficient integration of Ontologies (Malik, Prakash, & Rizvi, 2010).

Merging Tools

Some semi-automatic tools were available over the last two decades to assist the user in the ontology merging process. Prompt is an ontology algorithm and a Protégé plugin tool for interactively merging and aligning Ontologies. The PROMPT tab provides a semi-automatic approach to ontology merging and alignment, and performs some tasks automatically and guides the user in performing other tasks where intervention is required (Gruber, 1993). Most of the well-known tools have been summarized in (Amrouch & Mostefai, 2012), and these include FCA-Merge, IF MAP, Chimaera, GLUE, CAIMAN, ONION, and ConcepTool. Each of these tools have adopted different techniques to semi automatically assist the knowledge engineer in the ontology merging and mapping process. A recent search for these tools found that the tools have not been updated or made available since 2003. The Prompt plugin was found to be no longer available for Protégé 5.0. Other merging tools such as OntoMerge ("OntoMerge," n.d.) and Chimaera ("Chimaera Home Page," n.d.; McGuinness, Fikes, Rice, & Wilder, 2000) appear to no longer be active projects since 2003; and while OntoMerge online service is available for very limited mapping, its tools are no longer available for download. OWL API (Matentzoglu & Palmisano, 2016) is a Java API and reference application for creating, manipulating and serializing OWL Ontologies, and has an active user base on GitHub as of July 2020. Protégé is the only tool that has continuously been updated to this date, hence this paper will utilize Protégé and OWL API, which are existing tools with active users and updates, and are also available for ontology merging.

Protégé

As stated in the previous section, recent ontology research utilizes the Protégé ontology tool, which has many advantages over other ontology tools as it is free, open source, allows generation and manipulation of ontologies, and ontologies built using Protégé can be accessed from Java programs through the Protégé-OWL API as well as other Java OWL API's. Protégé is one of the most widely used tools for ontology creation and merging (Negi & Malik, 2018), and the latest version 5.5.0 was used for this paper. This paper focuses on the 'merging' feature in Protégé which is exemplified with a case study of merging two existing and publicly available ontologies, namely, the Person Ontology which can be obtained from http://ebiquity.umbc.edu/ontology/person.owl and the Institutional Ontology which can be obtained from https://www.isibang.ac.in/~bisu/ontology/instOntology.owl. These ontologies were chosen based on their relative homogeneity, manageable size, and concepts which are similar, which provides an insight into how the merging process handles the union of concepts which are similar or equivalent.

Person and Institutional Ontology

The focus of the outcome of the merging will be on the Person – Student class in the Person Ontology, as shown in Figure 1, and the same Person – Student class in the Institutional Ontology, as shown in Figure 2. These are equivalent concepts of a person who is also a student, which exists in both ontologies. The student concept in each ontology further contains different subtypes in each of the ontologies.

Student class in Person Ontology is further classified into B.S Student, M.S. Student, MPS Student, Ph.D. Candidate, and Ph.D. Student, whereas the Student class in Institutional Ontology is further classified into Postgraduate Student, Research Student, and Undergraduate Student.

