A Service Clustering Approach Based on Semantics Ant Colony Algorithm

Zhou Xiang-bing1,3, Luo qi2
1 Dep of Software Key Lab of Sichuan Normal University,
Chengdu Sichuan 610068, China
8017zk@gmail.com
2School of computer science, Wuhan Institute of Technology
Wuhan 430073,China
luoqi@gmail.com
3Department of Mathematics , Aba Teachers College,
Chengdu Sichuan 611741, China

Abstract—In order to solve accuracy of Topic Maps of navigation and positioning for service clutering in the distributed software system,we proposed a navigation and positioning model of Topic Maps based on ontology relation(NPTMO), and according to NPTMO, we build a service clutering model and semantics ant colony algorithm. Therefore, which implement navigation and positioning of resources from relation of Topic Maps. Firstly, we define ontology relation and calculate approach, and analyze associate characteristic of Topic Maps based on ontology; Secondly, we analyze and study to match and merge approach of Topic Maps, and employ tag relation of RDF, OWL-S, WSDL and XTM what to build navigation and positioning system,and build a service clutering model, on the basic of these context above description. We design a semantisc ant colony algorithm(SACA) to calculate optimum solution of the model, and compared SACA with basic ant colony algorithm(ACA) showed that results of SACA was better than ACA. Finally, we define recall rate of ontology, accuracy of Topic Maps and efficiency of SACA to evaluate the approach showed that increased accuracy of Topic Maps connect resources.

Index Terms—Ontology relation;Service clutering model; Navigation and positioning; Semantisc ant colony algorithm; Semantics web service composition

I. INTRODUCTION

Topic Maps have been published as ISO 13250 standard since 1999, which efficiently apply many fields such as Enterprise Information Integration, web sites and portals, knowledge management and knowlege organization, e-learning, and more[1]. In fact, Topic Map is also a ontology, and has semantics recognition capability, which is a structuring semantics in the information resources layer, and employ relation to describe semantics among concept.Then obtain some different resources to guide other resources connection. Topic Map is different from ontology, which provide a knowledge code topic-oriented and standard-based, and the semantics has been well defined before, which own the simultaneously description langage and query langage. To build a knowledge navigation interface. Therefore, is known as bridge of information management and knowledge management, which is a locator of GPS in the information resources. But Topic Map is important difference from Semantic Web, which is applied to description of concept formalization and classification. However, Topic Map is applied to implement formalization of index and glossary in the building process[2], therefore, give Semantics Web high accuracy degree. But interrelated research and application of Topic Maps is still developing. Therefore, many researcher attempt to contribute beneficially theory and approach, which has proposed a lot of approach and some open sources framework at present. For example, Jung-Min Kim,Hyopil Shin and Hyoung-Joo Kim proposed schema and constraints-based matching and merging of Topic Maps[3], Myongho Yi proposed information organization and retrieval using a Topic Maps-based ontology-Results of a task-based evaluation[4], however open source software mainly have TM4J , TMTab, Omnigator, StarTree and more[5]. In particular, some international conferences on Topic Maps have been held, such as International Conference on Topic Maps Research and Applications(TMRA) have been held since 2005 at Leipzig, in Germany[6], which had discussed and proposed a lot of approach and application that was relevant to Topic Maps, included matching, similarity and description of Topic Maps, and other.

II. RELATION BASIC KNOWLEGDE

Topic Map is composed of topic, association, occurrence, facet, Scope and other items; Which showed in Fig.1[11]. In fact, Topic Map is also a ontology, and has semantics recognition capability, which is a structuring semantics in the information resources layer, and employ relation to describe semantics among concept. Then obtain some different resources to guide other resources connection. Topic Map is different from ontology, which provide a knowledge code topic-oriented and standard-based, and the semantics has been well defined before, which own the simultaneously description langage and query langage. To build a knowledge navigation interface. Therefore, is known as bridge of information management and knowledge management, which is a locator of GPS in the information resources. But Topic Map is important difference from Semantic Web, which is applied to description of concept formalization and classification. However, Topic Map is applied to implement formalization of index and glossary in the building process[2], therefore, give Semantics Web high accuracy degree. But interrelated research and application of Topic Maps is still developing. Therefore, many researcher attempt to contribute beneficially theory and approach, which has proposed a lot of approach and some open sources framework at present. For example, Jung-Min Kim,Hyopil Shin and Hyoung-Joo Kim proposed schema and constraints-based matching and merging of Topic Maps[3], Myongho Yi proposed information organization and retrieval using a Topic Maps-based ontology-Results of a task-based evaluation[4], however open source software mainly have TM4J , TMTab, Omnigator, StarTree and more[5]. In particular, some international conferences on Topic Maps have been held, such as International Conference on Topic Maps Research and Applications(TMRA) have been held since 2005 at Leipzig, in Germany[6], which had discussed and proposed a lot of approach and application that was relevant to Topic Maps, included matching, similarity and description of Topic Maps, and other.

© 2009 ACADEMY PUBLISHER
doi:10.4304/jsw.4.9.1022-1031
(4) Support machine learning;
(5) Manage distributing knowledge and information;
(6) Compose information and knowledge;
(7) Have navigation and positioning capability of information resources;
(8) Have semantics discernment and extraction capability;
(9) Provide the best flexible structure;
(10) Possess facet classification of Topic Maps.

