Ontology Merging and Matching Using Ontology Abstract Machine

Gopinath Ganapathy[1] , Ravi Lourdusamy[2]

[1]*Department of Computer Science,*
Bharathidasan University, Trichy, INDIA
gganapathy@gmail.com

[2]*Department of Computer Science,*
Sacred Heart College, Tirupattur, INDIA
raviatshc@yahoo.com

ABSTRACT

Ontology mediation is enabled through interoperability of semantic data sources. It helps data sharing between heterogeneous knowledgebase and reuse by semantic applications. Ontology mediation includes operations such as, mapping, alignment, matching, merging and integration. This paper aims at discussing a new approach of ontology merging and matching using ontology abstract machine with an illustration from a health care domain.

Keywords
Ontology Mediation, Ontology Abstract Machine, Ontology Operations

1.0 INTRODUCTION

In any semantic solution, data is annotated using ontologies. Ontologies are shared specifications and therefore the same ontologies can be used for the annotation of multiple data sources, like web pages, XML documents, relational databases, and so on. Their shared terminologies enable a certain degree of interoperability between the data sources using the same ontologies. To enable such an interoperation, mediation is required between the ontologies. Ontology mediation includes operations such as, ontology mapping, ontology alignment, ontology matching, ontology merging and ontology integration.

The aim of this article is to give readers a comprehensive overview of a new approach on ontology merging and matching using Ontology Abstract Machine(OAM). The article is organized as follows: Section 1 deals with an introduction. The terminologies involved are outlined in Section 2. In Section 3, the related works are discussed to understand the background of the proposed research work. Ontology merging and ontology matching operations are described using the OAM in Section 4 and in Section 5, conclusion is presented.

2.0 TERMINOLOGIES

2.1 Ontology Mediation

2.1.1 Ontology Mapping

An ontology mapping M is a declarative specification of the semantic overlap between two ontologies O_S and O_T. The correspondences between different entities of the two ontologies are typically expressed using some axioms formulated in a specific mapping language. Mapping can be unidirectional or bi-directional. The different phases in the generic mapping process as in (Livia et al., 2006) is depicted in Figure 1.

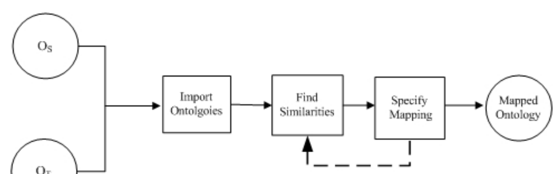

Figure. 1 Mapping Process

Import of Ontologies: Ontologies can be specified in different languages, which indicate a need to convert them to a common format so that the mapping can be specified. Furthermore, the ontologies need to be imported in the tool, which is used to specify the mapping. Finding Similarities: Many systems use the match operator to automatically find similarities between ontologies. For any two-source ontology, the match operator returns the similarities between ontologies. Specifying Mapping: After similarities between ontologies have been found, the mapping between the ontologies needs to be specified.

2.1.2 Ontology Alignment

The automated or semi-automated discovery of correspondences between two ontologies is called *ontology alignment*. Ontology alignment is the task of creating links between two original ontologies. Ontology alignment is made, if the sources found to be consistent with each other, but are kept separate or when sources are from the complementary domains.

2.1.3 Ontology Matching

Ontology matching is the process of discovering similarities between two sources of Ontologies. The result of matching operation is a specification of similarities between two ontologies. Ontology matching is carried out through the application of match operator (Erhard and Philip, 2001).

2.1.4 Ontology Merging

In ontology merging, a new ontology is created which is the union of source ontologies in order to capture all the knowledge from the original ontologies. There are two different approaches in ontology merging. In the first approach, the input of the merging process is a collection of ontologies and the outcome is, one new merged ontology which captures the original ontologies, as given in Figure 2.

