

Semi-Automatic Merging of Ontologies using Protégé

Ayesha Ameen

Department of I.T,
Deccan College of Engineering
and Technology, Hyderabad

Khaleel Ur Rahman Khan

Dept. of C.S.E, ACE
Engineering College, JNTU,
Hyderabad

B.Padmaja Rani

Head of Department C.S.E,
JNTU, Hyderabad

ABSTRACT

Ontologies play a crucial role in bringing the Semantic Web vision to its full potential. Ontologies are developed at different levels of abstraction and by different people for various purposes. Knowledge represented by the ontologies is scattered because of existence of many ontologies representing the same concepts, therefore it becomes difficult to analyse, study and use the knowledge spread across many ontologies if they are studied individually. Knowledge represented by ontologies can be combined into a single ontology which is shown in this paper. Ontologies can be merged to combine the knowledge from different ontologies. In this paper we have shown the merger of two ontologies first ontology is university second ontology is the Student profile ontology containing details of Student which were developed in educational domain. Ontologies were developed and merged using protégé 4.0 alpha tools.

General Terms

Artificial Intelligence

Keywords

Ontology, Semantic Web, Protégé, Algorithm, Merger

1. INTRODUCTION

In the vision of Semantic Web ontologies play a very crucial role. Proper utilization of ontologies is very important for the Semantic Web vision. Much ontology exists in every domain at various level of abstraction representing the same concepts. This paper uses two ontologies developed in education domain. First ontology is university ontology as in [1]. University ontology contains the detail of working of university. There are several classes representing all important concepts in the university domain.

The second ontology used is Student profile ontology containing all type of information gathered from Student as in [2]. There are many advantages of using ontological Student profile. Additional user characteristics can be inferred by using ontology relations, conditions, restrictions. Ontological profile can be reused and shared.

The paper describes the ontologies which were developed earlier in [1, 2] in section III. In the following section shows the merger of ontological Student profile with University ontology. Merger combines the information from both the ontologies into a single ontology. Merger combines the information automatically using protégé 4.0 alpha tool. Automatic merging does not solve the inconsistencies created as a result of merged. Manual intervention is required to solve the inconsistencies of merged ontology. The last section of the paper checks the consistency of the merged ontology using a FACT++ reasoner.

2. RELATED WORK

Ontologies provide huge amount of information. To share information of ontology developed in same domain ontologies can be merged. New ontology is created as a result of merger.

Merged ontology is union of input ontologies and it contains all information of the input ontologies. Many approaches are used for merging as in [3] where an algorithm is used for merging ontologies by using hybrid strategy which performs matching, similarity check and use heuristics function. Several methods have been proposed for determining semantic similarity between concepts of ontology. They are divided in to four main categories [3]. Other approaches use semantic and syntactical matching as proposed in [4, 5]. PROMPT is a semi-automatic method for ontology merging in Protege-2000, it merges classes, slots.

Semi-automatic approach proposed in this paper is using the ontologies developed in protégé 4.0 alpha which were not used in any of the earlier work. Merged ontology generated using protégé is inconsistent because of violation of unique name assumption. The inconsistencies created by merging were resolved by analyzing the merged ontology.

3. INPUT ONTOLOGIES

The input ontologies for merging are University ontology and Student profile ontology. University ontology gives the complete information of university by using several classes that demonstrate the working of university. There are several steps that have to be followed in developing the University ontology as in [1]. Student profile created is static which requires information to be entered explicitly by student. Profile contains complete detail of student. The steps required to construct the profile are given in [2].

3.1 Classes

Class hierarchy consists of classes given in Figure 1. University class is the root of the ontology which is further divided into several classes that demonstrate the working of the university. Student profile hierarchy is given in Figure 2.

3.2 Properties

Properties represent relationships. Three types of properties are defined in owl. Annotation properties are used to add information to classes, individuals and object/data type properties. Annotation properties for University ontology and Student profile ontology are shown in Figure 3 and Figure 4 respectively. Data type properties links individuals to an XML schema data type or an RDF literal. Data type properties for University ontology and Student profile ontology are shown in Figure 5 and Figure 6 respectively. Object properties relate one instance of a domain class to instance of range class. Object Properties of University ontology and Student profile are given in Figure 7 and Figure 8 respectively.

