

Building Requirements Semantics for Networked Software Interoperability

Bin Wen, Keqing He, Jian Wang

State Key Lab of Software Engineering, Wuhan University, Wuhan, China.
Email: binwenwebb@gmail.com

Received November 9th, 2009; revised December 1st, 2009; accepted December 20th, 2009.

ABSTRACT

Naturally, like the web, integrated software systems in Internet will have to be distributed and heterogeneous. To improve the interoperability of services for SAAS, it is crucial to build requirements semantics that will cross the entire lifecycle of services especially on requirements stage. In this paper, a requirements semantics interoperability extending approach called Connecting Ontologies (CO) that will act as semantics information carrier designing to facilitate the requirements identification and services composition is proposed. Semantic measurement of Chinese scenario is explored. By adopting the approach, a series of tools support for transport domain are developed and applied based on CO and DPO (Domain Problem Ontology) to enforce requirements engineering of networked software efficiently.

Keywords: Networked Software, Requirements Semantics, Requirements Engineering, Connecting Ontologies

1. Introduction

Ideally, users can access services based on their requirements without regard to where the services are hosted or how they are delivered. Various computing paradigms have promised to deliver IT as services including grid computing, P2P computing, and more recently Cloud computing. The latter term denotes the infrastructure as “Cloud” from which businesses and users are able to access application from anywhere in the world on demand. Thus, the computing world is rapidly transforming towards developing software for millions of consume as a service, rather than to run on their individual computers [1].

The development of networked software has emerged varied forms and definitions. One is pervasive computing, such as grid computing, e-science, and transparent computing, which focus on resource sharing. Another category is cloud computing based on SAAS (software as a service) and related studies include SOA, Web Service, Semantic Web Service etc. SAAS and virtualization of hardware and software are two main features for Cloud computing. Networked software that this paper refers to [2] belongs to the second sort that is complex information system based on Internet towards service computing. Distribution, autonomy, opening and heterogeneity are its basic features and stakeholders to be faced having various sorts and interests. Typically, supporting diversified, personalized and dependable services to improve

user QoE (Quality of Experience) is the highest goal.

Requirements engineering (RE) is crucial to the success of software engineering, especially for networked software, and considering issues mainly include dynamic elicitation and analysis, evolution modeling, requirements management and model verification of user requirements and so on. Requirements modeling methods mostly are classified as structural requirements modeling and object-oriented requirements modeling according to paradigm, and both of them can deal with functional and nonfunctional requirements analysis. Now the typical software RE approaches are goal-oriented, ontology-oriented, scenario-based, problem framework, pre-requirements analysis based on domain modeling, document driving and aspect-oriented method [3].

The most widely significant approaches for networked software RE are goal-oriented and pre-requirements analysis based on domain ontology approach. Goal-oriented approach concentrates on analysis and modeling of early requirements so as to help developer understand the motivation and expectation for various roles, and involves the identification and analysis of functional and nonfunctional requirements goal. At present software RE is switching from object-oriented to goal-oriented [4,5], whereas goal-oriented approach has produced commercial products for tool supporting, for instance Cediti goal analyzer: Objectiver. Accordingly goal-oriented requirements analysis has become the hot spot of the

studying of RE.

Virtually, pre-requirements analysis based on domain modeling [6,7] is the process of requirements analysis based on domain-level ontology knowledge. The issue of ODE method based ontology [8] only acquires domain conceptual knowledge especially, but it ignores the modeling for task and functional knowledge.

All the above-mentioned requirements modeling methods consider only for object-oriented development. The applicability and feasibility of those approaches for service-oriented computing must be reconsidered. Regarding the features for service computing, role, goal, process and service, the four fundamental elements can be used to modeling for the users' truly intentions of networked software. A meta-modeling framework containing the four fundamental elements, namely RGPS [9], is presented for conducting synergy and ordered structure requirements specification from disordered requirements information. Furthermore, choosing ontology meta-modeling [10] and encapsulating domain reusable core services asset, O-RGPS (Ontology-RGPS) meta-model proposal [2] is also put forward (see Figure 1).

Based on O-RGPS requirements meta-model framework, user requirements can be described from different angle, level and granularity in order to form domain requirements asset and store as OWL for reuse.