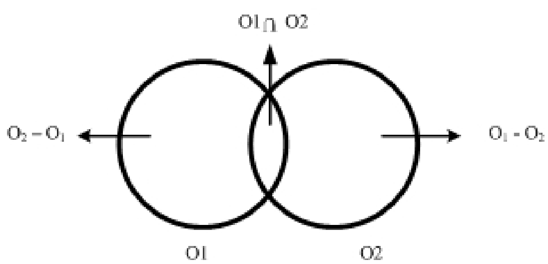

Figure. 2 Output of Merging Process (Approach 1)

In the second approach the original ontologies are not replaced, but rather a view called bridge ontology is created which imports the original ontologies and specifies the correspondence using bridge axioms as in Figure 3.

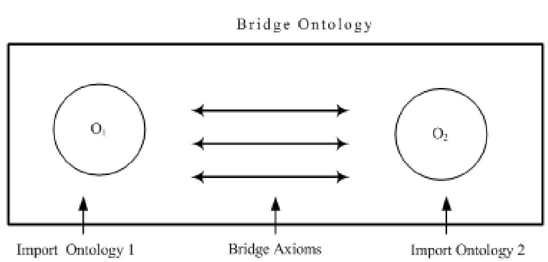

Figure. 3 Output of Merging Process (Approach 2)

2.1.5 Ontology Integration

Ontology integration is the process of generating a single ontology in one subject from two or more existing and different ontologies in different subjects. The different subjects of the different ontologies may be related. Some change is expected in a single integrated ontology (Namyoun, Yeol and Hyoil, 2006).

2.2 Ontology and Automata

2.2.1 Ontology Abstract Machine(OAM) (Leang, Jennifer, Julia and Alton, 2009)

The Ontology Abstract Machine model is defined in a format similar to that used for a finite state automaton. OAM is a 5-tuple representation of an ontology and is denoted by $M = (Q, \sum, \partial, Q_0, F)$. Q is a set of nodes and $Q = Q_c \cup Q_i \cup Q_v$. Q_c denotes a set of classes, Q_i denotes a set of instances, and Q_v denotes a set of values. $\sum$ is a set of relationships types and $\sum = \sum_B \cup \sum_E$. $\sum_B$ represents a set of relationship types and $\sum_E$ represents a set of extended relationship types. Q_0 is a set of source nodes. These are nodes with no incoming $\sum_B$ edge. This set can be identified from ∂ . Q_0 is a subset of $(Q_c \cup Q_i)$. Source nodes can only be elements of the set of classes or elements of the set of instances.F is a set of root nodes. These are nodes with no outgoing $\sum_B$ edge. F is a subset of Q_c. ∂ is a set of relationships in the form of edge(node, relationshiptype,node), $Q \times \sum \rightarrow Q$, hence each element is a child node, a relationship type, or a parent node.

3. 0 REVIEW OF LITERATURE AND RELATED WORKS

3.1 Ontology Mediation

The OntoMorph (Hans, 2000) system aims to facilitate ontology merging and the rapid generation of knowledge base translators. It combines two powerful mechanisms namely syntactic rewriting and semantic rewriting to describe KB transformations.

OBSERVER (Ontology Based System Enhanced with Relationships for Vocabulary hEterogeneity Resolution) (Mena, Illarramendi, Kashyap and Sheth, 2000) is a system, which is intended to overcome problems with heterogeneity between distributed data repositories by using component ontologies and the explicit relationships between these components. OBSERVER uses a component-based approach to ontology mapping.

ONION (ONtology compositION) (Prasenjit, Gio and Jan, 1999), (Prasenjit and Gio, 2001) is an architecture based on a sound formalism to support a scalable framework for ontology integration that uses a graph-oriented model for the representation of ontologies. The special feature of this system is that it separates the logical inference engine from the representation model of the ontologies. This allows for the accommodation of different inference engines in the architecture.

The process of FCA Merge (Stumme and Maedche, 2001) tool consists of three steps, namely (i) instances extraction and computing of two formal contexts K_1

and K_2, (ii) the FCA Merge core algorithm that derives a common context and computes a concept lattice and (iii) the generation of the final merged ontology based on the concept lattice.