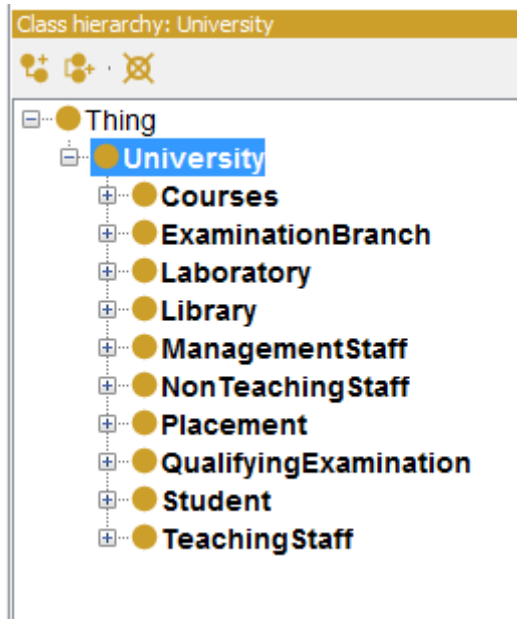


Fig 1: University ontology subclasses

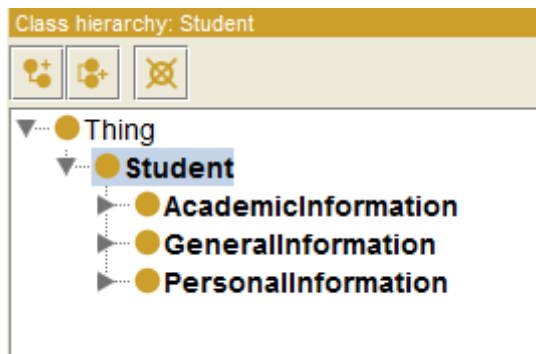


Fig 2: Student profile class hierarchy

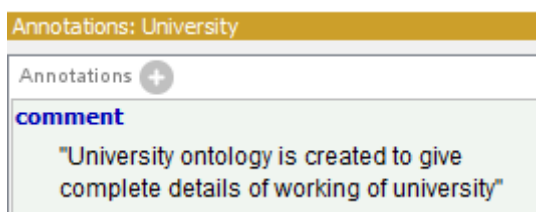


Fig 3: Annotation properties of University ontologies

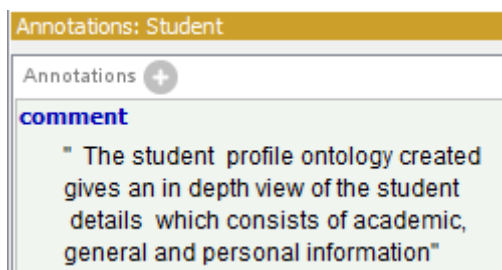


Fig 4: Annotation properties of Student profile

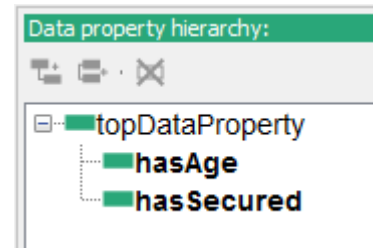


Fig 5: Data type properties of University ontologies.

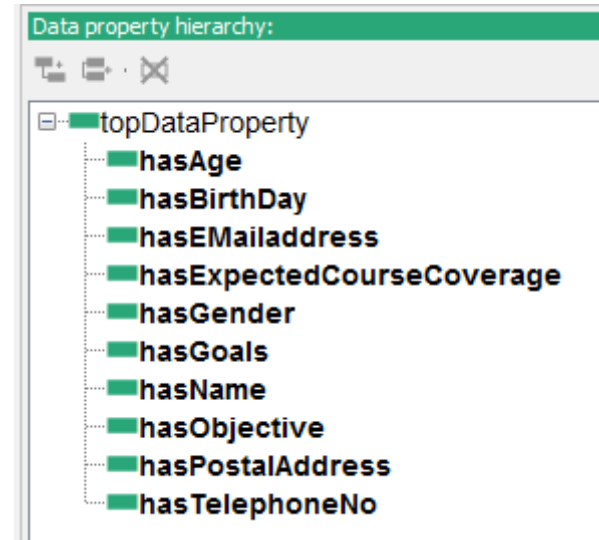


Fig 6: Data type properties of Student profile

3.3 Restrictions

Restrictions describe class of individuals based on the relationship that members of the class participate in [6]. Restriction is a class that contain all individuals that satisfy the restrictions. Properties are used to define restrictions on the classes. Restrictions can be categorised as

3.3.1 Quantifier Restrictions

Quantifier restrictions can be further divided into existential and universal restrictions. Existential restrictions are used to describe a class of individuals that participate in at least one relationship along a specified property to individuals that are members of specified class [6].

Existential restriction for ArchitecturePG class is shown in Figure 9. Student seeking admission in ArchitecturePG must have passed some architecture under graduation course. Universal restrictions are used to describe classes of individuals that for a given property only have relationship along this property to individuals that are members of a specified class [6]. Universal restrictions for law under graduation course are shown in Figure 10. Student seeking admission in law under graduation course must have passed lawcet which is the entrance examination conducted by University.

Existential restriction for Student profile is shown in Figure 11 for the class educational background i.e. a Student can have an educational background if he has passed some last passed examination. Universal restriction for Student profile is shown in Figure 12 for the class qualifying examination passed i.e. a Student can be a member of qualifying examination passed only if he has passed some qualifying examination.