Interaction and collaboration of networked software is a restricted semantic interoperable issue on essence. Then, how to constrain and extend the semantic interoperability in the process of self-organization and action emergence for the distributing services resource? How to categorize the structure of interoperability? How to satisfy stakeholders' requirements?

Regarding the above issues, this paper proposes an

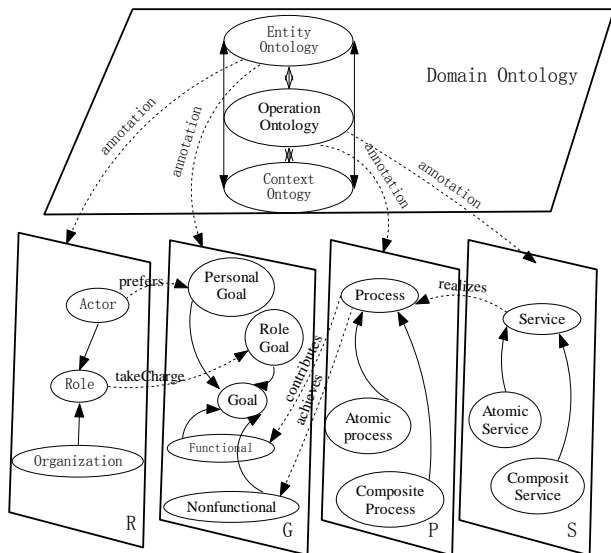


Figure 1. Domain asset customizing based-on O-RGPS

requirement semantic interoperable extending approach for networked software based on connecting ontologies (CO) and furnishes the unified and dynamic semantic information carrier for service aggregating and evolution modeling.

The rest of the paper is organized as follows: Section 2 explores software RE method based on domain ontology and requirements asset; furthermore, provides formal definition and aggregating method of connecting ontologies, and presents the related algorithm and integrating environment design for interoperable extending of networked software requirements semantics; Section 3 summarizes the related cutting-edge work in the research community; at the last, we conclude the paper and survey the future work.

2. Connecting Ontologies for Networked Software

Networked software system includes the overall architecture and goal software system that can embody dynamic property of the architecture. Goal software system is composed of services, whereas service resources distribute in network and are loosely coupled, dynamic binding and permit various levels of semantic interoperability.

Since service resources are dynamically distributed, for the sake of acquiring requirements knowledge from multi-domain service resources, disseminated ontology registry repositories in network require ontology encapsulation which is unified annotation of service with respect to requirements semantic. Ontology registry repositories will accord with ISO meta-model framework MFI (ISO/IEC SC32 19763) [11] that we participate. Requirements are gained by requirements acquiring & analysis (RAA) approach, and Requirements Sign Ontology (RSO, Definition 11, similar to process specification or workflow of application) is generated. Based on RSO, published ontologies of requirements semantic for available services are dynamic found and matched in network. Matched ontologies and RSO form ontologies group that is loosely coupled connected and dynamic generated, named Connecting Ontologies (CO). Stated in Figure 2, is requirements modeling approach for networked software based on CO. In ontology level, requirements semantic are dynamic acquired with semantic extending and matching. Furthermore, initial requirements model is generated by reusing multi-domain requirements asset. CO is the process of dynamic generating and continuous evolving, as stakeholders' requirements are uninterruptedly changed and loosely coupled for multi-domain requirements asset.

2.1 Domain Ontology Based on Description Logic

In the line of computer, ontology is explicit representation and description of conceptualization objects.

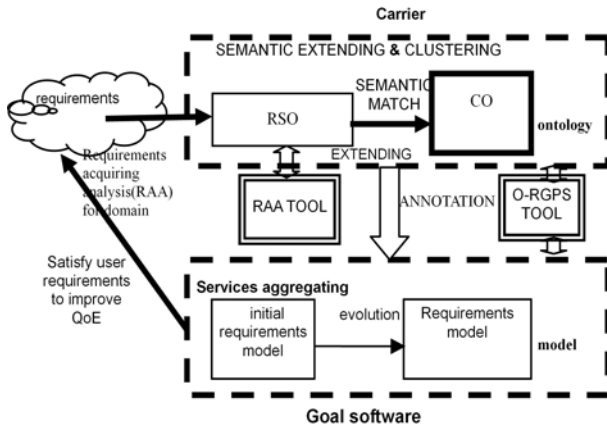


Figure 2. Requirements modeling for networked software based-on connecting ontologies

Ontology can also be used for software RE as requirements representation and carrier. At the same time, since reusability of broad-spectrum ontology is relatively hard, the principal application direction of ontology for software requirements should be domain-oriented and problem-oriented.