GLUE (Doan, Madhavan, Domingos and Halevy, 2002) is a system, which employs machine-learning technologies to semi-automatically create mappings between heterogeneous ontologies, where an ontology is seen as taxonomy of concepts. GLUE focuses on finding 1-to-1 mappings between concepts in taxonomies.

MAFRA(MApping FRAmework for distributed ontologies) (Alexander, Boris, Nuno and Raphael, 2002) is a framework defined for mapping distributed ontologies on the semantic web. MAFRA has been implemented as a plug-in of KAON (Ontology management tool) and introduces two new concepts namely, semantic bridges and service-centric approaches.

The PROMPT (Noy and Musen, 2003) suite contains a set of tools that has had an important impact in the area of merging, aligning and versioning of ontologies. The different tasks in multiple ontology management are closely interrelated and share several components and heuristics. Thus, tools for supporting some of the tasks in the context of multiple ontology management can benefit greatly from their integration with others. The key components of the PROMPT suite have been developed as extensions (plug-ins) of the Protégé 2000 ontology development environment.

In (Peter, Bruce, and Ken, 2003) the authors have presented a matcher designed to handle mismatches between representations. Mismatches are addressed by using transformation rules whose antecedent and consequent are alternative encodings of the same information. These transformations along with the algorithm for applying them were described in detail and the proposed method was applied to the task of critiquing military Courses of Actions. The proposed method's performance on this task was compared with other match algorithms. The results showed that proposed matcher performed significantly better because of its use of transformations.

In (Dan, 2003), the issue of using some aspects of the Conceptual Graph Theory formalism to define functions on ontologies is explored. It demonstrates that projection as defined in Conceptual Graph Theory, can then be used for comparing and merging ontologies.

The QOM(Quick Ontology Mapping) tool (March and Steffen, 2004) represents an approach that considers both the quality of mapping results as well as the run-time complexity. The hypothesis is that mapping algorithms may be streamlined such that the loss of quality is marginal, but the improvement of efficiency is so tremendous that it allows for the ad-hoc mapping of large-size, light weight ontologies

Two important improvements to facilitate knowledge interchange are proposed in (Alma and Adolfo, 2006) namely the OM (Ontology Merging) Notation that provides substantial improvements to these languages and the OM Algorithm, which is totally automatic in comparison with others (Prompt, Chimaera, OntoMerge, FCA-Merge, IF-Map and ISI) where the user manually solves the most important problems found in the merging.

Ontology matching produces correspondences between elements of two ontologies and it is a basic problem in many database application domains. A Description Logics-based ontology matching approach is proposed in (Guohua, Zhiqiu, Xiaodong, Lei and Gaoyou, 2007). The approach achieves the matching by means of semantic interpretation. The algorithm of deciding ontology correspondences (such as equivalence, more general, less general, disjointness, overlapping) is presented, which is reasoned based on concept inclusion. The reasoner RacerPro is selected to deduce ontology mapping, and the experimental results demonstrate that the approach is feasible.

The methods of merging and aligning OWL-coded ontologies are provided in (Monika, 2008).

In (Pavel and Jerome, 2008) the authors aim at analyzing the key trends and challenges of the ontology matching field. The main motivation behind this work is the fact that despite many component matching solutions that have been developed so far, there is no integrated solution that is a clear success, which is robust enough to be the basis for future development, and which is usable by non expert users. This research work provides the basics of ontology matching and presents the general trends of the field and discusses ten challenges for ontology matching, thereby aiming to direct research into the critical path and to facilitate the progress of the field.

3.2 Ontology and Automata

In (Leang, Jennifer, Julia and Alton, 2009) the authors have introduced the OAM model and the related algorithms that enable maintenance of an ontology that supports node-based access.

In (Sergey, Lyudmila, Yelena and Olga, 2008) the summary of automata theory ontology is presented.

4.0 ONTOLOGY MERGING AND ONTOLOGY MATCHING

4.1 Ontology Merging

Considering the two ontologies of O_1 (Figure 4) and O_2 (Figure 5) representing two different hospitals as examples, the merging process is discussed in this section.