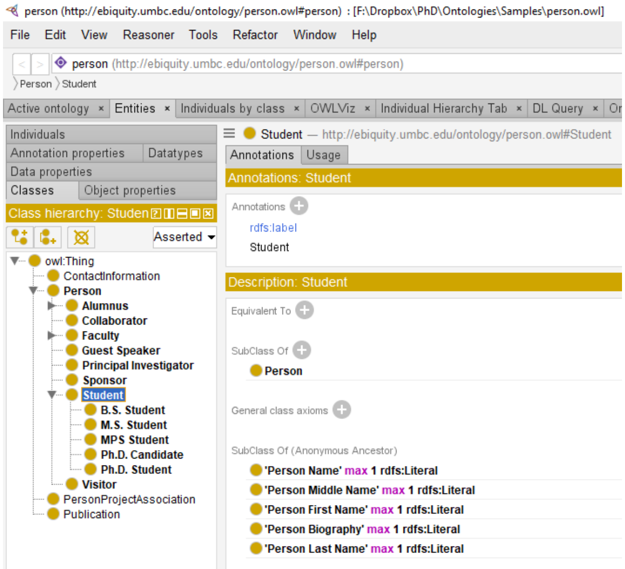

person (http://ebiquity.umbc.edu/ontology/person.owl#person) : [F:\Dropbox\PhD\Ontologies\Samples\person.owl]
File Edit View Reasoner Tools Refactor Window Help
person (http://ebiquity.umbc.edu/ontology/person.owl#person)
Person Student
Active ontology × Entities × Individuals by class × OWLViz × Individual Hierarchy Tab × DL Query × Or
Individuals
Annotation properties Datatypes
Data properties
Classes Object properties
Class hierarchy: Studen
Asserted
owl:Thing
ContactInformation
Person
Alumnus
Collaborator
Faculty
Guest Speaker
Principal Investigator
Sponsor
Student
B.S. Student
M.S. Student
MPS Student
Ph.D. Candidate
Ph.D. Student
Visitor
PersonProjectAssociation
Publication
Student — http://ebiquity.umbc.edu/ontology/person.owl#Student
Annotations Usage
Annotations: Student
Annotations
rdfs:label
Student
Description: Student
Equivalent To
SubClass Of
Person
General class axioms
SubClass Of (Anonymous Ancestor)
'Person Name' max 1 rdfs:Literal
'Person Middle Name' max 1 rdfs:Literal
'Person First Name' max 1 rdfs:Literal
'Person Biography' max 1 rdfs:Literal
'Person Last Name' max 1 rdfs:Literal

Figure 1: Person Ontology Viewed in Protégé Class Hierarchy

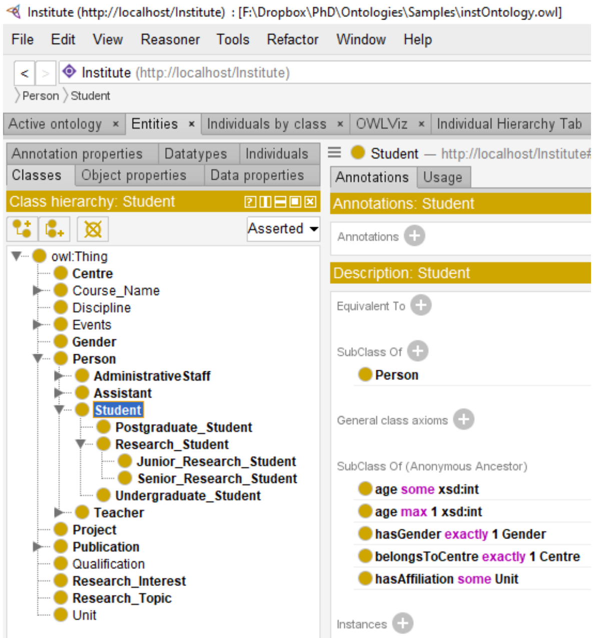
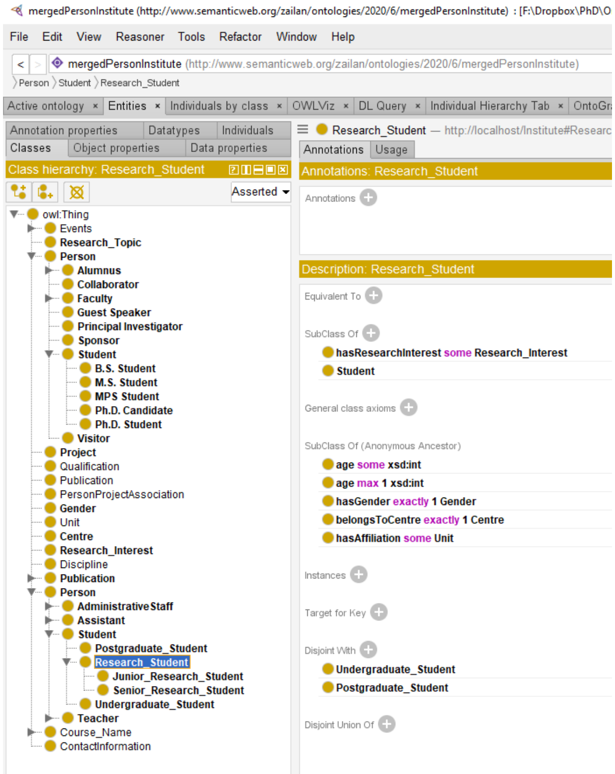


Figure 2: Institutional Ontology Viewed in Protégé Class Hierarchy

Protégé Merge

The Protégé Refactor – Merge Ontologies function was used to merge the two ontologies above, and the merged ontology was saved as a new ontology called mergedPersonInstitute.owl, and the outcome is shown in Figure 3.