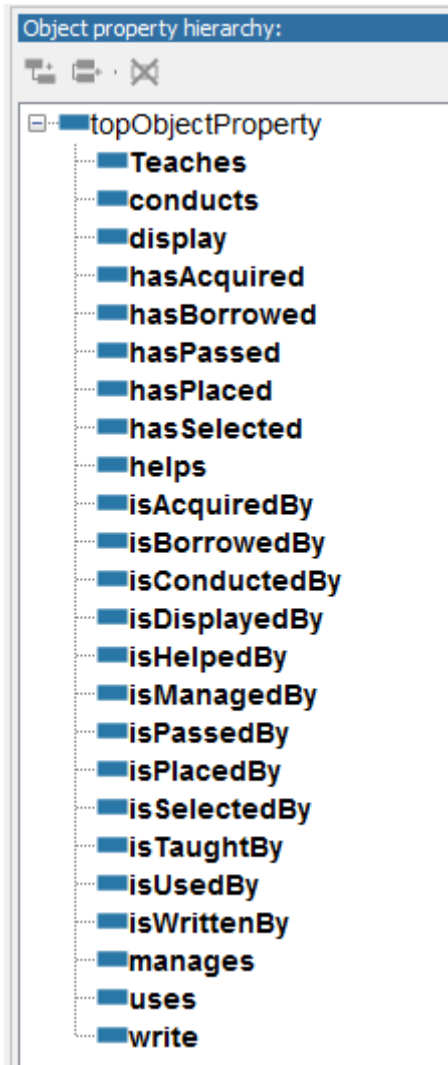


Fig 7.Object properties of University ontologies

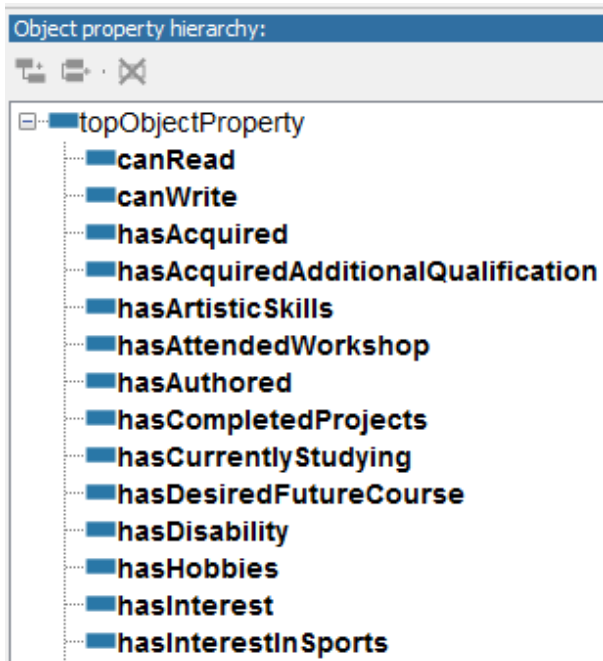


Fig 8.Object properties of Student profile



Fig 9 .Existential restrictions for ArchitecturePG class

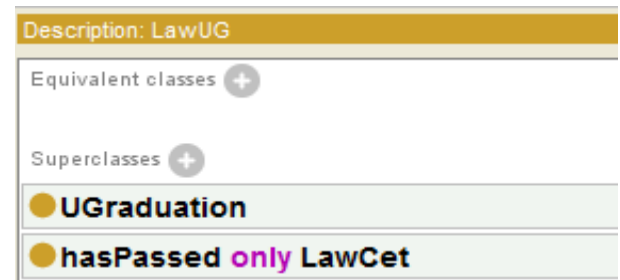


Fig 10.Universal restriction for LawUG class

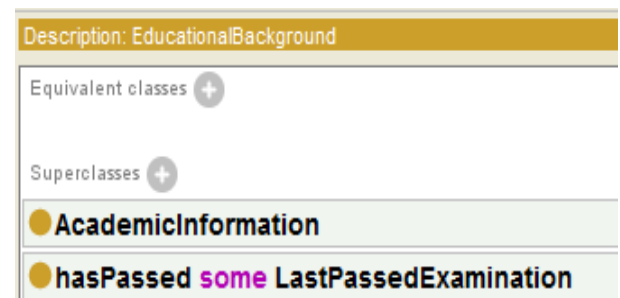


Fig 11 .Existential restrictions for EducationalBackground class

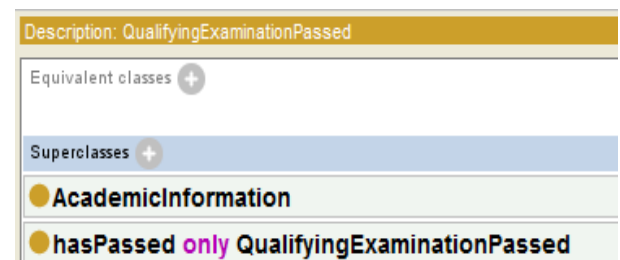


Fig 12.Universal restriction for QualifyingExaminationPassed class

3.1.2 Cardinality restrictions

Cardinality restrictions are used to specify class of individuals that have, at least, at most or exactly a specified number of relationships with other individuals. In University ontology the cardinality restriction can be used to specify that a Student seeking admission in post-graduation arts course must have passed at least one under graduation arts course as shown in Figure 13. Cardinality restriction for Student profile is given for the class CurrentPursuingCourse i.e. a Student must currently pursue exactly one current pursuing course shown in Figure 14.