Firstly, this section gives the definition of domain ontology based on description logic and other related definitions. Next section will apply these definitions. Then requirements elicitation based on domain ontology and requirements asset is designed and implemented.

Definition 1 (Domain Ontology based on description logic). Domain Ontology is expressed as $DO = \langle D, C, T, A, LH \rangle$, where D represents domain; C represents a set of concepts; T represents TBox; A represents ABox; and LH represents lattice hierarchy of concepts.

Definition 2 (Relation Triple among concepts). For domain ontology $DO = \langle D, C, T, A, LH \rangle$ with $c_p, c_q \in C$ and relation $r(c_p, c_q) \in T$, if c_p and c_q satisfy (1) $c_p \sqsubseteq c_q$ (2) $c_p \sqsubseteq \exists r.c_q$ (3) $c_p \sqsubseteq \forall r.c_q$, and \sqsubseteq is concepts inclusion relation, then $\chi = \langle c_p, r, c_q \rangle$ represents relation triple between c_p and c_q .

Definition 3 (Semantic Association). For two relations $\chi_1 = \langle c_{p1}, r_1, c_{q1} \rangle$ and $\chi_2 = \langle c_{p2}, r_2, c_{q2} \rangle$, $\chi_1 \square \chi_2$ denotes semantic association between χ_1 and χ_2 where $\chi_1.c_{q1} = \chi_2.c_{p2}$.

Definition 4 (semantic association path). For a set of relations $X = \{\chi_i \mid \chi_i = \langle c_{pi}, r_i, c_{qi} \rangle, i = 1, 2, \dots, n\}$ and relation triples $\chi_s = \langle c_{ps}, r_s, c_{qs} \rangle$, $\chi_d = \langle c_{pd}, r_d, c_{qd} \rangle$, $\chi_s, \chi_d \in X$, if $\exists m \leq n-2$, for $\forall \chi_{i+j} \in X, j = 1, 2, \dots, m$, then DO have a semantic association path in X from χ_s to χ_d where semantic associations $\chi_s \square \chi_{i+1}, \chi_{i+1} \square \chi_{i+2}$ and

$\chi_{i+m} \square \chi_d$ exist, namely semantic association path between concept c_{qs} and c_{qd} .

Definition 5 (concept semantic depth, Depth). Apart from the class for itself, the meaning of ontology concept is also described by the associated classes, namely concept semantic depth. To calculate semantic depth, let the Depth of ontology root concept is zero, if the Depth of concept c , $Depth(c)$, is I , then the Depth of its father concept (if existed) is $I-1$ and the Depth of its child concept (if existed) is $I+1$.

2.2 Connecting Ontologies

Connecting ontologies based on semantic matching of multi-domain requirements asset only utilize local or part of ontologies registry repositories for services. Modularization is an important technique of ontology reuse for services. Different researchers have different definitions or designations including segment, module, view or sub-ontology etc. The paper adopts sub-ontology [12] notion. Some definitions and algorithms are presented as follows.

Definition 6 (sub-ontology). For domain ontology $DO = \langle D, C, T, A, LH \rangle$, a sub-ontology sub- $Onto$ consists of 5 elements $\langle C_{sub}, T_{sub}, A_{sub}, LH_{sub}, I \rangle$, where C_{sub} represents the set of sub- $Onto$ concepts which denotes the context of sub-ontology; $|C_{sub}| < |C|$; there exist semantic association or semantic association path in C_{sub} ; $T_{sub} \subseteq T$, $A_{sub} \subseteq A$ represent sub- $Onto$'s local knowledge base for T_{sub} , A_{sub} ; LH_{sub} represents lattice hierarchy of concepts; and I represents index pointer towards DO . If sub-ontology = DO or sub-ontology have nondeterministic domain, then I is nil.