Now, according to regulation and application of Topic Maps, we build a navigation and positioning framework of Topic Maps, which are simplified TMNLF; To implement service clustering model. Therefore, we divide TMNLF into five layers, which include application layer, semantics layer, information layer, resource layer and Topic Maps layer, showed in Fig.4. At the same time, we adopt semantics to implement resource extraction and recognition positioning, however, at present, structuring data storage employ relation database to store, such as SQLServer, Oracle, and so on. But semi-structured data have been applied many fields, especially web service application, and WSDL lack semantics description; which have a large challenge of semantics recognition and extraction in the web service, and for navigation and positioning of resource is also difficult. Therefore, we must solve four problems following:

(1) Between ontology and RDF of mapping regulation;
(2) Between WSDL and RDF of mapping regulation;
(3) Among ontology, WSDL and RDF of navigation and positioning approach;
(4) Service clustering approach.

In the Fig.3, Application Layer: which is composed of service consumer, service provider, service maker, information searcher, services. To provide result of demand and feedback, and develop difference business logic function, and other.

Semantics Layer: which implement semantics interoperability and composition, generally employ OWL-S and WSDO to describe, based on ontology.

Information Layer: all kinds of business systems compose the layer, and according to demand of application layer and recognition derivation of semantics layer meet business function of web service (WS) and semantics web service (SWS).

Resource Layer: all kinds of data resources compose the layer, those resources may be distributed difference fields, and the data storage is also varied, which is structured relation database, unstructured and semi-structured XML data, and other storage method.

Topic Maps layer: the layer is aim to navigation and positioning for application layer, semantics layer, resource layer and information layer.

### III. NAVIGATION AND POSITIONING MODEL OF SERVICE CLUSTERING BASED ON ONTOLOGY RELATION

#### A. Theory Foundation of Navigation and Positioning Model

**Definition 1** Ontology relation can be defined five tuples: \( OR = \langle O, S, D, L, R, M(a, b) \rangle \), where

\( O \) is ontology, and \( O = \langle C, P, R, H, r, \psi, \alpha \rangle \), then an ontology set is \( O = \{ o_1, o_2, \ldots, o_n \} \), which hierarchical concept \( H^r(\alpha) \subseteq C \times C \) is a directed relation; axiom set \( A^O \) must meet constraint of concept \( C \) and relation \( R \) in the ontology; Therefore, an ontology relation can define \( O_{nt} = \langle O, \xi, \psi, \alpha \rangle \), where, \( \xi \) is an ontology logic language, \( \psi \) is ontology structure dictionary. If \( C^\xi = \{ c_1, c_2, \ldots, c_n \} \) is \( O \)'s concept node set, \( R^\xi = \{ r_1, r_2, \ldots, r_n \} \) is relation node set, then denote \( Arc = \langle C \times R \rangle \) and \( Arc = \langle C \times R \rangle \) as relation node set, then denote \( Arc = \langle C \times R \rangle \) and \( Arc = \langle C \times R \rangle \).
(2) S denote source state of ontology, if define $S_\{S_1,S_2,...,S_n\}$, then adopt $S(O)$ to denote $S$.
(3) DL denote description language of ontology, which include OWL-S and WSMO, and define $DL=OS,WM,Sign=\sum$.
(4) $R$ is relation among ontology, for $\forall O\rightarrow S(O),\exists DL \rightarrow f:O \rightarrow R=(O \times O)$.
(5) $M(a,b)$ denotes mapping and matching approach of ontology.

1) To define ontology matching: $M(a)=\langle So,To,Mr,P,I,Pf\rangle$, which separately denote source ontology, target ontology, mapping relation, constraint parameter, mapping interface and support platform. Where,
$So=\{so_1,so_2,...,so_n\}$, which employ tree structure $T_1(So,\alpha)$ to describe; $To=\{to_1,to_2,...,to_m\}$, which employ tree structure $T_2(So,\beta)$ to describe; $P=\{p_1,p_2,...\}$, which employ queue $Q(P)$ to express.
At the time, mapping relation denote: $Mr=(( T_1(So)\times T_2(To)), Q(P))\rightarrow(( T_2(To)\times T_1(So), Q(P))$. If $Mr=\{mr_1,mr_2,...,mr_r\}$ is a mapping relation set, and meet $Mr(P)\rightarrow\langle So,To\rangle$; then adopt (1) to calculate similarity between source ontology and target ontology, and select the strongest similarity as mapping object.

$$Sim(So,To)=\sum_{i,j=0}^{n} \frac{\sum(\mid Os_{i,j}-T_i(So,\alpha), \mid Os_{i,j}-T_i(To,\beta))}{\sum_{i,j=0}^{n}(\mid Os_{i,j}-T_i(So,\alpha))^2 + \sum_{i,j=0}^{n}(\mid Os_{i,j}-T_i(To,\beta))^2}$$

In the (1), which employ improved cosine theorem to define similarity, where, $(p,k)$ and $(q,k)$ separately denote corresponding relation of source ontology set and target ontology set to ontology set, and their corresponding relation probability separately define $P(p|k)=\frac{\mid So \mid}{\mid Os \mid}$, $P(q|k)=\frac{\mid To \mid}{\mid Os \mid}$, and $So,To\subset Os$. $k,i,j$ separately denote parameters of mapping relation set, constraints and so on.

