Automatic Detection and Processing of Attributes Inconsistency for Fuzzy Ontologies Merging

Yonghong Luo*1, Zhou Yu2, Yanhao Zhuang3, Zhaopeng Zheng3
1School of Computer and Communication Engineering, Changsha University of Science and Technology
Changsha 410114, Hunan, China
2School of Information and Engineering, Henan Institute of Science and Technology
Xinxiang 453003, China
3School of Civil Engineering and Transportation, South China University of Technology
Guangzhou 510641, China
*Corresponding author, e-mail: 58720150@qq.com

Abstract
Semantic fusion of multiple data sources and semantic interoperability between heterogeneous systems in distributed environment can be implemented through integrating multiple fuzzy local ontologies. However, ontology merging is one of the valid ways for ontology integration. In order to solve the problem of attributes inconsistency for concept mapping in fuzzy ontology merging system, we present an automatic detection algorithm of inconsistency for the range, number and membership grade of attributes between mapping concepts, and adopt corresponding processing strategy during the fuzzy ontologies merging according to the different types of attributes inconsistency. Experiment results show that with regard to merging accuracy, the fuzzy ontology merging system in which the automatic detection algorithm and processing strategy of attributes inconsistency is embedded is better than those traditional ontology merging systems like GLUE, PROMPT and Chimaera.

Keywords: fuzzy ontology, ontology merging, automatic detection, attributes inconsistency, membership grade

1. Introduction
As precise ontology model can’t fully represent uncertain knowledge and vague information such as “young people”, “very expensive” in many application domains, it is necessary to introduce fuzzy ontology for describing fuzzy information. Straccia was the first man to propose the idea of fuzzy ontology [1], which can express fuzzy concepts and fuzzy relations between concepts through integrating fuzzy logic [2]. Nowadays, applied ranges of fuzzy ontology has been gradually extended; but constructing fuzzy ontology with different languages, methods and descriptive ways might result in heterogeneity of multiple ontologies in the same So, semantic fusion of multiple data sources and semantic interoperability between heterogeneous systems in distributed environment such as grid computing [3, 4] and cloud computing [5, 6] usually can be implemented through integrating multiple fuzzy local ontologies. However, ontology merging is one of the valid ways for ontology integration.

Currently there are few works on fuzzy ontology merging. Authors in literature [7, 8] proposed an integration method of fuzzy ontology based on consensus in which integrated algorithm have not be verified by experiments, so the accuracy and effectiveness of fuzzy ontology integration can not be ensure. Literature [9] used concept lattice gluing to merge fuzzy ontologies, and proposed a method of fuzzy ontology merging based on fuzzy concept gluing. Existing various methods of precise ontologies merging mainly include GLUE [10], Hcone [11], PROMPT [12], ASMOV [13] and so on. We all know that these precise methods are semi-automatic, and are not suitable for merging fuzzy ontologies. So, there are still no effective solutions to solve the problem of attributes inconsistency for concept mapping in fuzzy ontologies merging system at present. For the inconsistencies regarding the range, number and membership grade of attributes, fuzzy ontology merging systems need more human intervention to repair inconsistency relations between concepts.
In order to effectively deal with the problem of attributes inconsistency for concept mapping in fuzzy ontologies merging system, and strive to realize full automation of fuzzy ontology merging, we present automatic detection algorithm and corresponding processing strategy for attributes inconsistency according to the aforementioned situation. The elimination of attributes inconsistency for concept mapping during fuzzy ontologies merging includes following three steps: (1) creating initial mapping solution between source fuzzy ontologies based on similarity degree calculating of concept; (2) using detection algorithm of attributes inconsistency to find out all of inconsistent mappings from initial mapping solution; (3) performing corresponding processing strategies according to the types of attributes inconsistency, respectively.

2. Description of Fuzzy Ontology

Definition 1. A fuzzy ontology can be defined as a 7-tuple form: $\text{FO} = (C, A, V, U, R, Z, I)$ where $C$ is set of concepts, a concept is fuzzy concept if it contains at least one attribute with membership grade or it is set of fuzzy set; $A$ is set of attributes belonging to the concepts in $C$. An attribute is a fuzzy attribute if its value is a fuzzy set; $V$ is the domain of $A$. $V$ also can be expressed as a set of attribute values, and $V = \bigcup_{a \in A} V_a$ where $V_a$ is the domain of the attribute $a$; $U$ is set of membership functions, and the range of each membership function is a concrete fuzzy concept; $R$ is set of fuzzy relations between concepts: $R = \{R_1, R_2, ..., R_n\}$ where $R_i \subseteq C \times \times (0,1)$ for $i=1, 2, ..., n$. A relation is then a set of pairs of concepts with a weight representing the degree to which the relationship should belong; $Z$ is set of axioms, which can be interpreted as integrity constraints or relationships between instances and concepts; $I$ is set of instances. An instance is fuzzy instance if its corresponding concept is fuzzy concept.