Each ontology is represented as OAM to facilitate the merging process. The two ontologies are represented as abstract machines defined as M_1 and M_2 respectively. Then the abstract machines M_1 and M_2 are combined to get the union of them which is termed as M_3. From the resultant abstract machine M_3 the merged ontology is obtained as shown in Figure 6.

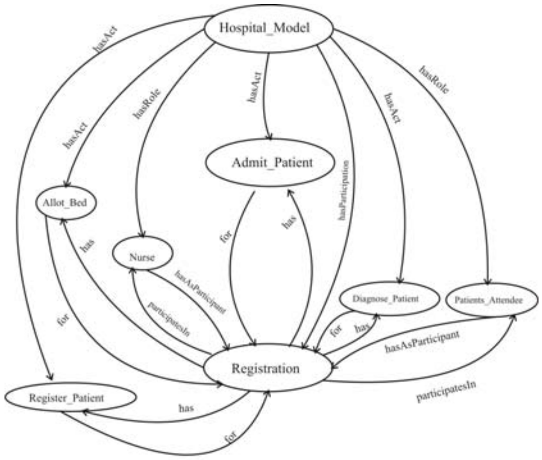

Figure. 4 Ontology of Hospital Model 1(O1)

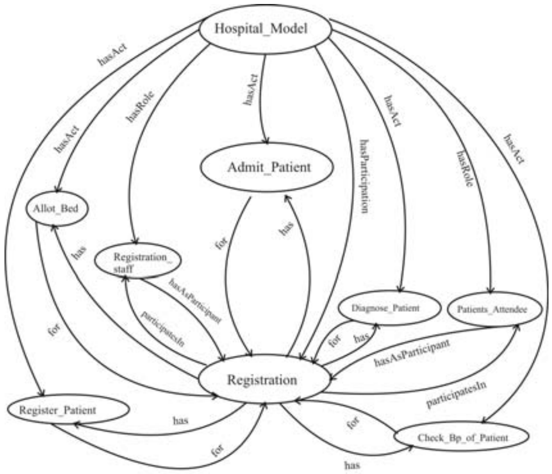

Figure.5 Ontology of Hospital Model 2 (O2)

Ontology Abstract Machine of Hospital Model 1:

$M_1 = (Q_1 , \sum_1 , \partial_1 , Q_{01} , F_1)$

Q_1 = { Hospital_Model , Allot_Bed , Register_patient , Admit_patient , Registration , Diagnose_patient , Patients_attendee , Nurse }

$\sum_1$ = { has Act , hasParticipation , hasRole , for , has , hasAsParticipant , participatesIn }

Q_{01} = { Hospital_Model }

F_1 = { }

∂_1 = { (Hospital_Model , hasAct , Allot_Bed) , (Hospital_Model , hasAct , Register_patient), (Hospital_Model , hasAct , Admit_patient) , (Hospital_Model , hasParticipation , Registration), (Hospital_Model , hasAct , Diagnose_patient) , (Hospital_Model , has Role , Patients_attendee) (Admit_patient , for , Registration) , (Registration , has , Admit_patient), (Registration , has , Allot_Bed) , (Allot_Bed , for , Registration), (Register_patient , for , Registration), (Registration , has , Register_patient), (Registration , has , Diagnose_patient) , (Diagnose_patient , for , Registration), (Patients_attendee , hasAsParticipant , Registration), (Registration , participatesIn , Patients_attendee), (Hospital_Model , hasRole , Nurse) , (Nurse , hasAsParticipant , Registration), (Registration , participatesIn , Nurse) }

Ontology Abstract Machine of Hospital Model 2:

$M_2 = (Q_2 , \sum_2 , \partial_2 , Q_{02} , F_2)$

Q_2 = { Hospital_Model , Allot_Bed , Register_patient , Admit_patient , Registration , Diagnose_patient , Patients_attendee , Registration_staff , Check_BP_of_patient }

$\sum_2$ = { has Act , hasParticipation , hasRole , for , has , hasAsParticipant , participatesIn }