Algorithm 1 Ontology mapping algorithm.

```
Input: So,To
Output: The best result of ontology mapping
1) $\forall So,To \in Os, \exists M(a) \rightarrow OR$;
2) if $\{So \rightarrow To \rightarrow Mr \} \& \& (P \rightarrow (So,To))$,
3) $(So,To) \rightarrow M(a)(\alpha \leftarrow \beta)$; //Source ontology and target ontology shake hands via interface I in the $M(a)$, which use data structure with Tree and Queue to express.
4) for(int $r=0$; $r<\text{num}$$r++;$) //num is given number, and num is(natural number).
5) Evaluating (Given(Ex)(x)) $\rightarrow RC$;
6) \{ Computing($P(pk),P(qk)$)$\rightarrow V(V_1, V_2)$; //Obtain a pair of value.
7) if($\frac{Sim(So,To),C(P,X)}{\mid Os \mid}$)
8) Calculating and Running $(\mid Sim(So,To),C(P,X)\rightarrow V[Sim,C]_\rightarrow$ //Obtain a group of Sim and $C\rightarrow$ value.
9) Obtain(max $\{V[Sim,C],Ex(x)\}$)$\rightarrow$ mapping $(So,To)$;
10) return mapping result; }
```

2) To define ontology matching $M(b)=\langle Os,Mo(a)\rangle$, if $\forall O_1, O_2 (\in Os)$'s membership function denote $M_1(a), M_2(a)$, then employ weight hamming formula to calculate distance between $M_1(a)$ and $M_2(a)$:

$$R_\alpha(O_1, O_2)=\frac{1}{n} \sum_{i=1}^{n} w(a_i)|M_1(a_i)-M_2(a_i)|$$

where $W(a)$ is weight coefficient on the $a_i$, and meet $\frac{1}{n} \sum_{i=1}^{n} w(a_i)=1$. And employ hamming nearitude to calculate distance among ontology.

$$N_\alpha(O_1, O_2)=1-R_\alpha(O_1, O_2)$$

Now, still define a ontology matching entropy, which reach accurately matching among ontology, thus employ $O(M_1(a_1), M_2(a_2),..., M_2(a_n))$ to denote ontology vector, and we employ Draka entropy to express ontology matching entropy.

$$H(O)=\frac{1}{n\ln n} \sum_{i=1}^{n} [\frac{M_1(\alpha_i)\ln M_1(\alpha_i)-(1-M_1(\alpha_i))\ln(1-M_1(\alpha_i))]}$$

According to above formula in the 2), we define matching formula following

$$Match(O_1, O_2)=\kappa \frac{N_\alpha(O_1, O_2)}{H(O)} \cdot R_\alpha(O_1, O_2)$$

where, $\kappa$ denote a proportionality coefficient between hamming nearitude and Draka matching entropy.
We employ a online bookshop system\(^{(18)}\) (OBS) to explain calculating approach of Algorithm 1 and Algorithm 2, and adopt OBS’squery to explain, which include four query terms: Bookname, Author, Press, Full-text, is spimled to BAPF, showed in Fig.5, therefore, at the moment, \(\text{ontology set: } O_{s} = \{o_{s1},...,o_{s150}\}\), \(\text{source ontology set: } O_{s} = \{so_{1},...,so_{15}\}\) and target ontology set: \(T_{o} = \{to_{1},...,to_{j}\}\) \((j\in [21,26])\), which calculated result showed in table 1 and table 2.

### Table 1 Calculating approach of ontology mapping

<table>
<thead>
<tr>
<th>(S_{o} )</th>
<th>(T_{o} )</th>
<th>(P(k) )</th>
<th>(P(q(k)) )</th>
<th>(Sim(O_{s},O_{o}) )</th>
<th>(C_{d}(P,X) )</th>
<th>(E(o) )</th>
<th>(R_{s}(O_{s},O_{o}) )</th>
<th>(w(a_{i}) )</th>
<th>(N_{s}(O_{s},O_{o}) )</th>
<th>(H(O) )</th>
<th>(Match(O_{s},O_{o}) )</th>
<th>(\kappa=3/4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>21</td>
<td>0.100</td>
<td>0.140</td>
<td>0.445</td>
<td>0.516</td>
<td>0.750</td>
<td>...</td>
<td>1025...</td>
<td>19...</td>
<td>18...</td>
<td>20...</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>22</td>
<td>0.107</td>
<td>0.147</td>
<td>0.482</td>
<td>0.502</td>
<td>0.810</td>
<td>...</td>
<td>2025...</td>
<td>33...</td>
<td>32...</td>
<td>18...</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>23</td>
<td>0.113</td>
<td>0.153</td>
<td>0.406</td>
<td>0.569</td>
<td>0.799</td>
<td>...</td>
<td>2025...</td>
<td>33...</td>
<td>32...</td>
<td>18...</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>24</td>
<td>0.120</td>
<td>0.160</td>
<td>0.505</td>
<td>0.511</td>
<td>0.855</td>
<td>...</td>
<td>2025...</td>
<td>33...</td>
<td>32...</td>
<td>18...</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>25</td>
<td>0.127</td>
<td>0.167</td>
<td>0.512</td>
<td>0.519</td>
<td>0.872</td>
<td>...</td>
<td>2025...</td>
<td>33...</td>
<td>32...</td>
<td>18...</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>26</td>
<td>0.133</td>
<td>0.173</td>
<td>0.457</td>
<td>0.567</td>
<td>0.800</td>
<td>...</td>
<td>2025...</td>
<td>33...</td>
<td>32...</td>
<td>18...</td>
<td></td>
</tr>
</tbody>
</table>