Definition 2. A fuzzy concept $\text{FC}$ can be defined as a triple form: $\text{FC} = (A^\text{FC}, V^\text{FC}, f^\text{FC})$ where $A^\text{FC} \subseteq A$ is set of attributes describing the concept, including fuzzy attributes and precise attributes; $V^\text{FC} \subseteq V$ is the union of each attribute domain; $f^\text{FC}$ is set of fuzzy functions which represent the degrees that the attributes of fuzzy concepts describe their concept instances, $f^\text{FC}: A^\text{FC} \to [0,1]$.

Definition 3. A concrete fuzzy concept $\text{cfc} \subseteq C$ can be defined as a 4-tuple from: $\text{cfc} = (V_{\text{cfc}}, V'_{\text{cfc}}, L_{\text{cfc}}, f_{\text{cfc}})$ where $\text{cfc}$ is the unique identifier of concrete fuzzy concept; $V_{\text{cfc}} \subseteq V$ is the domain of concrete fuzzy concept $\text{cfc}$; $V'_{\text{cfc}} \subseteq (0,1]$ presents fuzzy values of the concrete set $V_{\text{cfc}}$; $L_{\text{cfc}} \subseteq V$ models linguistic qualifiers, which is determined by the strength of the attribute value in $V_{\text{cfc}}$; $f_{\text{cfc}}: V_{\text{cfc}} \to V'_{\text{cfc}}$, $\forall v \in V_{\text{cfc}}, f_{\text{cfc}}(v) \in V'_{\text{cfc}}$.

3. Fuzzy Ontology Mapping

According to the definitions of concrete fuzzy concept (property), the similarity degree between fuzzy concepts or fuzzy properties depends on their fuzzy sets. We assume that $\text{FC}_1 = \{A, V, f_A\}$ and $\text{FC}_2 = \{A', V', f_A'\}$ are the concrete fuzzy concepts. $V$ and $V'$ are the concrete sets, $A$ and $A'$ are the corresponding fuzzy sets of set $V$ and set $V'$ respectively. We specify $U = V \cup V'$, the operations between fuzzy subsets can be defined as follows:

\[ \forall x \in U, f_{A \cap A'}(x) = \min \{f_A(x), f_A'(x)\} \quad (1) \]

\[ \forall x \in U, f_{A \cup A'}(x) = \max \{f_A(x), f_A'(x)\} \quad (2) \]

Automatic Detection and Processing of Attributes Inconsistency for Fuzzy… (Yonghong Luo)
∀ x ∈ U, f_{A\Delta A'}(x) = \min \{ f_A(x), 1 - f_A(x) \} \tag{3}

∀ x ∈ U, f_{A\Delta A'}(x) = \min \{ 1 - f_A(x), f_A(x) \} \tag{4}

A-A' is the fuzzy subset of elements that belong to A and not to A', A'-A is the fuzzy subset of elements that belong to A' and not to A. So, the similarity degree between attribute (concept) a_1 and attribute (concept) a_2 can be calculated with the following expressions:

\[ \text{Sim}(a_1, a_2) = \sup_{x \in U} f_{A \cap A'}(x) \rightarrow \text{Sim}(a_1, a_2) = \sup \{ \min \{ f_A(x), f_{A'}(x) \} \} \tag{5} \]

\[ \text{Sim}(a_1, a_2) = \sup_{x \in U} f_{A \cap A'}(x) \rightarrow \text{Sim}(a_1, a_2) = \sup \{ \max \{ f_A(x), f_{A'}(x) \} \} \tag{6} \]

\[ \text{Sim}(a_1, a_2) = \sup_{x \in U} f_{A \cap A'}(x) \rightarrow \text{Sim}(a_1, a_2) = \sup \{ \min \{ f_A(x), 1 - f_{A'}(x) \} \} \tag{7} \]

\[ \text{Sim}(a_1, a_2) = \sup_{x \in U} f_{A \cap A'}(x) \rightarrow \text{Sim}(a_1, a_2) = \sup \{ \min \{ 1 - f_A(x), f_{A'}(x) \} \} \tag{8} \]