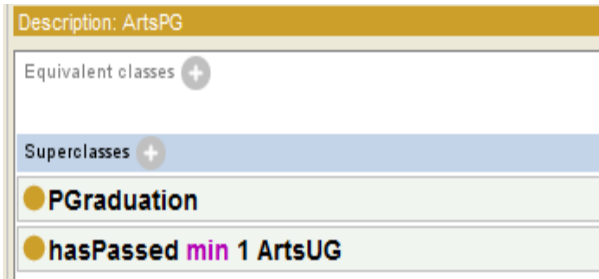


Fig 13. Cardinality restriction for ArtsPG class

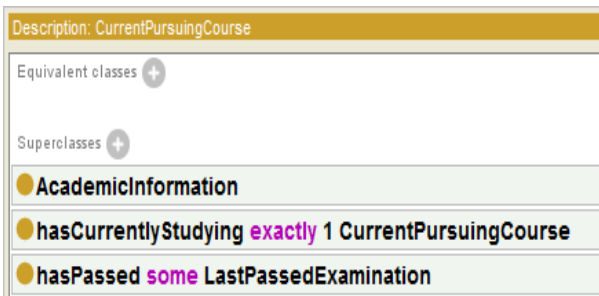


Fig 14. Cardinality restriction for CurrentPursuingCourse class

3.1.3 hasValue Restrictions

hasValue restriction can be used to specify an instance of class. An instance of Student can be created as Jack and Sam as shown in Figure 15. Names of different universities can be added as instances in University ontology as shown in Figure 16.

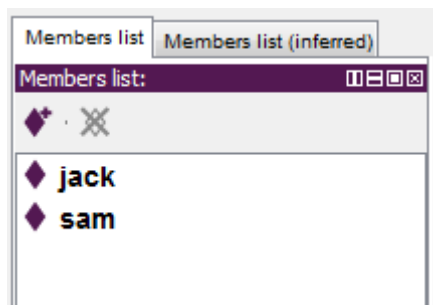


Fig 15. hasvalue restriction for Student class

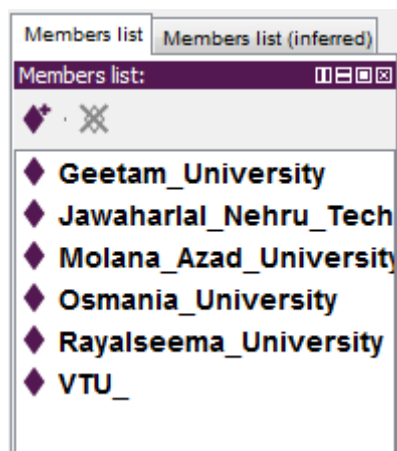


Fig 16. hasvalue restriction for University class

4. MERGING APPROACH

Merging approach starts with the analysis of the two ontologies that have to be merged. The ontologies to be merged have to be studied in detail carefully to analyse the merge point. This section starts with the analysis of the ontologies then gives the steps involved in merging of ontologies. The steps to be followed for merging are given in the form of Algorithm given in Table 1 as follows. Graphical representation of merging is shown in Figure 17.

Table 1. Algorithm for merging

Input: University ontology, Student profile ontology.
Output: Updated University ontology with Student profile.
1. Analyze the ontologies to be merged.
2. Check for similarities.
3. Use the merge option in protégé 4.0 alpha for performing merge.
4. Sort inconsistencies created after merge.
5. Check the consistency of the merged ontology using a reasoner.

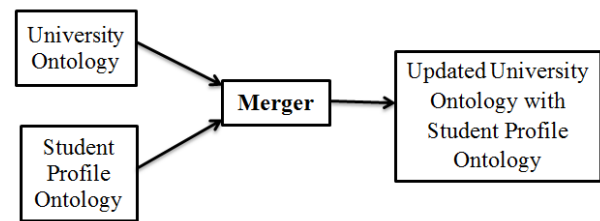


Fig 17. Graphical representation for merging

4.1 Analysis of ontologies

Ontologies to be merged are developed in same education domain therefore there exist similarity between the concepts they represents. Similarities are obtained by analysing the classes, properties and restrictions.