### Table 2 Calculating approach of ontology matching

<table>
<thead>
<tr>
<th>(S_{o} )</th>
<th>(T_{o} )</th>
<th>(R_{s}(O_{s},O_{o}) )</th>
<th>(w(a_{i}) )</th>
<th>(N_{s}(O_{s},O_{o}) )</th>
<th>(H(O) )</th>
<th>(Match(O_{s},O_{o}) )</th>
<th>(\kappa=3/4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>21</td>
<td>0.700</td>
<td>0.10</td>
<td>0.300</td>
<td>0.1642</td>
<td>0.959</td>
<td>...</td>
</tr>
<tr>
<td>16</td>
<td>22</td>
<td>0.008</td>
<td>0.05</td>
<td>0.992</td>
<td>0.1664</td>
<td>0.036</td>
<td>...</td>
</tr>
<tr>
<td>17</td>
<td>23</td>
<td>0.210</td>
<td>0.30</td>
<td>0.790</td>
<td>0.1621</td>
<td>0.768</td>
<td>...</td>
</tr>
<tr>
<td>18</td>
<td>24</td>
<td>1.575</td>
<td>0.35</td>
<td>0.000</td>
<td>0.1663</td>
<td>0.000</td>
<td>...</td>
</tr>
<tr>
<td>19</td>
<td>25</td>
<td>0.358</td>
<td>0.11</td>
<td>0.642</td>
<td>0.1660</td>
<td>1.000</td>
<td>...</td>
</tr>
<tr>
<td>20</td>
<td>26</td>
<td>0.450</td>
<td>0.09</td>
<td>0.550</td>
<td>0.1656</td>
<td>1.000</td>
<td>...</td>
</tr>
</tbody>
</table>

**Definition 2** WS Ontology can be defined six tuples:

\(OWS=<ID, DWSDL, WSS, UDDL, OR, QoS>\), where

- \(ID\) is a ID set of WS Ontology; \(DWSDL\) is fuction description of WS Ontology; \(WSS\) is semantics set of WS Ontology; \(UDDL\) is registration set of WS Ontology; \(OR\) is relation to \(WSS\), which denote description of ontology relation and SWS; \(QoS\) is NOT-function description of service composition, namely Quality of Service.

At this time, we define WS Ontology service composition: \(OWSC=<OWS, \pi>\), where \(\pi\) denote a operation method of \(OWSC\).

Definition 3\(^{(1)}\) We define a Topic Map model as following seven tuples:

\(TM=<T_{c}, T_{a}, T_{r}, T_{p}, R_{s}, R_{p}, R_{g}>\), where

\(T_{c}\) denotes a set of topic types; \(T_{a}\) denotes a set of occurrence types; \(T_{r}\) denotes a set of association types; \(T_{p}\) denotes a set of role types; \(T_{s}\) denotes a set of instance topics; \(R_{s}\) denotes a set of subsumption hierarchy relations; \(R_{p}\) denotes a set of associative relations.

And the same \(T_{c}\) play different \(T_{r}\), namely \(OR \rightarrow TM\).

**Definition 4** Implementation of Topic Maps as following three tuples: \(TM=<TS, AR, RO>\), where

- \(TS\) : adopt subject to express Topic; \(AR\) : adopt Relationship to express Association, \(RO\) denote connection of Occurrence and Topic.

And meet \(TS, AR \rightarrow TAO\).

In connection with BAPF, we design six Topic Maps, and is distributed to three layer, which included Knowledge layer, Information layer and Business layer, showed in Fig.4.

\[\text{© 2009 ACADEMY PUBLISHER}\]
Topic Maps types, hierarchical relationship and incidence

TMj
relationship for

denote. Thus we define similarity of hierarchical
implement process of

Where

© 2009 ACADEMY PUBLISHER

If hierarchical structure of Topic Maps employ a height
distance of ontology and Topic.

Which DistTM, DistName separately denote different
distance of ontology and Topic.

If hierarchical structure of Topic Maps employ a height
h’s m tree structure to description, then root of sub-tree
adopt number a to denote, parent adopt number b to
denote. Thus we define similarity of hierarchical
relationship for TMi and TMj:

\[ \text{Dist} = \frac{\text{Dist}(\text{TM}, \text{Child}, \text{TM}, \text{Child})}{(a_i + a_j)} + \]
\[ \frac{\text{Dist}(\text{TM}, \text{Parent}, \text{TM}, \text{Parent})}{(b_i + b_j)} \]

where

\( \text{Dist}() \) denote distance of child and parent of TMi and
TMj.

And define similarity of incidence relation for TMi and
TMj:

\[ \text{Sim}_{\text{RA}}(\text{TM}_i, \text{TM}_j) = \left| \text{TM}_i \cap \text{TM}_j \right| - \left| \text{TM}_i \cup \text{TM}_j \right| \]

At the time, we define similarity of Topic Maps:

\[ \text{TSim}(\text{TM}_i, \text{TM}_j) = \sqrt[3]{\text{Sim} \cdot \text{Sim} \cdot \text{Sim}} \]

(3) D denotes query of Topic Maps, namely: query BAPF.

(4) F denotes entry point of matching facet of Topic
Maps, such as different user search book. And distance
among Topic Maps, we employ Euclid formula to
calculate, thus, if Topic matching is attached to OR’s
matching, and define membership function of TMi and
TMj is \( \mu_{\text{TM}_i}(x_i) \) and \( \mu_{\text{TM}_j}(x_j) \), then

\[ \text{Di}(\text{TM}_i, \text{TM}_j) = \sum_{i,j} \frac{1}{n} \left( \mu_{\text{TM}_i}(x_i) - \mu_{\text{TM}_j}(x_j) \right)^2 \]

Theorem 1 A web service ontology(WSO) is regarded
as a Subject of Topic.