Q_{02} = { Hospital_Model }

F_2 = { }

∂_2 = { (Hospital_Model , hasAct , Allot_Bed) , (Hospital_Model , hasAct , Register_patient), (Hospital_Model , hasAct , Admit_patient) , (Hospital_Model , hasParticipation , Registration), (Hospital_Model , hasAct , Diagnose_patient) , (Hospital_Model , has Role , Patients_attendee) (Admit_patient , for , Registration) , (Registration , has , Admit_patient), (Registration , has , Allot_Bed) , (Allot_Bed , for , Registration), (Register_patient , for , Registration), (Registration , has , Register_patient), (Registration , has , Diagnose_patient) , (Diagnose_patient , for , Registration), (Patients_attendee , hasAsParticipant , Registration), (Registration , participatesIn , Patients_attendee), (Hospital_Model , has Role , Registration_staff) , (Hospital_Model , has Act , Check _BP_of_patient), (Check _BP_of_patient , for , Registration) , (Registration , has , Check_BP_of_patient), (Registration_Staff , hasAsParticipant , Registration), (Registration , participatesIn , Registration_Staff) }

Merging of Ontology Abstract Machines of Hospital Model 1 and Hospital Model 2 to get the resultant Abstract Machine M_3:

$M_1 = (Q_1 , \sum_1 , \partial_1 , Q_{01} , F_1)$
$M_2 = (Q_2 , \sum_2 , \partial_2 , Q_{02} , F_2)$
$M_3 = (Q_3 , \sum_3 , \partial_3 , Q_{03} , F_3)$

Where

$Q_3 = Q_1 \cup Q_2$ = { Hospital_Model , Allot_Bed , Register_patient , Admit_patient , Registration , Diagnose_patient , Patients_attendee , Nurse , Check _BP_of_patient, Registration_staff }

$\sum_3 = \sum_1 \cup \sum_2$ = { has Act , hasParticipation , hasRole , for , has , hasAsParticipant , participatesIn }

$\partial_3 = \partial_1 \cup \partial_2$
= { (Hospital_Model , hasAct , Allot_Bed) , (Hospital_Model , hasAct , Register_patient), (Hospital_Model , hasAct , Admit_patient) , (Hospital_Model , hasParticipation , Registration), (Hospital_Model , hasAct , Diagnose_patient) , (Hospital_Model , hasRole , Patients_attendee) (Admit_patient , for , Registration) , (Registration , has , Admit_patient), (Registration , has , Allot_Bed) , (Allot_Bed , for , Registration), (Register_patient , for , Registration), (Registration , has , Register_patient), (Registration , has , Diagnose_patient) , (Diagnose_patient , for , Registration), (Patients_attendee , hasAsParticipant , Registration), (Registration , participatesIn , Patients_attendee), (Hospital_Model , has Role , Nurse) , (Nurse , hasAsParticipant , Registration), (Registration , participatesIn , Nurse) , (Hospital_Model , has Role , Registration_staff) , (Hospital_Model , has Act , Check_BP_of_patient), (Check _BP_of_patient , for , Registration) , (Registration , has , Check_BP_of_patient), (Registration_Staff , hasAsParticipant , Registration), (Registration , participatesIn , Registration_Staff) }