Definition 7 (sub-ontology space in same source). For $DO = \langle D, C, T, A, LH \rangle$, sub-ontology space in same source Space represents $\{ \langle sub-Onto_b, B, DO \mid \forall sub-Onto_b, I = DO, B \in Index \rangle \}$

2.2.1 Algorithm: Sub-Ontology Extracting Algorithm

For $DO = \langle D, C, T, A, LH \rangle$, $\langle CON, n, DO \rangle$ is the input of sub-ontology extracting, where $CON = \{con_1, con_2, \dots, con_k\}$ represents a set of concepts which will be matched; DO represents father ontology; n represents the depth of travel. Based on [12], we can get sub-ontology extracting algorithm. The outcome of the algorithm is a sub-ontology sub- $Onto$.

Sub-ontology extracting algorithm can be seen from Algorithm 1.

Attentively, semantic similarity matching can be described in details: for any two concepts $C1$ and $C2$, assuming string $S1$ and $S2$ is the name of $C1$ and $C2$ respectively. Firstly, lexical analysis that preposition, conjunction, pronoun and interjection are cancelled is carried out for two strings, whereas continuous and meaning words are reserved. Strings $S1$ and $S2$ will be transferred

Algorithm 1 sub-ontology extracting algorithm.

```

INPUT: <CON, n, DO>
OUTPUT: sub-Onto
1: i<-1
2: Repeat
3:   get i-th concept coni from CON
4:   remove coni from CON
5:   if exist a concept c satisfy concept coni semantic
   similarity match constraint in DO
6:   then
7:     do breadth-first traversal from c through rela-
   tion in DO
8:     if another concept c' in DO is reached in
   traversal
9:     then
10:    add the triples along the traversed path from c
   to c' into Tsub
11:    add c to a set Csub
12:    add c' to a set Csub
13:    remove c' from CON
14:    end if
15:    stop when traversal up to a depth of n;
16:  end if
17:  i<- i+1
18:  Until CON is empty or i>=k
19:  m<- |Csub|
20:  If m<k then return nil
21:  end if
22:  get the set of propertys Tsub from DO for all the
   concepts in CON
23:  get the set of individuals Asub from DO for all the
   concepts in CON
24:  get the extracting

```

Algorithm 2 sub-ontology merging algorithm.

```

INPUT: onto-set={Sub-Onto1, Sub-Onto2,..., Sub-Onton}, n >=2
OUTPUT: Merge(onto-set)
1: add all concepts in Sub-Onto1.Csub from onto_set
   to a new set C
2: add all items in Sub-Onto1.Tsub from onto_set to a
   new set T
3: add all items in Sub-Onto1.Asub from onto_set to a
   new set A
4: i<- 2;
5: repeat
6:   get i-th Sub-Onto Sub-Ontoi from onto_set
7:   for each concept CK in Sub-Ontoi.Csub
8:     if Ck not in C then
9:       add Ck to C
10:    end if
11:  end for
12:  for each item Mj in Sub-Ontoi.Tsub
13:    if Mj not in T then
14:      add Mj to T
15:    end if
16:  end for
17:  for each item Aq in Sub-Ontoi.Asub
18:    if Aq not in A then
19:      add Aq to A
20:    end if
21:  end for
22:  i<- i+1
23:  until i>=n
24:  get the corresponding Onto = <C,T,A,LH,nil>

```

to $\langle S1_{w1}, \dots, S1_{wn} \rangle$ and $\langle S2_{w1}, \dots, S2_{wm} \rangle$. For any words $S1_{wi} \in \langle S1_{w1}, \dots, S1_{wn} \rangle$ and $S2_{wj} \in \langle S2_{w1}, \dots, S2_{wm} \rangle$, we can calculate two words' similarity $\text{similarityScore}(S1_{wi}, S2_{wj}) = \text{wst.lookup}(S1_{wi}, S2_{wj})$. This similarity is acquired by looking up similarity table which is generated by experts in matching computing by using words association tool (such as WordNet) in advance. If $n \leq m$, then for $S1_{wi}$, we can find $S2_{wj}$ in accordance with maximum similarity, namely $\text{matchscore}(S1_{wi}, S2_{wj}) = \text{similarityScore}(S1_{wi}, S2_{wj})$. Finally similarity between two concepts is $\text{matchscore}(C1, C2) = \text{Sum}(\text{matchscore}(S1_{wi}, S2_{wj}))/n$.