SubClass Of
hasResearchInterest some Research_Interest
Student

SubClass Of (Anonymous Ancestor)
age some xsd:int
age max 1 xsd:int
hasGender exactly 1 Gender
belongsToCentre exactly 1 Centre
hasAffiliation some Unit

Disjoint With
Undergraduate_Student
Postgraduate_Student

Figure 3: Merged Ontology mergedPersonInstitute.owl Viewed in Protégé Class Hierarchy

Discussion

Merging is defined as 'to combine, blend, or unite gradually so as to blur the individuality or individual identity of ("Merge | Definition of Merge at Dictionary.Com," n.d.). In this sense, the Protégé merging tool does combine and unite the two separate ontologies but does not blur or remove the individuality of the original ontologies, as we will illustrate further.

We can see from the merged ontology in Figure 3 that the merging did combine the two ontologies into one but did not remove the individuality of the original ontologies as exemplified by the Person and Student classes. These classes are replicated and duplicated in the merged ontology. The Protégé merging tool only copied and replicated all classes from both ontologies into a new ontology, with many duplicated classes. The 'Merge ontologies' function in Protégé 5.5.0 does not actually merge or unify any of the classes, but only replicates all classes and properties of both the source ontologies. The visualization of the merged ontology using WebVOWL can be seen in Figure 4.

Ontology Visualisation

We will use WebVOWL (http://vowl.visualdataweb.org/webvowl.html) as it allows to export the complete or filtered VOWL visualization as an SVG image that can be opened in other programs, scaled without loss of quality, edited, shared, and printed (Janowicz, Lohmann, Negru, Haag, & Ertl, 2016). This is not possible in the Protégé VOWL plugin.

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 6 Issue 22 (September 2021) PP. 148-157
DOI: 10.35631/JISTM.622012