$F_3 = F_1 \cup F_2$ = { }

Q_{03} is determined after scanning ∂_3

Q_{03} = { Hospital_Model }

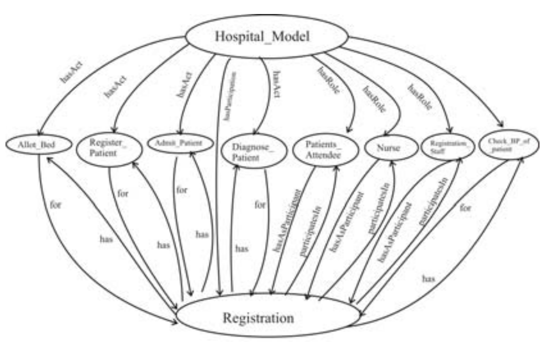

Figure 6 Merged Ontology

4.2 Ontology Matching

Considering the two ontologies of O_1 (Figure 1) and O_2 (Figure 2) representing two different hospitals as examples, the matching process is discussed in this section. Each ontology is represented as OAM to facilitate the matching process. The two ontologies are represented as ontology abstract machines defined as M_1 and M_2 respectively. Then the differences between the two ontology abstract machines namely $M_1 - M_2$ and $M_2 - M_1$ are determined. The resultant ontology abstract machines are namely M_4 and M_5. The ontologies for the ontology abstract machines M_4 and M_5 are shown in figure 7 and figure 8 respectively. Interestingly ontology O_4 represents the features those are in O_1 but not in O_2 and similarly ontology O_5 represents the features which are in O_2 but not in O_1. Hence the differences between any two ontologies determined as above help the users find out the unique features of an ontology in comparison with another ontology.

Then the symmetric difference between the two ontology abstract machines M_1 and M_2 is determined by merging the ontology abstract machines M_4 and M_5. Interestingly the symmetric difference captures the unmatched triples of the original ontologies O_1 and O_2. The ontology of symmetric difference is shown in figure 9.

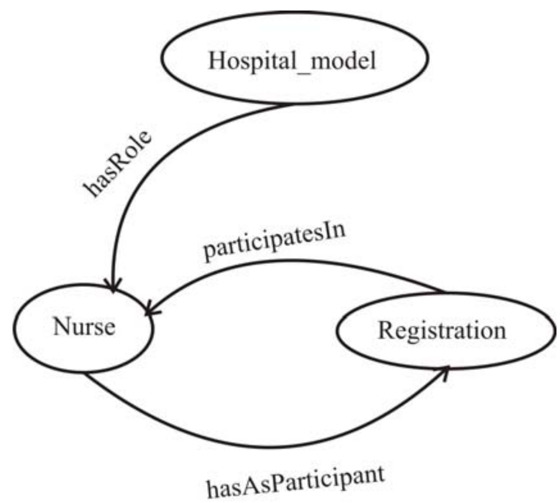

Figure 7 Ontology difference (O_1-O_2)

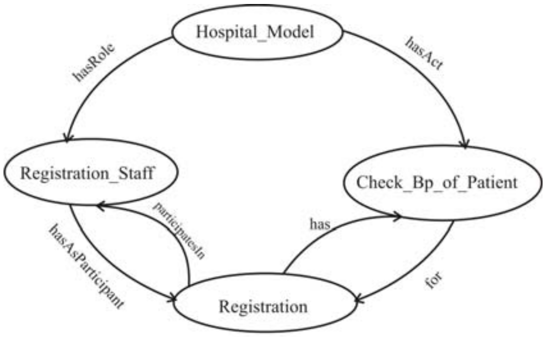

Figure 8 Ontology difference (O_2-O_1)

Difference between two ontologies using Ontology Abstract Machine(M_1-M_2):

$M_1 = (Q_1 , \sum_1 , \partial_1 , Q_{01} , F_1)$
$M_2 = (Q_2 , \sum_2 , \partial_2 , Q_{02} , F_2)$
$M_4 = M_1 – M_2 = (Q_4 , \sum_4 , \partial_4 , Q_{04} , F_4)$
Where

$\partial_4 = \partial_1 - \partial_2$
∂_4 = { (Hospital_Model , hasRole , Nurse) , (Nurse , hasAsParticipant , Registration), (Registration , participatesIn , Nurse) }

Q_4 is determined after analysing ∂_4
Q_4 = { Hospital_Model , Nurse , Registration }

$\sum_4$ is determined after analysing ∂_4
$\sum_4$ = { hasRole , hasAsParticipant , participatesIn }

Q_{04} is determined after analysing ∂_4
Q_{04} = { Hospital_Model }
F_4 is determined after analysing ∂_4
F_4 = { }

Difference between two ontologies using Ontology Abstract Machine(O_2-O_1):

$M_5 = M_2 – M_1 = (Q_5 , \sum_5 , \partial_5 , Q_{05} , F_5)$
Where
$\partial_5 = \partial_2 - \partial_1$
∂_5 = {
(Hospital_Model , has Role , Registration_staff) ,
(Hospital_Model , has Act , Check_BP_of_patient),
(Check_BP_of_patient , for , Registration) ,
(Registration , has , Check_BP_of_patient),
(Registration_Staff , hasAsParticipant , Registration),
(Registration , participatesIn , Registration_Staff) }