Description: WSO is a state of semantics service
oriented ontology, services are direct correspondence
OWS.DWSDL, and Topic Map is a group of incidence
relation which connect resource throught Topics. In the
oriented-service of software system, these resources can
come from difference service fuction, however, in the
paper, services are regarded as Topics. Thus

\( \forall \text{WSO} \in \text{WSO}, \exists \text{OWS.DWSDL} \rightarrow \text{WS} \)
\( \forall \text{TM}, \exists \text{Subject identifier:Sif and subject}
indicator:Sit \rightarrow \text{Subject} \)

Root:

\( \forall \text{WSO}, \text{TM} \rightarrow \text{OR} \)
\( \forall \text{OWS.DWSDL} \rightarrow \text{OR}, \text{TM}. \text{Tc} \rightarrow \text{OR} \)
\( \forall \text{TM} \rightarrow \text{OWS.DWSDL} \)
\( \forall \text{TM} \text{(Sif,Sit)} \rightarrow \text{Subject} \)
\( \forall \text{Subject} \rightarrow \text{OWS.DWSDL} \)

Again

\( \forall \text{OWS.DWSDL} \in \text{OWS} \)
\( \forall \text{OWS} \rightarrow \text{Subject} \)

Algorithm 3 Matching Algorithm of Topic Maps

Input: TM \rightarrow TAO;
Output: The best result of Topic Maps matching
1 \( \forall \text{(TM}_i, \text{TM}_f \rightarrow \text{TAO}) \rightarrow \exists \text{(TM}_i, \text{TM}_f \rightarrow \text{OR}) \rightarrow \text{TAO} \)
2 if \( \exists \text{(TM}_i . \text{Tc} \rightarrow \text{OR}) \rightarrow \text{TAO} \)
3 for(int i=1; i<num; i++) // num is given number, and num\( \in \text{natural number} \)
4 \{Find(TMMD and TMF) // Find D and F of Topic Maps;
5 VF = \log_2 \frac{\text{TM}_i \cap / \text{TM}_j \cup }{\text{TM}_i \cap / \text{TM}_j \cup } \rightarrow \text{TMMD} . \text{Di}(\text{TM}_i, \text{TM}_j) // adopt \text{VF} \) as instruction of TMMD selecting;
6 Computers Di(\text{TM}_i, \text{TM}_j);
7 max(TSim(\text{TM}_i, \text{TM}_j));
8 \}
9 return matching result;

Table 3 Matching calculating approach of Topic Maps(showed in Fig.4, different Topic assign difference ID
number, showed in Table3. When users search, make Topic Maps count is 20, namely: TMCount=20)
In the Table 3’s data, according to Fig.4, and we have developed six query Topic Maps, which implement in the open source software TM4J.

**Definition 6** Topic Maps merge:

**Merce:**

\[\forall OWS \rightarrow OWSC;\]

\[\exists TM_{OWS} \rightarrow TAO_{OWS}((OWS \rightarrow OWS):OR) \times TM_{OWS}(OWS \rightarrow OWS: OR) \rightarrow ((TM_{OWS} \rightarrow TMM) \times (TM_{OWS} \rightarrow TMM)) \rightarrow TM_{TAO}.\]

**Theorem 2** Semantics service composition direct influence in Topic Maps merge obviously.

### B. Approach of Service Clustering Description

At present, ontology description language have OWL-S and WSMO(Web Service Modeling Ontology), and we employ OWL-S to describe service clustering in the paper, this is because online bookshop system adopt OWL-S to describe. When implement ontology reasoning, which employ TransitiveProperty, SymmetricProperty, inverseOf, equivalent and other property to connect resource, namely: connect RDF, Finally accomplish ontology reasoning recognition; However, RDF(s) is a schema of description resource property, which assign property to connect resource. And the property is also resource, have property; showed in Fig.5.WSDL describe business fuction, but lack semantics description, therefore employ a special semantics language to improve semantics recognition capability of web services, which effectualy implement service clustering, accomplish service composition.

---

**RDF Description**

```xml
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:soap="http://schemas.xmlsoap.org/soap/"
  xmlns:mashup="http://LocalHost:8080/mashup/"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://LocalHost:8080/mashup/services">
  .......
</rdf:RDF>
```

**Description TM XTM-Based**

**OWL Description**

```xml
<owl:RDF
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://LocalHost:8080/mashup/services">
  .......
</owl:RDF>
```

**WSDL Description**

```xml
<definitions name="QueryService"
  targetNamespace="http://LocalHost:8080/QueryService"
  xmlns:interface="http://LocalHost:8080/QueryService-interface"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/
  xmlns="http://schemas.xmlsoap.org/wsdl/">
  .......
</definitions>
```

---

**Fig.5 Service Clustering Description Approach**
OWL-S\cite{19} is semantics description language of web service based on OWL, which is built services concept, is composed of Profiles, Models and Groundings, and we join Topic Maps in OWL-S, showed in Fig.6 and Code List L1. Employ URI, parameter and OWL in mapping web service, showed in Fig.7. And L1 describe Topic Maps based on OWL-S, in order to adopt Topics to map web services, and implement business function.