Therefore, the similarity degree of attributes between fuzzy concepts of FC_1 and FC_2 can be calculated as follows:

\[ AS(FC_1, FC_2) = \frac{\sum_{i=1}^{n} \text{Sim}(a_1^{FC_1}, a_2^{FC_2})}{\max(\|FC_1\|, \|FC_2\|)} \tag{9} \]

Where a^{FC_1} and a^{FC_2} are attributes belonging to FC_1 and FC_2, respectively; |FC_1| and |FC_2| denote the number of attributes for FC_1 and FC_2, respectively; n is the maximum value of |FC_1| and |FC_2|.

The calculation of similarity degree between fuzzy concepts usually needs to take into multiple important factors of concepts like name, semantic and so on account. Here we determine the final similarity degree between fuzzy concepts through aggregating name similarity, semantic similarity and attribute similarity.

The name similarity between concepts [14] is calculated as follows:

\[ NS(FC_1, FC_2) = \frac{\max(l(FC_1), l(FC_2)) - Edist(FC_1, FC_2)}{\max(l(FC_1), l(FC_2))} \tag{10} \]

The semantic similarity between concepts [13] is calculated as follows:

\[ SS(FC_1, FC_2) = \frac{2 \times IC(\text{sub}(FC_1, FC_2))}{IC(FC_1) + IC(FC_2)} \tag{11} \]

Where IC [15] is the function that returns information content of a concept [16], the information content of FC can be calculated as follows:

\[ IC(FC) = 1 - \frac{\log(hypo(FC) + 1)}{\log(TN)} \tag{12} \]

Finally, the integrated similarity degree between fuzzy concepts can be calculated as follows:

\[ S(FC_1, FC_2) = \frac{w_a \times AS(FC_1, FC_2) + w_n \times NS(FC_1, FC_2) + w_s \times SS(FC_1, FC_2)}{3} \tag{13} \]
Where \( w_a, w_n \) and \( w_s \) denote the weight of attribute similarity, name similarity and semantic similarity respectively, and \( w_a + w_n + w_s = 1 \).

With the formula of similarity degree between fuzzy concepts, the initial mapping algorithm between fuzzy ontologies is as follows:

**Input:** Source ontologies \( FO_1 \) and \( FO_2 \), \( w_a, w_n, w_s 

**Output:** mapping list of concepts between \( FO_1 \) and \( FO_2 

**Begin**

**Step 1:** \( n=|FO_1|; m=|FO_2|; k=0; l=|FO_1| \) denote the number of concepts in \( FO_1 

**Step 2:** for \( i=1 \) to \( n \) do 

**Step 3:** for \( j=1 \) to \( m \) do 

**Step 4:** \( s= S(C_{1i},C_{2j}); // C_{xy} indicate that the current sequence number of concept is \( y \) in \( FO_x \) 

**Step 5:** if \( s > \lambda \) then 

**Step 6:** \( \text{mapping_list}[k++]=( C_{1i},C_{2j}); \)

**Step 7:** end if 

**Step 8:** end for 

**Step 9:** end for 

**End.**

**4. Detection Algorithm of Attributes Inconsistency**

**Definition 4 (Range Inconsistency of Attributes).** Given that the range of object attribute \( a \) of concept \( C \) in ontology \( O_1 \) is \( R \), the range of object attribute \( a' \) of concept \( C' \) in ontology \( O_2 \) is \( R' \), and \( R \neq R' \). Then, we call mapping \( (C, C') \) as range inconsistency of attribute because of its violation of property subsumption criteria.

**Definition 5 (Number Inconsistency of Attribute).** Given that the attribute set of concept \( C \) is \( \{a_1, a_2, \ldots, a_m\} \), the attribute set of concept \( C' \) is \( \{a'_1, a'_2, \ldots, a'_n\} \), and \( m \neq n \).

Then, we call mapping \( (C, C') \) as number inconsistency of attribute.

**Definition 6 (Membership Inconsistency of Attribute).** Given that the attribute set of concept \( C \) is \( A \), the attribute set of concept \( C' \) is \( A' \). Then, we call mapping \( (C, C') \) as membership inconsistency of attribute between \( C \) and \( C' \) if \( u_{a}(C) \neq u_{a}(C') \) for \( \forall a \in A \land A' \).