4.2 Similarities

4.2.1 Classes

Student class is repeated in both ontologies. Student class in Student profile has the subclasses shown in Figure 18. Student class in University ontology has subclasses shown in Figure 19.

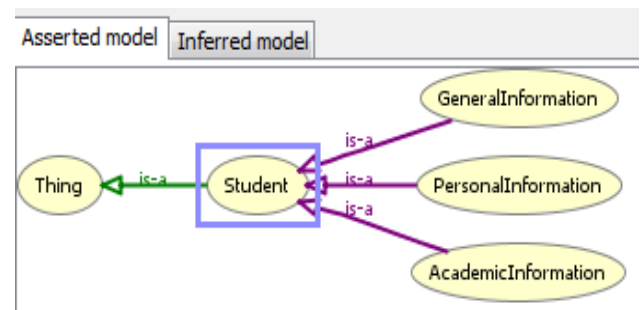


Fig 18. Owl Viz view of Student class of Student profile ontology



Fig 19. Owl Viz view of Student subclass of University ontology

4.2.2 Properties

hasPassed property is common between University ontology and Student profile. hasPassed in University ontology represents the relationship between Student class and QualifyingExamination class as shown in Figure 20. hasPassed in Student profile represents the relationship between Student class and EducationalBackground class as shown in Figure 21.

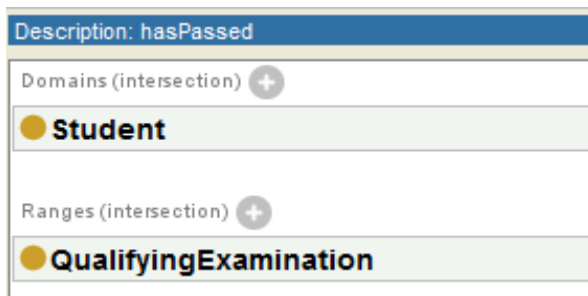


Fig 20. hasPassed object property in University ontology

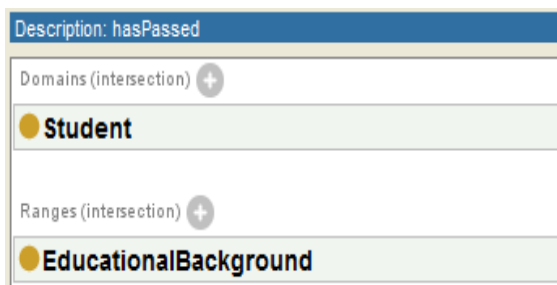


Fig 21. hasPassed object property in Student profile

4.3 Merge option

After finding similarities across ontologies. Ontologies are merged using the merge option in protégé 4.0 alpha the steps to be followed for merging are given below.

4.3.1 Open the ontology

The ontologies to be merged must be opened in the current window. Student profile ontology was first opened later University ontology was opened in the same window.

4.3.2 Select the merge ontology option from refactor menu.

Next step is to select the merge ontology option from refactor.

4.3.3 Select ontologies to be merged

The ontologies to be merged must be selected in this step as shown in Figure 22. Ontology1328692153093 is Student profile ontology and Ontology1339825779701 is University ontology.

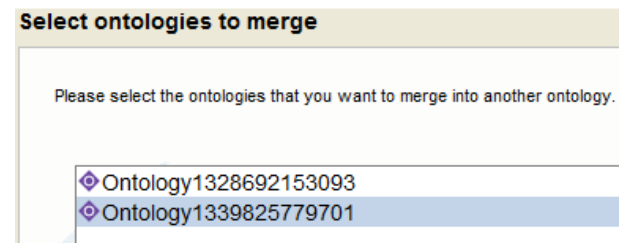


Fig 22. Selection of ontologies for merger

4.3.4 Select the merge type

In this step the merge type is selected. There are two options for merging. First option merges the ontologies into a new ontology. Second option merges the ontologies in to existing ontologies. In this paper we have chosen the second option for merging into existing ontology as shown in Figure 23.

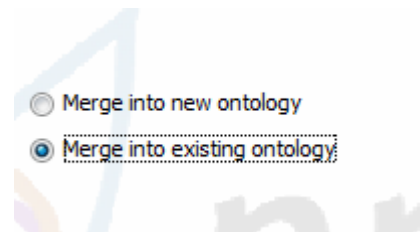


Fig 23. Selecting the merge type

4.3.5 Select the target ontology for merger

After selecting the second option for merging the target ontology for the merger must be chosen. University ontology is selected as the target ontology as shown in Figure 24.

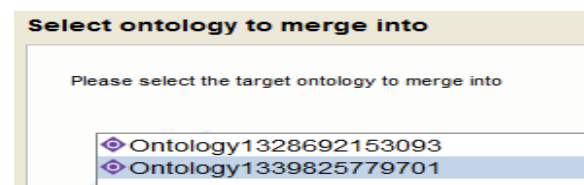


Fig 24. Selecting the target ontology for merger

4.3.6 . Result after merger

Merge option performs union of all classes, properties. Resultant ontology after merging is shown in Figure 25.