**Fig.6 Join Topic Maps in Service Ontology**

When implement service clustering model, we employ RDF Description, OWL Description, WSDL Description and Topic Maps Description based on XML, and adopt their tag relation to recognize semantics and connect resource. Therefore:

**Definition 7** RDF Description employ set $RdfS=\{<R_{i1}, F_{i1}>, <R_{i2}, F_{i2}>, \ldots, <R_{i3}, F_{i3}>\}$ to describe, which $<R_{i1}, F_{i1}>$ denote RDF tag and resource description.

OWL-S Description employ set $OwlS=\{<Ol_{i1}, O_{i1}>, <Ol_{i2}, O_{i2}>, \ldots, <Ol_{i3}, O_{i3}>\}$ to describe, which $<Ol_{i1}, O_{i1}>$ denote OWL-S tag and resource description.

WSDL Description employ set $WslS=\{<Wl_{i1}, BF_{i1}>, <Wl_{i2}, BF_{i2}>, \ldots, <Wl_{i3}, BF_{i3}>\}$ to describe, which $<Wl_{i1}, BF_{i1}>$ denote WSDL tag and business function description.

Topic Maps Description employ set $XtmS=\{<Xt_{i1}, NP_{i1}>, <Xt_{i2}, NP_{i2}>, \ldots, <Xt_{i3}, NP_{i3}>\}$ to describe, which $<Xt_{i1}, NP_{i1}>$ denote XTM tag and resource description.

According to definition 7, we build service clustering model following:

$$Ml=<RdfS, OwlS, WslS, XtmS>$$

And tag set adopt $Ml_{tag}$ to denote, and meet:

$$\bigwedge (RdfS \times OwlS) \bigwedge (RdfS \times WslS) \bigwedge (OwlS \times WslS) \bigwedge (OwlS \times XtmS) \bigwedge (WslS \times XtmS)$$

The research, we proposed a semantics ant colony algorithm:SACA to optimize and implement service clustering model on basis of ant colony algorithm, which operate $Ml$, therefore, we set four ant colonies, every colony have same or difference ants, then according to ant a hangover from the pheromone for a optimum path, namely: the path meet resources of service clustering, showed in Fig.8. And, according to similarity of tag string in the four fields, we build (*) model.

$$f(String_i, x) = \max \{ \sum_{String_j, similiar(s)} \left[ 1 - \frac{Sim_{max}(String_i, String_j)}{\alpha(1+(v-1)/\nu_{max})} \right] \}$$

Where
\[ x = 1, 2, 3, 4, \text{ which separately denote a set in the } M_l; \]

\[ \text{String}_i, \text{String}_j \text{ denote tag string in the } M_l; \]

\[ \text{Sim}_{\text{tag}} \text{ denote tag’s similarity in the } M_l; \]

\[ \text{domain}(x) \text{ express difference tag from different field, } \alpha \text{ denote similarity parameters, } \nu \text{ and } \upsilon_{\text{max}} \text{ separately denote ant velocity and maximum velocity.} \]

\[ \text{When meet } \]

\[ Y = \max \{ \text{Sim}(\text{So}, \text{To}), \text{Match}(\text{O}, \text{O}), \text{TSim}(\text{TM}_l, \text{TM}_j) \}, \exists \]

\[ \text{Compare}(f(\text{String}_1), 1) = \lambda_1; \]

\[ \text{Compare}(f(\text{String}_2), 2) = \lambda_2; \]

\[ \text{Compare}(f(\text{String}_3), 3) = \lambda_3; \]

\[ \text{Compare}(f(\text{String}_4), 4) = \lambda_4. \]

\[ \text{Which } \lambda_i (i = 1, 2, 3, 4) \text{ is incidence matrix.} \]

\[ \text{And set values following:} \]

\[ \text{Compare}(f(\text{String}_i, x)) = \begin{cases} 
0 \text{(disaffinity)} \\
1/2 \text{(similarity)} \\
1 \text{(difference)} 
\end{cases} \]

\[ \text{If } \text{Sim}_{\text{tag}}(\text{String}_i, \text{String}_j) \text{ employ KMP algorithm to calculate string matching} \]

\[ \text{next}[y] = \begin{cases} 
0 \text{(y = 1)} \\
\text{max} \{k | 1 < k < y \land (P_1 \cdots P_{k-1} = P_{y-k+1} \cdots P_{y-1}) (k \neq \Phi) \lor \text{I(Other)} \} 
\end{cases} \]

\[ \text{Then } \text{Sim}_{\text{tag}}(\text{String}_i, \text{String}_j) = \frac{\text{next}[y]}{|p|}, \text{ where} \]

\[ |p| \text{ denote String, length.} \]

**Algorithm 4 KMP algorithm**

Input: \( M_l \)

Output: \( \text{Sim}_{\text{tag}}(\text{String}_i, \text{String}_j) \)

