CHAPTER V
MATCHING AND MERGING OF ONTOLOGIES
USING ONTOLOGY ABSTRACT MACHINE

5.1. Introduction

The aim of this chapter is to present a comprehensive overview of ontology operations such as ontology merging, ontology difference, and ontology matching using Ontology Abstract Machine. The Ontology Abstract Machine model [136] is defined in a format similar to that used for a finite state automaton. Lee and colleagues have introduced the Ontology Abstract Machine model and the related algorithms that enable maintenance of an ontology that supports node-based access.

5.1.1. Definition: Ontology Abstract Machine

Ontology Abstract Machine is a 5-tuple representation of an ontology and is denoted by $M$

\[ M = (Q, \sum, \partial, q_0, F) \]

Where $Q$ 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 relationship 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 an element of $Q_0$ which is a set of source nodes. These are nodes with no incoming $\sum_B$ edge. This set can be identified from $\partial$.

$Q_0$ 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 $Q$. 

∂ is a set of relationships in the form of edge (node, relationship type, node), $Q \sum \rightarrow Q$, hence each element is a child node, a relationship type, or a parent node.

5.2. Ontology merging Using Ontology Abstract Machine

Considering the two ontologies of $O_1$ in Figure. 5.1 and $O_2$ in Figure. 5.2 representing two different hospitals. The merging process is illustrated and the results are presented in this section. Ontology $O_1$ is represented as ontology abstract machine $M_1$. Ontology $O_2$ is represented as ontology abstract machine $M_2$. Then the abstract machines $M_1$ and $M_2$ are combined to get their 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 5.3. Let $M_3 = (Q_3, \sum_3, \partial_3, q_{03}, F_3)$ where $Q_3$ is obtained as $Q_3 = Q_1 Q_2$. $\partial_3$ is obtained as $\partial_3 = \partial_1 \partial_2$, $F_3$ is obtained as $F_3 = F_1 F_2$ and $q_{03}$ is determined after scanning $\partial_3$.

Ontology Abstract Machine of Hospital Model 1($M_1$)

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

$Q_1 =$

{Hospital_Model, Allot_bed, Register_patient, Admit_patient, Registration,
 Diagnose_patient, Patients_attendee, Nurse}

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

$q_{01} =$ Hospital_Model

$F_1 = \{\}$

$\partial_1 =$

(Hospital_Model, hasAct, Allot_bed),
(Hospital_Model, hasAct, Register_patient),
(Hospital_Model, hasAct, Admit_patient),
(Hospital_Model, hasParticipation, Registration),
Ontology Abstract Machine of Hospital Model 2(M2)

\[ M_2 = (Q_2, \Sigma_2, \partial_2, q_{02}, F_2) \]

\[ Q_2 = \{ \text{Hospital\_Model, Allot\_bed, Register\_patient, Admit\_patient, Registration, Diagnose\_patient, Patients\_attendee, Registration\_staff, Check\_BP\_of\_patient} \} \]

\[ \Sigma_2 = \{ \text{has Act, hasParticipation, hasRole, for, has, hasAsParticipant, participatesIn} \} \]

\[ q_{02} = \{ \text{Hospital\_Model} \} \]

\[ F_2 = \{ \} \]

\[ \partial_2 = \{ \]

\[ \text{(Hospital\_Model, hasAct, Allot\_bed),} \]

\[ \text{(Hospital\_Model, hasAct, Register\_patient),} \]

\[ \text{(Hospital\_Model, hasAct, Admit\_patient),} \]
Merging of Ontology Abstract Machines of hospital model 1 and hospital model 2 to get the resultant abstract machine $M_3$:

$M_1 = (Q_1, \Sigma_1, \delta_1, q_{01}, F_1), \ M_2 = (Q_2, \Sigma_2, \delta_2, q_{02}, F_2)$

$M_3 = (Q_3, \Sigma_3, \delta_3, q_{03}, F_3)$ where

$Q_3 = Q_1 \cup Q_2 = \{\text{Hospital\_Model, Allot\_bed, Register\_patient, Admit\_patient, Registration, Diagnose\_patient, Patients\_attendee, Nurse, Check\_BP\_of\_patient, Registration\_staff}\}$

$\Sigma_3 = \Sigma_1 \cup \Sigma_2 = \{\text{has Act, hasParticipation, hasRole, for, has, hasAsParticipant, participatesIn}\}$
\[ \partial_3 = \partial_1 \cup \partial_2 = \{
\]

(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} \text{ is determined after scanning } \partial_3

q_{03} = \text{Hospital_Model}
Figure 5.1. Ontology of Hospital Model 1(O₁)
Figure 5.2. Ontology of Hospital Model 2(O₂)
Figure 5.3. Merged Ontology
5.3. Ontology Matching Using Ontology Abstract Machine

Considering the two ontologies of O₁(Figure. 5.1) and O₂(Figure. 5.2) representing two different hospitals. The matching process is illustrated and the results are presented in this section. Ontology O₁ is represented as Ontology Abstract Machine M₁. Ontology O₂ is represented as Ontology Abstract Machine M₂. Then the differences between the two Ontology Abstract Machines namely M₁-M₂ is determined. Let M₄ = M₁-M₂ = (Q₄, ∑₄, ∂₄, q₀₄, F₄) where ∂₄ is determined as ∂₄ = ∂₁-∂₂. Q₄, ∑₄, q₀₄ and F₄ are determined after analysing ∂₄. Similarly the differences between the two Ontology Abstract Machines namely M₂-M₁ is also determined. The resultant Ontology Abstract Machines are namely M₄ and M₅. The ontologies for the Ontology Abstract Machines M₄ and M₅ are shown in Figures. 5.4 and Figure 5.5 respectively. Interestingly ontology O₄ represents the features that are in O₁ but not in O₂ and similarly ontology O₅ represents the features which are in O₂ but not in O₁. The differences between any two ontologies can be determined as above leads to determine the unique features of an ontology in comparison with another ontology.