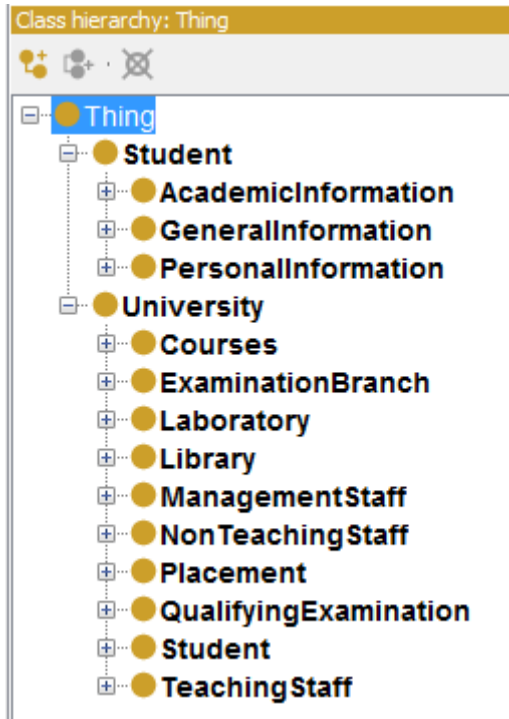


Fig 25.Merged ontology

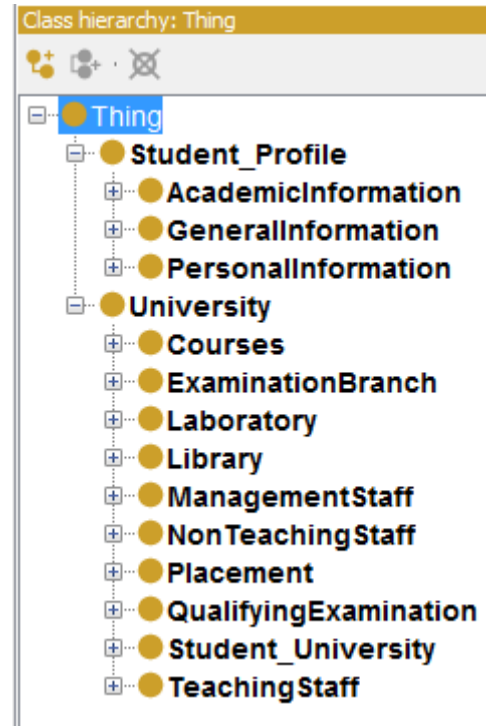


Fig 27.Student class renamed

4.4 Sorting inconsistencies

After merging the two ontologies, resultant ontology must be analysed to sort the inconsistencies created. Merged ontology is created by performing the union of two ontologies. Student class is contained in both the ontologies unique name assumption of owl is violated and if we run a reasoner we will get dialog box as reasoner is in progress but the reasoning will not complete and reasoner will hang because the input ontology is inconsistent as shown in Figure 26.

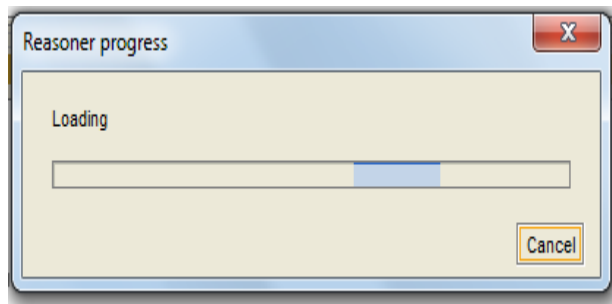


Fig 26 Reasoner progress window

To sort the inconsistencies merged ontology must be analysed. After analysis few inconsistencies were discovered which are removed by following steps mentioned below in order.

4.4.1 Change the name of Student class in

University ontology and Student profile ontology.

Student class in University ontology is renamed as Student_University and the Student class in Student profile ontology is renamed as Student_Profile as shown in Figure 27.

4.4.2 hasPassed, hasAcquired object property

hasPassed property is common in both ontology therefore the name of hasPassed property in University ontology is changed as hasPassedU and include courses in the range of hasPassedU since our design allows a Student to select a course only if he had passed previous course which is the minimum requirement for selecting a particular course as shown in Figure 28. In Student ontology hasPassed is renamed as hasPassedSP as shown in Figure 29.

hasAcquired object property is used in both the ontologies but the domain and range of hasAcquired is different in both ontologies, hasAcquired in Student profile relates instances of Student class to instances of EducationalBackground class. hasAcquired in University ontology relates instances in Student class to Qualification class.