Q_5 is determined after analysing ∂_5
Q_5 = { Hospital_Model , Registration_staff , Nurse , Check_BP_of_patient , Registration }

$\sum_5$ is determined after analysing ∂_5
$\sum_5$ = { hasRole , has Act , for , has, hasAsParticipant , participatesIn }

Q_{05} is determined after analysing ∂_5
Q_{05} = { Hospital_Model }
F_5 is determined after analysing ∂_5
F_5 = { }

Symmetric difference between two ontologies using Ontology Abstract Machine(O_1-O_2) U ((O_2-O_1):

$M_6 = (M_1 – M_2) \cup (M_2 – M_1) = (M_4 \cup M_5) = (Q_6 , \sum_6 , \partial_6 , Q_{06} , F_6)$
Where

$Q_6 = Q_4 \cup Q_5$ = { Hospital_Model , Nurse , Registration , Registration_staff , Check_BP_of_patient }

$\sum_6 = \sum_4 \cup \sum_5$ = { hasRole , hasAsParticipant , participatesIn , has Act , for , has }

$\partial_6 = \partial_4 \cup \partial_5$ {
(Hospital_Model , hasRole , Nurse) ,
(Nurse , hasAsParticipant , Registration),
(Registration , ParticipatesIn , Nurse)
(Hospital_Model , has Role , Registration_staff) ,
(Hospital_Model , has Act , Check_BP_of_patient),
(Check _BP_of_patient , for , Registration) ,
(Registration , has , Check_BP_of_patient),
(Registration_Staff , hasAsParticipant , Registration) ,
(Registration , participatesIn , Registration_Staff) }

$F_6 = F_4 \cup F_5$ = { }

Q_{06} is determined after scanning ∂_6
Q_{06} = { Hospital_Model }

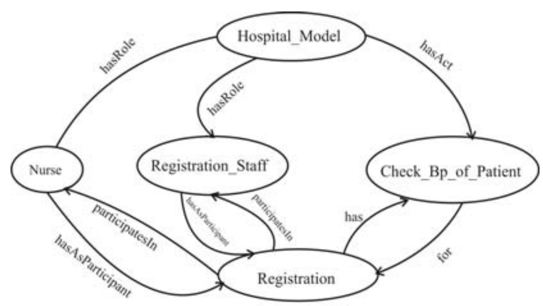

Figure 9 Ontology of Symmetric difference

Finally the matched ontology of ontologies O_1 and O_2 is obtained by determining the difference between the ontology abstract machines M_3 and M_6. It is good to recall that the ontology abstract machine M_3 is obtained by merging ontology abstract machines M_1 and M_2. M_6 represents the symmetric difference of the ontology abstract machines $M_1$and M_2. The resultant ontology is shown in figure 10.

5. 0 CONCLUSION

The proposed research work demonstrates a new approach of ontology matching and merging using ontology abstract machines. Ontology matching and merging methods are illustrated using examples from a health care domain. The results obtained are encouraging and leads to further research of relating ontology and automata in a formal way.

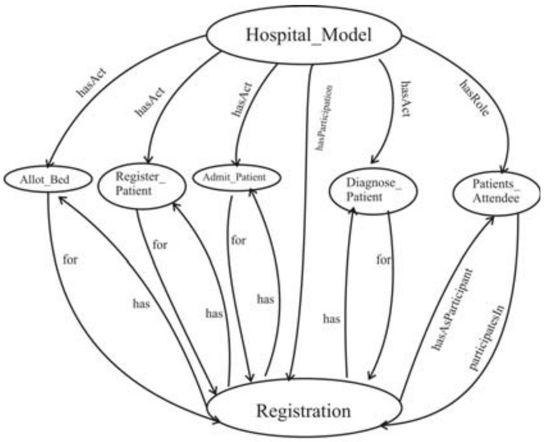

Figure 10 Matched Ontology

REFERENCES

Maedche, A., Motik, B., Silva, N., & Volz, R. (2002). MAFRA A Mapping Framework For Distributed Ontologies. *Proceedings of 13th European conference on knowledge Engineering and knowledge management(EKAW-2002),* Madrid, Spain.