The detection algorithm of attributes inconsistency is as follows:

**Input:** mapping_list 

**Output:** different type sets of attributes inconsistency

**Begin**

**Step 1:** \( n=\text{mapping_list.length} \); 

**Step 2:** for \( i=0; i<n-1; i++ \) do 

**Step 3:** Mapping \( \text{mp}_1=\text{mapping_list}(i); \)

**Step 4:** for \( j=i+1; j<n; j++ \) do 

**Step 5:** Mapping \( \text{mp}_2=\text{mapping_list}(j); \)

**Step 6:** for each attribute in \( \text{mp}_1.\text{Concept}_1 \) 

**Step 7:** If (attribute.rangeConcept disjoint \( mp_2.\text{Concept}_2 \)) then 

**Step 8:** adding \( \text{mp}_1 \) to range_inconsistency[]; 

**Step 9:** end if 

**Step 10:** end for 

**Step 11:** end for 

**Step 12:** end for 

**Step 13:** for \( i=0; i<n; i++ \) do 

**Step 14:** Mapping \( \text{mp}=\text{mapping_list}(i); \)

**Step 15:** \( A=\text{get_attributes(mp.\text{Concept}_1)}; \)

**Step 16:** \( B=\text{get_attributes(mp.\text{Concept}_2)}; \)

**Step 17:** if \( (|A|\neq|B|) \) then 

**Step 18:** adding mp to number_inconsistency[]; 

**Step 19:** end if 

**Step 20:** end for
Step 21: for (i=0;i<n;i++) do
Step 22: Mapping mp=mapping_list(i);
Step 23: for ∀a in (mp.Concept1.A \n mp.Concept2.B) do
Step 24: if u_a(mp.Concept1)!=u_a(mp.Concept2) then
Step 25: adding mp into membership_inconsistency[];
Step 26: end if
Step 27: end for
Step 28: end for
End.

5. Processing of attribute inconsistency

The initial mapping list obtained from fuzzy ontologies mapping mainly contains three types of attributes inconsistency: range inconsistency, number inconsistency and membership degree inconsistency.

(1) Processing for range inconsistency of attribute

Problem description: for mapping (FC1, FC2), FC1 is a concept in ontology O1 that includes a object attribute p1 with range of R1; FC2 is a concept in ontology O2 that also includes a object attribute p2 with range of R2: p1 = p2 \n R1 \n R2.

Merging processing strategy: we reset the range for object attribute p generated by p1 and p2 after FC1 and FC2 were merged into FC, and let R(p) = R1(p1) \cup R2(p2).

(2) Processing for number inconsistency of attribute

Problem description: for mapping (FC1, FC2), FC1 is an concept in ontology O1 with attributes set A1 = {a11, a12, ..., a1m}, FC2 is an concept in ontology O2 with attributes set A2 = {a21, a22, ..., a2n}, but m \n n.

Merging processing strategy: Merging the concepts of FC1 and FC2 into a concept FC, then the attributes set A of concept FC needs to be set to the union of A1 and A2, i.e. A = A1 \cup A2 and |A| = m + n - |A1 \n A2|.

(3) Processing for membership grade of attribute

Problem description: for mapping (FC1, FC2), FC1 is an concept in ontology O1 with attributes set A1, FC2 is an concept in ontology O2 with attributes set A2. For ∀a ∈ A1 \n A2, there will always be u_a(FC1) \n u_a(FC2).

Merging processing strategy: First merging the concepts of FC1 and FC2 into a concept FC, i.e. (FC1, FC2) \rightarrow C. If there will always be u_a(FC1) \n u_a(FC2) for ∀a ∈ A1 \n A2, we can calculate the membership grade of concept FC on attribute a as follows:

\[ u_a(FC) = \min \{u_a(FC_1), u_a(FC_2)\} \quad (14) \]

6. Experimental Results and Analysis

In order to check if the detection algorithm of attributes inconsistency and corresponding processing strategy proposed in this paper are effective, we developed a fuzzy ontology merging system based on the detection algorithm and processing strategy (FOMS), in which we carried out the experiment of fuzzy ontology merging for five fuzzy ontologies obtained from modified OAEI Conference ontology. The experimental data is shown in Table 1.