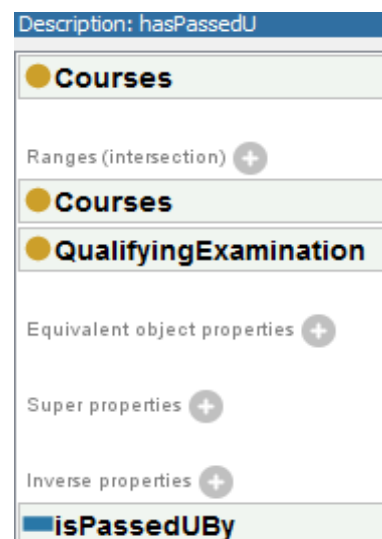


Fig 28.hasPassed property in University ontology

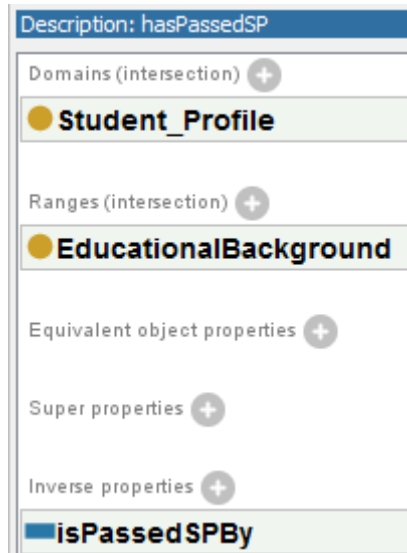


Fig 29. hasPassed property in Student profile Ontology

hasAcquired property is merged by taking union of the range of the property as shown in Figure 30.

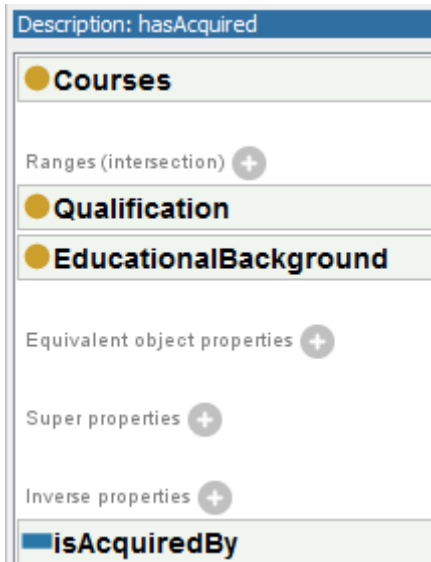


Fig 30 hasAcquired object property

4.4.3 Changing Restrictions

In University ontology existential restrictions are used to specify the relationship among classes. Using existential restrictions in merged ontology was giving inconsistencies because existential restrictions describe that individual participate in at least one relationship along a specified property to individuals that are members of a specified class[6] that was the bad design which was changed by replacing existential restrictions by universal restrictions which describe classes of individuals that for a given property only have relationships along this property to individuals that are members of a specified class as shown in the Figure 31.

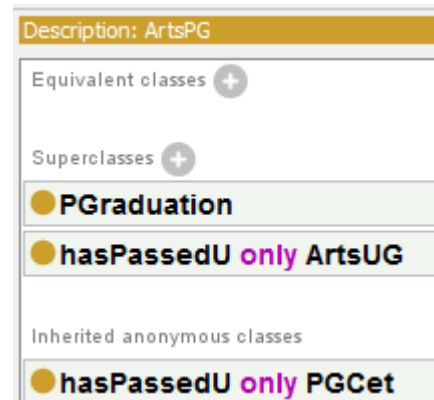


Fig 31. Change of existential restrictions to universal restriction

4.4.4 Name, Domain

Name, Domain is reserved words and must be removed from range of hasName property. Domain was used as a subclass of ProjectsCompleted class so it has to be removed.

4.5 Check the consistency of the ontology

To check the consistency of merged ontology a reasoner is selected and run. After running a reasoner the inferred class hierarchy is obtained as shown in Figure 32. Merged ontology is consistent if we get same inferred and asserted model as shown in Figure 33.