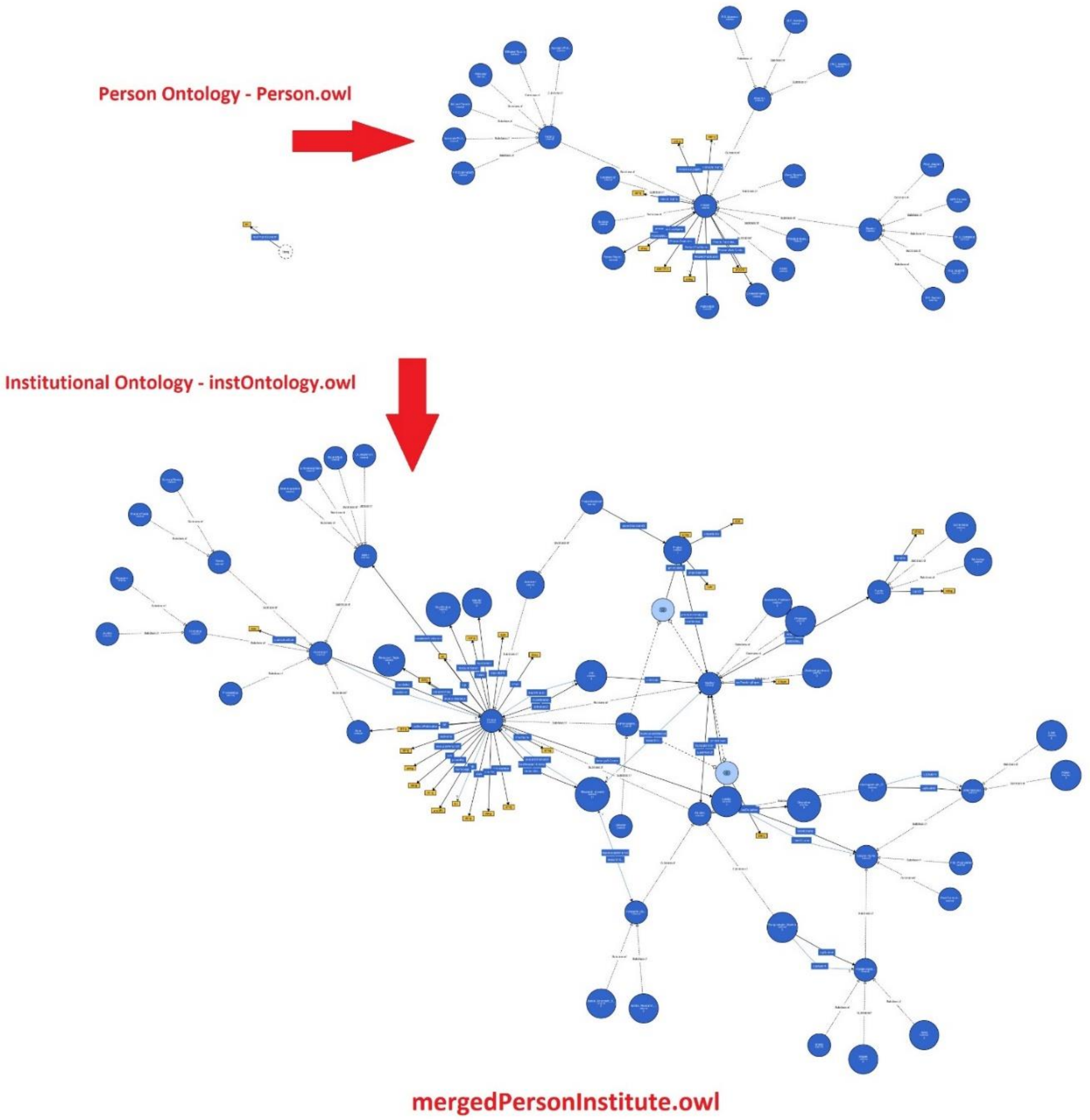

Person Ontology - Person.owl
Institutional Ontology - instOntology.owl
mergedPersonInstitute.owl

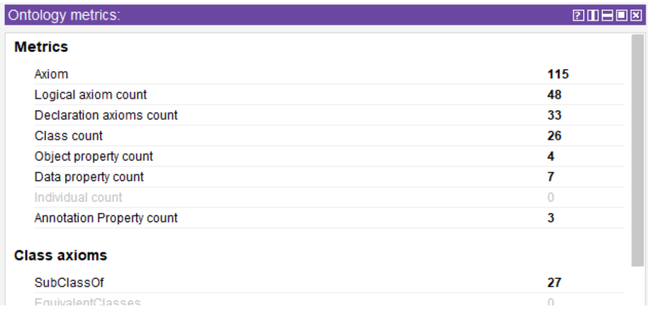
Figure 4: Merged Ontology mergedPersonInstitute.owl Viewed in WebVOWL

Observation

While the details of the classes cannot be clearly seen in Figure 4, the main purpose of this visualization is to note that the merged ontology has two distinct clusters of concepts, which are not connected to each other, even though they are in the single merged OWL file. As seen in Figure 3 the Person and Student classes exist as duplicated classes. Figure 4 clearly shows that the merged ontology does not 'merge' the ontology via any common classes or concepts that exist between the two ontologies. Rather, it unites the two ontologies into a single OWL file, and Protégé as an ontology editor then provides tools for the human editor to manually merge the classes.

Ontology Metrics

The ontology metrics view displays entity and axiom count for the axioms in the active ontology and its imports closure ("Ontology Metrics," n.d.). Figure 5 shows the ontology metrics for the Person Ontology.



Ontology metrics:

Metrics

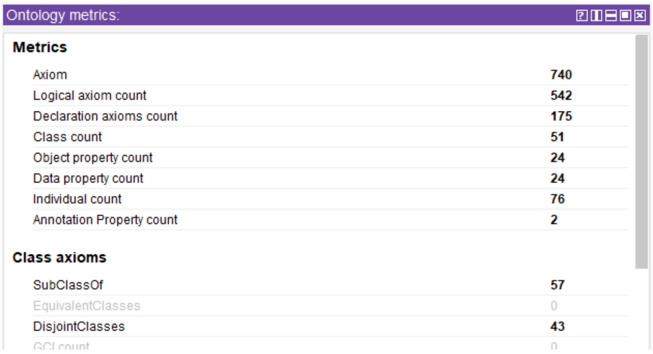
Axiom	115
Logical axiom count	48
Declaration axioms count	33
Class count	26
Object property count	4
Data property count	7
Individual count	0
Annotation Property count	3

Class axioms

SubClassOf	27
EquivalentClasses	0

Figure 5: Ontology Metrics for Person Ontology

Figure 6 shows the ontology metrics for the Institutional Ontology.