1. \( \exists \text{Some tags} \rightarrow M_l; \)
2. IF \( \text{Sim}(\text{So}, \text{To}) \& \& \text{Match}(\text{O}, \text{O}) \& \& \text{TSim}(\text{TM}_l, \text{TM}_j) \)
3. \{FOR \( i = 1; r < |Z_r|; r++) // Z’Value calculate method \}
4. \{int Index_KMP(\text{SString}_i, \text{SString}_j, \text{int pos, int CalS}) \}
5. \{i(\text{T} = \Phi) \& \& (\text{pos} \leq \text{StrLength}(\text{S})) \}
6. \( i = \text{pos} + 1; \)
7. \( \text{while}(i < |\text{T}| \& \& j < |\text{T}|) \{ \}
8. \( \text{IF}[j = 0 \& |\text{T}| = |\text{T}|] \{ \+)
9. \( \text{else} j = \text{next}[j]; \}
10. \( \text{IF}(j > |\text{T}|) \text{return } i = |\text{T}|; \)
11. \( \text{else return}(0); \)
12. \( \text{CalS} = \text{StrLength}(\text{S}); \}
13. \{\text{get next}() // \text{SString}_i, \text{int &next[]} \}
14. \{i = 1; \text{next}[i] = 0; j = 0; \}
15. \{\text{while}(i < |\text{T}|) \{ \}
16. \( \text{IF}[j = 0 \& |\text{T}[i] = |\text{T}[i]] \{ \+)
17. \( \text{else } j = \text{next}[j]; \}\}\}

If probability of string \( \text{String}_i \) tag \( r \) for \( \text{String}_j \) matching is \( 1/4P_r \), then not-matching probability is \( (1 - 1/4P_r) \).

\[ \text{String}_i \text{ for String}_j \text{; not-matching expectation is} \]

\[ E = \frac{1}{4} P_r \prod_{r=1}^{n} (1 - \frac{1}{4} P_r) x_r, \]

\[ Z = \mid Rds_{\text{tag}} \mid \text{or } \mid \text{Owl}_{\text{tag}} \mid \text{or } \mid \text{Wdl}_{\text{tag}} \mid \text{or } \mid \text{Xms}_{\text{tag}} \]

(2)

When implement tag clustering, firstly design program \( \zeta \) of navigation and positioning of Topic Maps oriented ontology relation, in order to build service clustering model, which not-matching expectation is minimum value, is build following (3) and (4), thereby improve target matching efficiency of \( M_l \).

\[ E(\zeta) = \min \left\{ \sum \sum \left[ \frac{1}{4} P_r \prod_{r=1}^{n} (1 - \frac{1}{4} P_r) x_r \right] \right\} \]

(3)

\[ \begin{aligned} 
A, B & = \text{Rds} | \text{or } \text{Owl} | \text{or } \text{Wdl} | \text{or } \text{Xms} \\
Z & = \text{Rds}_{\text{tag}} | \text{or } \text{Owl}_{\text{tag}} | \text{or } \text{Wdl}_{\text{tag}} | \text{or } \text{Xms}_{\text{tag}} \\

P_y & = 1 - f(\text{String}_i, \text{String}_j) \]

(4)

s.t.

\( \begin{cases} 
(i) \sum_j x_{ij} = 1, r = 1, 2, \cdots, Z \\
(ii) \sum_j x_{ij} = 1, r = 1, 2, 3, 4 \end{cases} \)

When implement ant colony algorithm to optimize, assume ants are \( m \), and assign differently ants \( m (k) (m_1 + \cdots + m_m) \) to set in the \( M_l \), then assign \( m \) to difference tag. If transition probability of ant is (5) and (6)[21], namely: probability of selection resource \( j \) for tag \( r \):

\[ P_{p_j}(r, j) = \begin{cases} 
\arg \max_{\text{allowed}_k} \{ \tau(r, u) \cdot \eta^\alpha (r, u) \} & \text{if } q \leq q_0 \\
0 & \text{else} \end{cases} \]

(5)

\[ J = \begin{cases} 
\frac{\tau(r, j)^{\alpha} \cdot \eta^\beta (r, j)}{\sum_{\text{allowed}_k} \tau(r, j)^{\alpha} \cdot \eta^\beta (r, j)} & \text{if } j \in \text{allowed}_k \\
0 & \text{else} \end{cases} \]

(6)

And build semantics pheromone of \( r \rightarrow j \) following (7) for navigation and positioning of Topic Maps.

\[ \eta(r, j) = \left\{ \begin{array}{ll}
\text{max} \{ \text{Sim}(\text{So}, \text{To}) \cdot \text{Match}(\text{O}, \text{O}) \} \\
\text{TSim}(\text{TM}_l, \text{TM}_j) 
\end{array} \right\} \]

(7)

At this time, if ants \( k \) have assigned tag to build \( r \) in the \( E(\zeta) \), then adopt \( tabu_k \) to record tag that didn’t match again, and according to constraint condition ( i ), ( ii ) adjust \( E(\zeta) \), and adopt \( \text{allowed}_k = \{ \text{ML}_{\text{tag}} \cdot \text{tabu}_k \} \) to express tag set assigned.

1) When ants \( k \) solve practicable solving \( E(\zeta) \), according to (8) formula implement adjustment of local pheromone updated:

\[ \tau(r, j) = (1 - \rho) \cdot \tau(r, j) \cdot \text{Sim}_{\text{tag}}(\text{String}_i, \text{String}_j) \]

\[ + \rho \tau_0, 0 \leq \rho \leq 1 \]

Where

\[ \tau_0 = 1/(m \cdot E(\zeta)) \]

2) According to (9) formula implement adjustment of global pheromone updated:

\[ \tau(r, j) = (1 - \theta) \cdot \tau(r, j) \cdot \text{Compare}(f(\text{String}_i, \text{x})) \]

\[ + \theta E(\text{opvalue}) \]

\[ 0 \leq \theta \leq 1 \]

Where

\[ E(\text{opvalue}) \] denote current optimum expectation value.