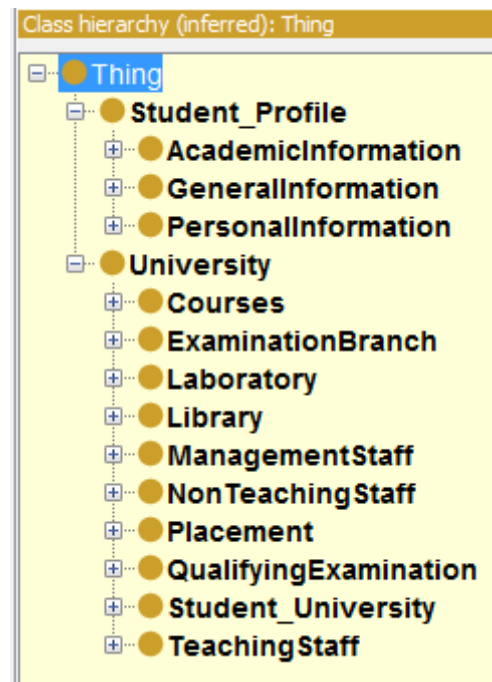


Fig 32. Inferred class hierarchy after reasoning

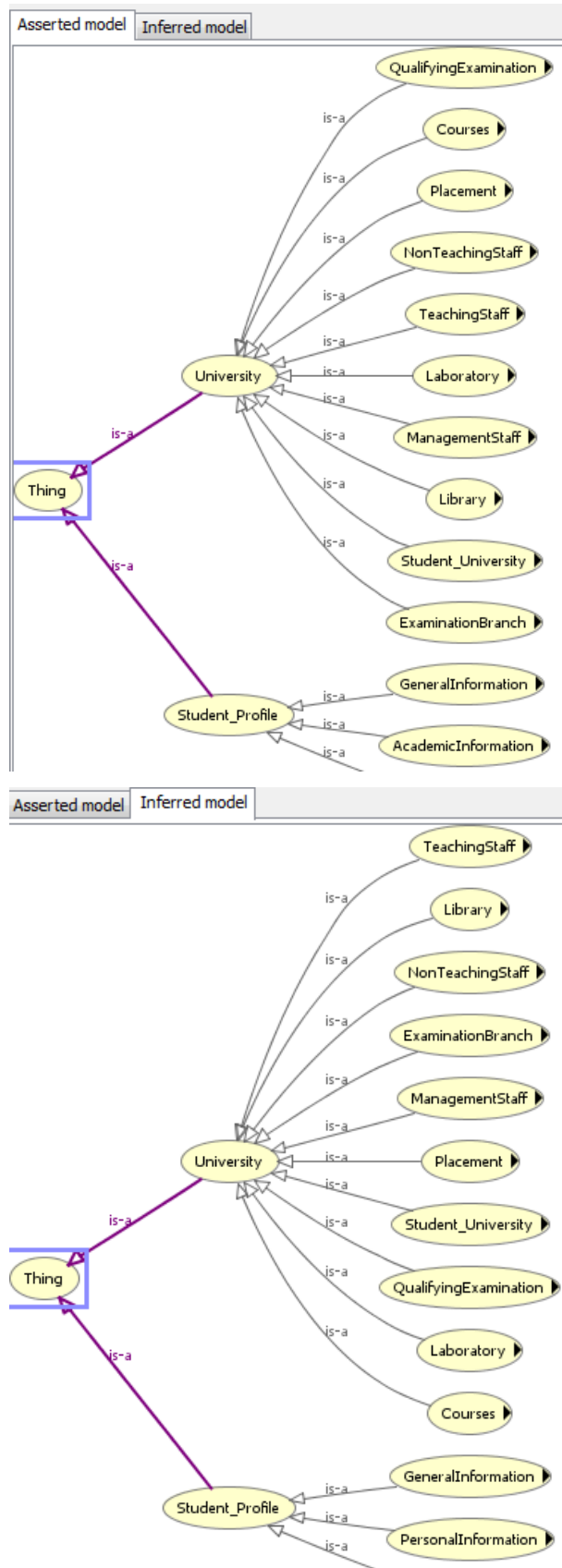


Fig 33.Asserted and Inferred model of merged ontology.

5. CONCLUSION

This paper demonstrates semi-automatic approach for merging ontology. Input ontologies are University ontology and Student profile ontology. The approach presented can be extended for more than two ontologies. Result generated by using merge option in Protégé is inconsistent because merge option performs union of input ontology to generate single output ontology. Human intervention is required to sort the inconsistencies generated after merge. Steps were taken to resolve the inconsistency generated after merge. Merged ontology was successfully tested using a reasoner.

6. REFERENCES

- [1] Ameen, Ayesha, KhaleelUr Rahman Khan, and B. Padmaia Rani. "Construction of University ontology." Information and Communication Technologies (WICT), 2012 World Congress on. IEEE, 2012.
- [2] Ameen, Ayesha, Khaleel Ur Rahman Khan, and B. Padmaja Rani. "Ontological Student profile." Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology. ACM, 2012.
- [3] 3.Robin, CR Rene, and G. Uma. "A novel algorithm for fully automated ontology merging using hybrid strategy." European Journal of Scientific Research 47.1 (2010): 074-081.
- [4] McGuinness, Deborah L., et al. "The chimaera ontology environment." AAAI/IAAI 2000 (2000): 1123-1124.
- [5] Gangemi, Aldo, Domenico M. Pisanelli, and Geri Steve. "An overview of the ONIONS project: Applying ontologies to the integration of medical terminologies." Data & Knowledge Engineering 31.2 (1999): 183-220.
- [6] Antoniou, Grigoris, and Frank Van Harmelen,2004, Semantic Web Primer. the MIT Press.