<table>
<thead>
<tr>
<th>Table 1. Description of Experimental Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ontology name</td>
</tr>
<tr>
<td>---------------</td>
</tr>
<tr>
<td>lasted</td>
</tr>
<tr>
<td>Edas</td>
</tr>
<tr>
<td>openConf</td>
</tr>
<tr>
<td>Confious</td>
</tr>
<tr>
<td>Linklings</td>
</tr>
</tbody>
</table>

TELKOMNIKA Vol. 11, No. 11, November 2013: 6514 – 6520
We first implemented the pairwise merging for five ontologies in Table 1 with FOMS, GLUE, PROMPT and Chimaera. Then, we compared the execution time of FOMS with those of GLUE, PROMPT and Chimaera, the comparison results are shown in Table 2.

Table 2. The Time Consumption of Comparison for All Merging Systems

<table>
<thead>
<tr>
<th>Merging objects</th>
<th>FOMS</th>
<th>GLUE</th>
<th>PROMPT</th>
<th>Chimaera</th>
</tr>
</thead>
<tbody>
<tr>
<td>(lasted, Edas)</td>
<td>753</td>
<td>749</td>
<td>755</td>
<td>756</td>
</tr>
<tr>
<td>(lasted, openConf)</td>
<td>528</td>
<td>535</td>
<td>524</td>
<td>522</td>
</tr>
<tr>
<td>(lasted, Confious)</td>
<td>504</td>
<td>511</td>
<td>496</td>
<td>498</td>
</tr>
<tr>
<td>(lasted, Linklings)</td>
<td>369</td>
<td>366</td>
<td>374</td>
<td>377</td>
</tr>
<tr>
<td>(Edas, openConf)</td>
<td>492</td>
<td>500</td>
<td>492</td>
<td>490</td>
</tr>
<tr>
<td>(Edas, Confious)</td>
<td>381</td>
<td>382</td>
<td>391</td>
<td>378</td>
</tr>
<tr>
<td>(Edas, Linklings)</td>
<td>258</td>
<td>246</td>
<td>263</td>
<td>259</td>
</tr>
<tr>
<td>(openConf, Confious)</td>
<td>221</td>
<td>223</td>
<td>234</td>
<td>222</td>
</tr>
<tr>
<td>(openConf, Linklings)</td>
<td>233</td>
<td>236</td>
<td>242</td>
<td>245</td>
</tr>
<tr>
<td>(Confious, Linklings)</td>
<td>200</td>
<td>211</td>
<td>206</td>
<td>213</td>
</tr>
</tbody>
</table>

Table 2 indicates that the time consumption of FOMS is almost equal to those of GLUE, PROMPT and Chimaera, even the average time consumption of FOMS is slightly lower than those of GLUE, PROMPT and Chimaera. For the accuracy of merging, FOMS proposed in this paper is obviously higher than GLUE, PROMPT and Chimaera (shown in Figure 1). In Figure 1, E/o represents the ontology merging between Edan and openConf, the representations for E/C, E/L etc are similar to E/o.

Therefore, the method proposed in this paper is more suitable for the merging of fuzzy ontologies than methods for the merging of precise ontologies like GLUE, PROMPT. During the validation of initial mappings in FOMS, the system only needs to execute pruning based on the results of depth analysis, thus reduces the number of suspected erroneous mappings. So, it saves time and cost of overall system.

![Figure 1. The Comparison of Merging Accuracy for All Merging Systems](image)

7. Conclusion

Due to the differences on tool and language for different R&D teams to construct fuzzy ontologies, especially on the understanding and description of concept structure, so that it can lead to the heterogeneity of various fuzzy ontologies in the same domain. Therefore, we have presented an automatic detection algorithm for the attributes inconsistency of concepts mapping in this paper, and implemented a fuzzy ontology merging system (FOMS). FOMS only need to analyze the results of initial mapping solution, then can find out all of the attribute inconsistency
mappings and correct them with processing strategy of attribute inconsistency. Experimental results demonstrate that FOMS has obtained the desired effect on merging accuracy, time consumption and automatic detection ability of consistency. In the future, we will highlight the following works: (1) Modifying the detection algorithm of attribute inconsistency to improve the merging accuracy for FOMS, and applying it to the domain of intelligent transport system. (2) For fuzzy ontologies described with different languages, studying on the automatic detection of semantic inconsistency during the merging of fuzzy ontologies.

Acknowledgements

This work was supported by National Natural Science Foundation of China (Grant no. 51277015) and Hunan Province Science and Technology Planning Project of China (Grant no. 2011GK3114).

References