2.2.2 Algorithm: Sub-Ontology Merging Algorithm

For a set of sub-ontology, onto-set consists of $\{\text{Sub-Onto}_1, \text{Sub-Onto}_2, \dots, \text{Sub-Onto}_n\}$, $n \geq 2$, and the outcome of the algorithm generates a sub-ontology $\text{Onto} = \text{Merge}(\text{onto-set})$.

Sub-ontology merging algorithm can be seen in Algorithm 2 in details.

Definition 8 (maximum self-contained sub-ontology on concepts). For a set of concepts C which will be matched and a sub-ontology extracting algorithm, the last sub-ontology represents maximum self-contained sub-ontology on concepts C , where the set of concepts in the extracted sub-ontology unable to increase along with addition of travel depths to cease the extracting process.

Definition 9 (domain requirements ontology). For

convenient requirements acquisition and matching, domain requirements ontology is a special DO which only have two concepts with semantic depth $\text{Depth}=1$ in the sets of concepts: Operation denotes requirements verb concept and Entity denotes requirements noun concepts. Maximum self-contained sub-ontology of the set of operation is called operation ontology and maximum self-contained sub-ontology of the set of entity is called entity ontology for domain requirements ontology.

Definition 10 (Domain Problem Ontology, DPO). Domain Problem Ontology (DPO) represents as $\text{Merge}(\bigcup_{\text{asset}_i \in \text{RGPS}} (\text{Extracting}(P, \text{Dep}, \text{asset}_i), \text{index}_i))$, where P represents a set of problem's concepts; Dep refers to travel depth; RGPS represents domain-customized asset based RGPS; index_i represents source ontology index with respect to matched problem concepts of RGPS asset.

Note that the Problem is a specific application context, for example travel is a Problem for traffic domain.

Definition 11 (Requirements sign ontology, RSO). Requirements sign ontology RSO consists of 3 elements $\langle \text{DSorl}, \text{Concept}, \text{Control} \rangle$, where DSorl represents input in domain requirements service language; C represents the set of extracting concepts from DSorl ; $\text{Concept} \supseteq \text{DPO}.C$; Control represents control structure among matched service ontologies mainly including sequence, choice, split-union, any order, cycle.

CO are a sub-ontologies set with different sources in which involve dynamic finding and matching ontologies of published services, and RSO serves as mediator and

conducts the process of generating CO for service-oriented requirements.

Definition 12 (connecting ontologies, CO). Connecting ontologies (CO) consists of $\langle RSO, DPO, \text{Mapping-onto-Set} \rangle$, where RSO represents requirements sign ontology; DPO represents problem-oriented domain problem ontology; Mapping-onto-Set represents matched sub-ontology set of different source.

Based on sub-ontology extracting algorithm and the direction of RSO, requirements semantic of CO firstly execute the matching for DPO. The rest of unabsorbed parts by DPO for CO run ontologies finding and matching from multi-domain services in network to satisfy requirements semantic for stakeholders. General speaking, the matched ontologies always denote some sub-ontologies of ontologies with respect to multi-domain services, and they are semantically matching with RSO, namely O_i ($i=1\sim n$). Then, as seen in Figure 3, connecting sub-ontology O_0 of DPO and sub-ontologies O_i of ontologies for multi-domain services according to RSO that acts as the center will dynamically generate CO. Accordingly, dynamically generated CO not only contain O_0 which is domain-oriented and tightly couple with DPO, but also do it include some services ontologies O_i for different domain i and loosely coupling with RSO. A few of unmatched services based CO will be solved by customizing manufacture.

2.3 Domain Problem Ontology

According to Definition 10, Domain Problem Ontology (DPO) is really a composite sub-ontology in terms of problem by extracting from Domain Ontology and RGPS requirements assets that express as OWL format. DPO is very important in the creating process of CO and acts as problem vision for CO. Creating CO firstly need adopting and matching with DPO, so the quality of DPO is crucial for the success of appropriate and preferred match regarding the contract ontology (i.e. CO) of all circles for software web clustering.