Ontology metrics:

Metrics

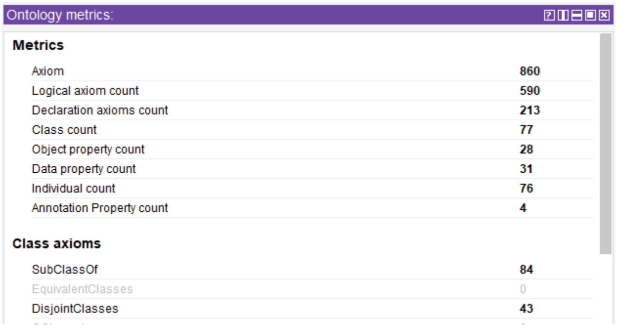
Axiom	740
Logical axiom count	542
Declaration axioms count	175
Class count	51
Object property count	24
Data property count	24
Individual count	76
Annotation Property count	2

Class axioms

SubClassOf	57
EquivalentClasses	0
DisjointClasses	43
GCI count	0

Figure 6: Ontology Metrics for Institutional Ontology

Figure 7 shows the Merged Ontology metrics (mergedPersonInstitute.owl)



Ontology metrics:

Metrics

Axiom	860
Logical axiom count	590
Declaration axioms count	213
Class count	77
Object property count	28
Data property count	31
Individual count	76
Annotation Property count	4

Class axioms

SubClassOf	84
EquivalentClasses	0
DisjointClasses	43

Figure 7: Ontology Metrics for Merged Person-Institute Ontology

Analysis of Ontology Metrics

Axioms are defined statements about entities. As we can see above, the number of logical axioms show that none of the axioms were merged, and the logical axiom count is as expected, 542+48 = 590.

The Protégé ontology metrics displays Axiom which is defined as the combined logical and non-logical axiom count. There should be 740 + 115 = 855 axioms, but there are 860 axioms in the merged ontology. The additional 5 axioms are declaration axioms, whose count is 33+175 = 208, but there are 213 in the merged ontology, 5 additional ones which are non-logical annotations created to record the new ontology Internationalized Resource Identifier (IRI), and other non-logical information associated with the new ontology.

Class count is as expected, 26+51 = 77.

Object property count is 4+24=28 and Data property count is 7+24=31 which is as expected. The ontology metrics confirm that no logical axioms, classes, or objects have been merged, and they have only been combined with no consideration of similarities, and no assisted mapping is provided.

The merging task above was also performed using Java OWL API (Matentzoglu & Palmisano, 2016), where the OWLOntologyMerger API was used to merge the Person and Institutional ontologies, and upon inspection it was found that the exact same merged ontology was produced as the Protégé merging tool.

Some semi-automatic tools were available in the past to assist the user in the ontology merging process. Prompt is an ontology algorithm and a Protégé plugin tool for interactively merging and aligning Ontologies. The PROMPT tab provided a semi-automatic approach to ontology merging and alignment, and performed some tasks automatically and guided the user in performing other tasks where intervention was required (Malik et al., 2010); however, Protégé plugin tools for semi-automatic merging such as PROMPT and SMART (Noy & Musen, 2003) are no longer available as plugins in the current version of Protégé.

Conclusion

Domain knowledge can be made explicit by using ontologies, but ontology development is largely done in silos, and there is no unified effort for a single ontology in most domains. These individual efforts each contain their own strengths and weaknesses and merging these resources could prove beneficial in providing a richer base of knowledge for any domain. Automated ontology merging tools appear to have taken a step back over the last 20 years, and Protégé now provides the Refactor menu with functions to manually edit the ontologies by copying/moving/deleting axioms. There is a need for automated or semi-automated tools to assist the user in the merging, alignment, and mapping of ontologies. Classes from different ontologies with the same spelling can be identified as potential candidates for unification, and classes with similar concepts can be presented to the user for further consideration. Merging of ontologies will benefit any domain which utilises formal standards to represent domain knowledge, as it will expand and combine cross domain knowledge that is machine readable, which will bring us a step closer to the realisation of the Semantic Web. Towards this there is a need to consolidate homogenous and heterogenous machine readable data across different domains automatically; and the opportunities for further research into automated ontology merging techniques and tools are substantial.