Alma Delia Cuevas Rasgado & Adolfo Guzman Arenas, (2006). A Language and Algorithm for Automatic Merging of Ontologies. *Proceedings of the 15th International Conference on Computing (CIC'06), IEEE.*

Corbett, D. (2003). *Comparing and Merging Ontologies: A Concept Type Hierarchy Approach*. Springer-Verlag Berlin Heidelberg.

Doan .A, Madhavan .J, Domingos .P, & Halevy .A, (2002). Learning To Map Between Ontologies On The Semantic Web. *Proceedings of the 11th International WWW Conference* [WWW 2002], Hawaii, USA.

Rahm, E., & Bernstein, P. A. (2001). A Survey of Approaches To Automatic Schema Matching. *VLDB Journal: Very Large databases, 10*(4), 334-350.

Shen, G., Huang, Z., Zhu, X., Wang, L., & Xiang, G.(2007). Using Description Logics Reasoner for Ontology Matching. *Workshop on Intelligent Information Technology Application, IEEE.*

Chalupsky, H. (2000). OntoMorph : A Translation System for Symbolic Knowledge. *Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning*, pp. 471-482, Brecknridge, Colorado, USA.

Lee, L., Leopold, J., Albath, J., & Coalter, A. (2009). An Ontology Abstract Machine. *World Academy of Science, Engineering and Technology, 52.*

Predoiu, L., Feier, C., Scharffe, F., Jos de Bruijn, & Martin, F. – Recuerda, Dimitar Manov, & Marc Ehrig, (2006). *D4.2.2 State-of-the-art Survey on Ontology Merging and Aligning V2*.

Ehrig, M., & Staah, S. (2004). Quick Ontology Mapping. Hiroshima, Japan *, November.*

Mena. E, Illarramendi.A, Kashyap.V, & Sheth.A, (2000). OBSERVER: An Approach For Query Processing In Global Information Systems Based On Inter Operation Across Pre-Existing Ontologies. *Distributed and parallel Databases, An international journal, 8*(2), 223-271.

Koprowska, M. (2008). Remarks on Automated Ontology Merging Algorithm. *19th International Conference on Systems Engineering, IEEE.*

Choi, N., Song, I. Y., & Han, H. (2006). A Survey On Ontology Mapping. *SIGMOD Record*, *35*(3), Sep 2006.

Noy N.F., & Musen M.A. (2003). The PROMPT Suite: Interactive Tools For Ontology Merging And Mapping. *International Journal of Human Computer Studies*, *59*(96), 983 – 1024.

Shvaiko, P., & Euzenat, J. (2008). *Ten Challenges for Ontology Matching*. OTM 2008, Part II, LNCS 5332, pp. 1164–1182, Springer-Verlag Berlin Heidelberg.

Yeh, P. Z., Porter, B., & Barker, K. (2003). Using Transformations to Improve Semantic Matching. *Proceedings of K-CAP'03, Sanibel Island, Florida, USA*.October 23–25.

Mitra, P., & Wiederhold, G. (2001). An Algebra For Semantic Interoperability Of Information Sources". *In IEEE International conference on Bio informatics and Biomedical Engineering,* 174-182.

Mitra, P., Wiederhold, G., & Jannink, J. (1999). Semi-Automatic Integration Of Knowledge Sources. *Proceedings of Fusion 99*, California, USA, July 1999.

Krivoi, S., Matveyeva, L., Lukianova, Y., & Sedleckaya, O. (2008). Ontology View On Automata Theory. International Journal Information Theories & Applications, 15**.**

Stumme.G, & Maedche.A,(2001). FCA-Merge: Bottom-up Merging of Ontologies. *In Seventh International Conference on Artificial Intelligence (IJCAI'01)*, pp. 225-230, Seattle, WA, USA.