© 2009 ACADEMY PUBLISHER
Algorithm 5 Semantics ant colony algorithm

Step 1 The initialization of tag scope every description language;

Step 2 To calculate \( f(String_i, x) \), and calculate \( \max[Sim( So, To), Match( O_i, O_j), TSim( TM_i, TM_j)] \), which separately obtain the best data of ontology relation similarity, matching and the best data of Topic Maps similarity.

Step 3 Employ KMP algorithm to implement tag matching every field, and obtain a group of effective matching value, and adopt (2) to calculate expectation value of not-matching tag.

Step 4 Adopt (3) to calculate tag matching program of waiting assignment, and obtain \( Pij \) in the (4).

Step 5 According to constraint condition evaluate (2) and (3), and obtain matching program to assign ants.

Step 6 Adopt (5) to calculate \( Pij (r, j) \), which implement ants to assign, and accomplish difference navigation and positioning of tag, at the same time, adopt (6) to update optimum path of ants.

Step 7 Adopt (7) to accomplish pheromone of ants optimum path to update. And (8) is a key of semantics service clutering.

Step 8 What evaluate \( E(\xi) \) again, if don’t meet demand of online bookshop system, then according to Step9 and Step10 implement local and global pheromone updated.

Step 9 Adopt (8) to implement local pheromone updated.

Step 10 Adopt (9) to implement global pheromone updated.

Step 11 If meet demand, then end, else return Step2.

When implement semantics ant colony algorithm, we set \( m=50(m_1=12, m_2=15, m_3=8, m_4=15) \), \( q_0=0.7, \alpha=1.5, \beta=2, \rho=0.2, \theta=0.05 \) to experiment in the online bookshop system, which compared SACA with ACA, and adopt 20 times to experiment, iterative operation 2000 times, showed Fig.9. Thereby, SACA reach optimum solving at 600 times, and ACA reach optimum solving at 1600 times.

Experiment results showed in Table 4, Fig.10 and Fig.11.

Table 4 Adopt BAPF query to make experiments

<table>
<thead>
<tr>
<th>Search item</th>
<th>Recall</th>
<th>Precis</th>
<th>Effici</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bookname</td>
<td>95.653</td>
<td>96.843</td>
<td>94.500</td>
</tr>
<tr>
<td>Author</td>
<td>97.013</td>
<td>95.433</td>
<td>96.022</td>
</tr>
<tr>
<td>Press</td>
<td>94.788</td>
<td>95.578</td>
<td>94.778</td>
</tr>
<tr>
<td>Full-text</td>
<td>94.001</td>
<td>95.333</td>
<td>95.022</td>
</tr>
</tbody>
</table>

Fig.9 Result compared SACA with ACA

V. EXPERIMENT RESULT

Experiment results mainly make experiments from recall rate of web service ontology oriented ontology relation for navigation and positioning of Topic Maps, precision of Topic Maps, optimum calculation efficiency of SACA, and evaluate feasibility and advantage. Therefore, the experiment platform employ J2EE, and other open softwares which protégé[22]: ontology maker and knowledge acquisition, Axis[23]: development web service, TM4J[24]: development Topic Maps, saxon[25]: process XML, and so on. And we develop three program packages based on JAVA, which includes ontology relation package: Com.OntRelaTopicMapic.OntologyRelatin.*, navigation and positioning of Topic Maps package: Com.OntRelaTopicMapic.TopicMapicNP.*, semantics ant colony optimum package: Com.OntRelaTopicMapic.SACA.*. Finally, common integration make experiment in the Eclipse. Experiment system adopt online bookshop system, and experiment data make 30 services, 20 ontologies and 6 Topics. At the same time, set four counter, which separately count ontology(include source ontology and target ontology), Topic Maps, services(include request services and response services), service composition, thus define:

Recall rate of ontology relation:

\[ \text{Recall} = \frac{|So \cap To|}{|So \cup To|}; \]

Precision of Topic Maps:

\[ \text{Precis} = \frac{|OWS \rightarrow \text{Subject}|}{|TM|}; \]

Optimum efficiency of SACA:

\[ \text{Effici} = \frac{|String_i \cap String_j|}{|MI|}. \]

Fig.11 efficiency charge of BAPF along with delay time charge
Fig. 12 Efficiency compare new method with traditional method

VI. CONCLUSION

In the paper, we research condition of navigation and positioning of Topic Maps for service clustering model, therefore, firstly build software hierarchy of navigation and positioning. Secondly, research ontology relation and describe language relation among RDF, OWL-S, WSDL and XTM; on basic of which build service clustering model, and build a semantics ant colony algorithm to optimize. In connection with these problems we define computational method, and give concrete algorithm and experiment data. Finally, employ online bookshop system to make experiment platform based on J2EE, and define recall rate of ontology relation, precision of navigation and positioning of Topic Maps and optimum efficiency of SACA. But alone with ontology and Topic Maps apply in all fields, and software system based on service is maturing, in this times, if ontology and Topic Maps apply in all fields, and software system based on service is maturing, in this times, if optimize the approach, efficaci ously increase precision of service composition, thereby enhance user’s demand. Especially we proposed semantics ant colony algorithm, which would improve semantics recognition and extraction capacity of service composition.

ACKNOWLEDGMENT

The Natural Sciences Foundation of Sichuan Provincial Education Bureau, China (No. 07ZC002).

REFERENCES

[17] ZHOU Xiang-bing, YANG Xing-jiang, XIANG Chang-cheng. SOA Component Localizer Mechanism and Algorithm Based on Ontology, Systems Engineering and Electronics, 2008, 30(12):11-17