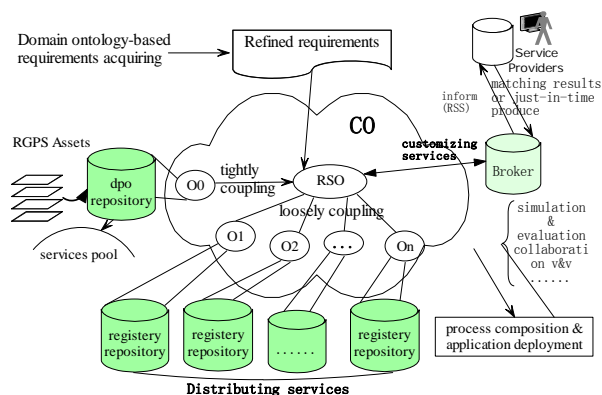


Figure 3. Connecting ontologies

We believe that: 1) semantic distance is only necessary and fundamental measure method for semantic interoperability capability; 2) for semantic interoperability measurement, semantic distance is not sufficient condition; 3) not only do semantic interoperability capability relate to similarity but also tightly associate with the contracted standard (i.e. CO) for both sides and really CO is sufficient condition for interoperability.

Generating DPO can adopt two fashions: semi-automated method directed by domain experts and fully automated method. We have realized the first fashion in our domain modeling tool designing to acquiring RGPS assets and automated fashion is now designing and optimizing. For automated fashion, we considered problem as follows: 1) the relation between DPO extracting depth (traverse depth) and CO matching degree with RSO; 2) the relation between DPO extracting depth (traverse depth) and extracting time cost.

For the above issues, we work out an experiment for evaluating these relations.

2.3.1 Experiment Design

Regarding low-scale Transport ontology (concepts number below 200) and OWL formatted R, G and P, experiment will evaluate the capability between DPO extracting depth associated with CO matching degree and time spending. Firstly, using Algorithm 1, 4 ontologies including Transport ontology, R, G and P [9], will be executed in accordance with the word “travel” and its synonym and outcome will be merged to generate DPO by Algorithm 2. RSO can be obtained by requirements acquiring tool [13] that we have implemented. Matching degree is manually achieved by domain experts between RSO and DPO.

2.3.2 Result Evaluation and Discussion

In the simulate experiment, the initial value of DPO extracting depth is 1. Through changeable extracting depth, we can get different matching degree and time cost for different depth value in order to analysis the influence of depth for entire CO generating process. Figure 4 is the result for different depth value.

According to the result, higher depth value will have higher matching degree with RSO. When DPO extracting depth is higher, the scale of DPO sub-ontology is also biggish correspondingly. Considering the principle of space locality, the reuse probability of DPO will evidently increase to enhance the matching degree with RSO. But higher depth value will lead to more time spending for creating DPO. At the same time, matching degree do not obviously enhance when the depth value increase from 6 to 8. It shows that only increasing depth value is not always efficient for improving matching degree. Since adopting sound depth value is very important for DPO to optimize the matching performance. The time

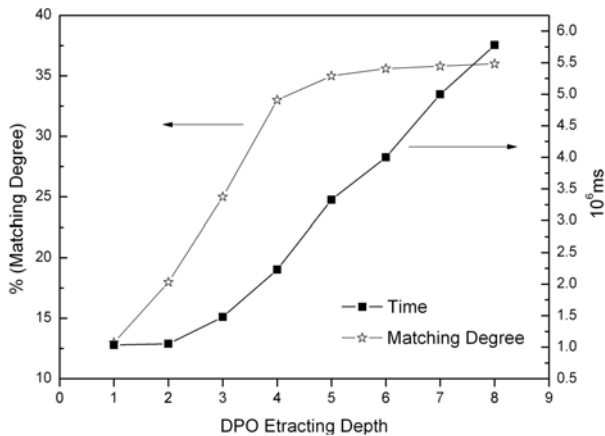


Figure 4. DPO extracting simulation result

cost of the experiment is higher than large-scale single ontology extracting in [12] because the experiment adds the spending of merging process.

The drawback of this experiment is low-scale original ontology, so future work will execute on large-scale ontologies to obtain valuable result for real-world.