Then the symmetric difference between the two Ontology Abstract Machines M₁ and M₂ is determined by merging the Ontology Abstract Machines M₄ and M₅. Interestingly the symmetric difference captures the unmatched triples of the original ontologies O₁ and O₂. The ontology of symmetric difference is shown in Figure 5.6.

Difference between two ontologies using Ontology Abstract Machine (M₁-M₂):

\[ M₁ = (Q₁, ∑₁, ∂₁, q₀₁, F₁), \quad M₂ = (Q₂, ∑₂, ∂₂, q₀₂, F₂) \]

\[ M₄ = M₁-M₂ = (Q₄, ∑₄, ∂₄, q₀₄, F₄) \]

The components of M₄ are determined as follows:

\[ ∂₄ = ∂₁-∂₂ \]

\[ ∂₄ = \{ \]

Hospital_Model, hasRole, Nurse),
(Nurse, hasAsParticipant, Registration),
Registration, participatesIn, Nurse)
Q₄, ∑₄, q₀₄ and F₄ are determined after analysing ∂₄

Q₄ = {Hospital_Model, Nurse, Registration}

∑₄ = {hasRole, hasAsParticipant, participatesIn}

q₀₄ = Hospital_Model

F₄ = { }

**Difference between two ontologies using Ontology Abstract Machine (M₂-M₁):**

M₅ = M₂-M₁ = (Q₅, ∑₅, ∂₅, q₀₅, F₅)

The components of M₅ are determined as follows:

∂₅ = ∂₂-∂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) }

Q₅, ∑₅, q₀₅ and F₅ are determined after analysing ∂₅

Q₅ = {Hospital_Model, Registration_staff, Nurse, Check_BP_of_patient, Registration}

∑₅ = {hasRole, has Act, for, has, hasAsParticipant, participatesIn}

q₀₅ = {Hospital_Model},

F₅ = { }
Symmetric difference between two ontologies using Ontology Abstract Machine

\[(M_1-M_2) \cup (M_2-M_1)\):

\[M_6 = (M_1-M_2) \cup (M_2-M_1) = (M_4 \cup M_5) = (Q_6, \sum_6, \partial_6, q_{06}, F_6)\]

The components of \(M_6\) are determined as follows:

\[Q_6 = Q_4 \cup Q_5 = \{\text{Hospital}_\text{Model}, \text{Nurse}, \text{Registration}, \text{Registration}\_\text{staff}, \text{Check}\_\text{BP}\_\text{of}\_\text{patient}\}\]

\[\sum_6 = \sum_4 \cup \sum_5 = \{\text{hasRole}, \text{hasAsParticipant}, \text{participatesIn}, \text{has Act}, \text{for}, \text{has}\}\]

\[\partial_6 = \partial_4 \cup \partial_5\]

\[= \{\]

\[\begin{align*}
\text{Hospital}_\text{Model}, \text{hasRole}, \text{Nurse}, \\
\text{Nurse}, \text{hasAsParticipant}, \text{Registration}, \\
\text{Registration}\_\text{staff}, \text{hasRole}, \\
\text{Hospital}_\text{Model}, \text{has Role}, \text{Registration}\_\text{staff}, \\
\text{Hospital}_\text{Model}, \text{has Act}, \text{Check}\_\text{BP}\_\text{of}\_\text{patient}, \\
\text{Check}\_\text{BP}\_\text{of}\_\text{patient}, \text{for}, \text{Registration}, \\
\text{Registration}, \text{has}, \text{Check}\_\text{BP}\_\text{of}\_\text{patient}, \\
\text{Registration}\_\text{Staff}, \text{hasAsParticipant}, \text{Registration}, \\
\text{Registration}, \text{participatesIn}, \text{Registration}\_\text{Staff},
\end{align*}\]

\[\}\]

\[F_6 = F_4 \cup F_5 = \{\}\]

\[q_{06}\] is determined after scanning \(\partial_6\)

\[q_{06} = \text{Hospital}_\text{Model}\]

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 \(M_2\).
M₆ represents the symmetric difference of the Ontology Abstract Machines M₁ and M₂. The resultant ontology is shown in Figure 5.7.

The fitness of the matched ontology can be computed using a simple formula as follows:

\[
\text{Fitness (Matched Ontology)} = \frac{\text{No of Triples of Matched Ontology}}{\text{Maximum (No of triples of } O_1, \text{ No of Triples of } O_2)}
\]

Figure 5.4. Ontology difference (O₁-O₂)

Figure 5.5. Ontology difference (O₂-O₁)
Figure 5.6. Ontology of Symmetric Difference \((O_1 - O_2) \cup (O_2 - O_1)\)

Figure 5.7. Matched ontology

The fitness of the matched ontology is approximately 0.7 for our example. The union and difference operations for finite state automata do not consider the semantic characteristics of ontologies. To address this issue transformations specific to a domain can be used to realize matching of triples with semantics.
5.4. Summary

The proposed research study demonstrates a new approach of ontology management using Ontology Abstract Machines. Ontology matching, Ontology merging and Ontology difference 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.