Acknowledgement
The authors would like to express gratitude to the National University of Malaysia (UKM) for providing the opportunity and funding under the Fundamental Research Grant Scheme code FRGS/1/2017/ICT04/UKM/02/8.

References

Amrouch, S., & Mostefai, S. (2012). Survey on the literature of ontology mapping, alignment and merging. *2012 International Conference on Information Technology and E-Services, ICITeS 2012*. https://doi.org/10.1109/ICITeS.2012.6216651

Cheatham, M., Cruz, I., Euzenat, J., & Pesquita, C. (2017). *Special issue on ontology and linked data matching Special Issue on Ontology and Linked Data Matching Editorial 1*. https://doi.org/10.3233/SW-160251ï

Chimaera Home Page. (n.d.). Retrieved September 22, 2020, from http://www.ksl.stanford.edu/software/chimaera/

Gruber, T. R. (1993). Technical Report KSL 92-71 Revised April 1993 A Translation Approach to Portable Ontology Specifications by A Translation Approach to Portable Ontology Specifications. *Knowledge Creation Diffusion Utilization*, *5*(April), 199–220. https://doi.org/http://dx.doi.org/10.1006/knac.1993.1008

Guarino, N. (1998). Formal Ontology and Information Systems. *Formal Ontology in Information Systems: Proceedings of the 1st International Conference*, *46*(June), 3–15. https://doi.org/10.1.1.29.1776

Hitzler, P., Krotzsch, M., Ehrig, M., & Sure, Y. (2005). What Is Ontology Merging? *Web*, 1–4. Retrieved from papers2://publication/uuid/28D629A7-A9B9-4EA8-8943-D1FEBDDB2CA5

Janowicz, K., Lohmann, S., Negru, S., Haag, F., & Ertl, T. (2016). Visualizing Ontologies with VOWL. *Semantic Web*, *7*(4), 399–419. https://doi.org/10.3233/SW-150200

Koprowska, M. (2008). *Remarks on Automated Ontology Merging Algorithm*. https://doi.org/10.1109/ICSEng.2008.58

Malik, S. K., Prakash, N., & Rizvi, S. A. M. (2010). Ontology merging using prompt plug-in of protégé in Semantic Web. *Proceedings - 2010 International Conference on Computational Intelligence and Communication Networks, CICN 2010*, 476–481. https://doi.org/10.1109/CICN.2010.151

Matentzoglu, N., & Palmisano, I. (2016). An Introduction to the OWL API. Retrieved September 22, 2020, from http://syllabus.cs.manchester.ac.uk/pgt/2020/COMP62342/introduction-owl-api-msc.pdf

McGuinness, D. L., Fikes, R., Rice, J., & Wilder, S. (2000). An Environment for Merging and Testing Large Ontologies. *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning KR2000*, 483–493. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.109.1812&amp;rep=rep1&amp;type=pdf

Merge | Definition of Merge at Dictionary.com. (n.d.). Retrieved September 22, 2020, from https://www.dictionary.com/browse/merge

Negi, S., & Malik, S. K. (2018). An Algorithm for Merging Two Ontologies: A Case Study. *International Journal of Applied Engineering Research*, *13*(12), 10327–10338. Retrieved from http://www.ripublication.com

Noy, N., & Musen, M. A. (2003). *The PROMPT suite : Interactive tools for ontology merging and mapping The PROMPT Suite : Interactive Tools For Ontology Merging And Mapping*. (April 2017). https://doi.org/10.1016/j.ijhcs.2003.08.002

Ontology Metrics. (n.d.). Retrieved September 22, 2020, from https://protegeproject.github.io/protege/views/ontology-metrics/

OntoMerge. (n.d.). Retrieved September 22, 2020, from http://www.cs.yale.edu/homes/dvm/daml/ontology-translation.html

Pawełoszek, I., & Korczak, J. (2018). Merging of ontologies – Conceptual design issues. *ACM International Conference Proceeding Series*, 59–63. https://doi.org/10.1145/3285957.3285987