2.4 Interoperability Extending Integrating Environment for Requirements Semantic Based CO

Regarding travel problem in urban traffic domain, simulation tests for acquiring requirements semantic based on CO [14] have shown that the semantic interoperability extending approach provides semantic information carrier for networked software and furnishes semantic goal for on-demand service aggregating. But now both RSO perfection and CO dynamic generating mainly rely on manually participating and customizing by requirements analyzers frequently, and quantitative measurement is absent for denoting semantic distance and interoperability level. Farther studies are listed as follows: 1) interoperability extending integrating environment for requirements semantic; 2) measurement system for requirements semantic interoperability.

2.4.1 Requirements Semantics Distance for Chinese Context

Now, software requirements semantics mainly adopts ontology encapsulation style, and requirements matching will reduce to similarity comparing among entities. Basic elements of entity include concept, relation and instance. Main measurement feature of concept are: concept name (no semantics, only consider linguistic and literal similarity, such as some distance formula [15]), concept semantics similarity, concept structure. Main measurement feature in relation involve property name, domain and range. Instance is auxiliary measurement for concept.

Semantics distance refers to a measurement of seman-

tics similarity or association between two semantic entities. Semantic entities involving this paper are key words of documents. In general, semantics distance is a real number in $[0, \infty)$. Semantics distance has tight association with word similarity. Between two words, the bigger semantics distance is, the lower semantics similarity is and vice versa. They can be built a simple correspondence that need satisfy some conditions as follows: 1) similarity is 1 when semantics distance is 0 between two words; 2) similarity is 0 when semantics distance is infinity between two words; 3) between two words, the bigger semantics distance is, the lower semantics similarity is (monotony descend).

For two words w_1 and w_2 , similarity expressed as $Sim(w_1, w_2)$, semantics distance is $Dis(w_1, w_2)$, then one can define a simple transfer relation that satisfy the above conditions:

$$Sim(w_1, w_2) = \frac{\alpha}{Dis(w_1, w_2) + \alpha} \quad (1)$$

α is a adjustable parameter that embody the words' distance value when similarity is 0.5. In the most cases, directly computing the words' similarity is difficult, so distance measurement can be calculated in advance and then transfer the similarity for words.

In general, thesaurus is the basis of the semantics distance measurement throughout computing MSCA (the Most Specific Common Abstraction) to acquire. To calculate semantics distance, one must use a comprehensive and exact structural semantic resource repository. Hownet (<http://www.keenage.com>) that involves more complete semantics knowledge content and is referred in some Chinese information processing is suitable for this studying.

Hownet includes two main definitions: concept and sememe. Concept is a description for vocabulary's semantics and every word can be expressed several concepts. Concept applies a knowledge representation language that uses sememe as vocabulary to describe.

Differentiated from the other thesaurus (e.g. Wordnet), Hownet don't reduce concept to a tree-like hierarchical architecture and that try to depict every concept using a series of sememes. Hownet adopts 1500 sememes which are divided into some categories as follows:

- 1) Event; 2) entity; 3) attribute; 4) aValue; 5) quantity;
- 6) qValue; 7) SecondaryFeature; 8) syntax; 9) EventRole;
- 10) EventFeatures.

For these sememes, they can be reduced to 3 groups: group 1 is called basic sememe to describe semantics feature for single concept containing sememes from category 1 to category 7; syntactic sememe only include category 8 to describe syntactic feature for words; group 3 contain category 9 and 10 called relation sememe to denote relation between concepts (similar to lattice relation from lattice syntax).

Semantics distance $d_1(p_1, p_2)$ between two sememes p_1 and p_2 is the path length from p_1 to p_2 in the sememe hierarchy structure.

For concept S_1 and S_2 which they have only one sememe in Hownet, semantics distance $d_1(S_1, S_2)$ is called the first basic sememe; except from the first basic sememe expression, for concept S_1 and S_2 which their semantics in Hownet is a set of basic sememes, $d_2(S_1, S_2)$ is defined as this part's semantic distance.

Corresponding to relation sememe description, its value is a feature structure. Considering every feature for the feature structure, its attribute is a relation sememe and its value is a basic sememe or a concrete word. This part of semantics distance for two concept S_1 and S_2 denote as $d_3(S_1, S_2)$.

For every feature of the above feature structure, if its value is a set in which the element of the set is a basic sememe or a concrete word, $d_4(S_1, S_2)$ can be designed to describe the part of relation signal sememe's semantics distance for concept S_1 and S_2 .

Naturally, for the first basic sememe $d_1(S_1, S_2)$, $S_1(S_2)$ have a element-sememe $p_1(p_2)$ in Hownet, then $d_1(S_1, S_2) = d_1(p_1, p_2)$.

For the other basic sememes, if S_1 includes m sememes, S_2 includes n sememes, then

$$D_2(S_1, S_2) = \begin{cases} \text{avg}[d(p_i, p_j)], m > 0, n > 0 \\ |m - n|, \text{else} \end{cases} \quad (2)$$

where p_{1i} is the sememe of S_1 , p_{2j} is the sememe of S_2 .

The following is a java program for calculating relation sememe:

```
private double disMap(
    Map<String, List<String> map1,
    Map<String, List<String> map2)
{
    if (map1.isEmpty() || map2.isEmpty())
    {return
    Math.abs(map1.keySet().size() - map2.keySet().size());
    }
    double min = DEFAULT_DISTANCE;
    for (String key : map1.keySet()) {
        if (map2.containsKey(key)) {
            List<String> list1 = map1.get(key);
            List<String> list2 = map2.get(key);
            double sim = disPriList(list1, list2);
            if (sim < min) {
                min = sim;
            }
        }
    }
    return min;
}
```

Similarly, we can also get the java program for calculating relation signal sememe's semantics distance.

Considering the above-mentioned factors, for two concepts S_1 and S_2 , semantics distance is defined as [15]:

$$d(S_1, S_2) = \sum_{i=1}^4 \beta_i \square d_i(S_1, S_2) \quad (3)$$

where β_i ($1 \leq i \leq 4$) is adjustable parameter and $\beta_1 + \beta_2 + \beta_3 + \beta_4 = 1$, $\beta_1 \geq \beta_2 \geq \beta_3 \geq \beta_4$; if $d_i = 0$, then β_i will assign other item proportionally. The act for global similarity from d_1 to d_4 is descending order. Since the first basic sememe expression reflects the main feature for concept, its weigh value should be defined comparatively bigger and larger than 0.5 usually.

Based on semantics distance between Chinese concepts, we can calculate semantics distance between two sentences w_1 and w_2 for Chinese SORL [16], where w_1 contains m concepts (S_{11}, \dots, S_{1m}), w_2 has n concepts (S_{21}, \dots, S_{2n}).

If w_1 is context-unaware and S_{1i} is unknown, then $Dis(w_1, w_2) = \min Dis(S_{1i}, S_{2j})$, $1 \leq i \leq m$, $1 \leq j \leq n$.

If w_1 is context-aware and S_{1i} is definite, then $Dis(w_1, w_2) = \min Dis(S_{-}\{1i\}, S_{-}\{2j\})$, $1 \leq j \leq n$.

Similarity measurement between two ontologies will be calculated based on the above parts according to weight value synthetically. The relation between ontology similarity measurement and connecting ontologies can be induced as follows: firstly the extracting operation for ontologies is processed to adopt limited candidate ontologies; then calculating ontology similarity among ontologies will be run in order to choose the most similar ontologies for matching.

On the basis of studying in this section, we have designed Chinese semantics distance measurer and matcher for software requirements semantics matching measurement on connecting ontologies to build a measurement ground for connecting ontologies generating.

2.4.2 Integrating Environment

This section presents the design of interoperability extending integrating environment for requirements semantic based CO in Figure 5. Applying sub-ontology extracting algorithm, DPO can be generated from requirements asset that has been produced by domain modeling tool in the phase of requirements elicitation. DPO and domain requirements asset together become reusable asset for requirements acquiring and modeling tool.

Within the requirements acquiring and modeling tool, semantic matcher, which can execute matching operation with semantic distance measurement tool to achieve the matching for role, goal, and process of requirements asset, will be added. Main functions of semantic distance measurer include: measure semantic distance between two concepts; measure semantic distance between two ontologies; measure semantic distance between two services. Existing basis is: 1) thesaurus: WordNet (English), HowNet (Chinese); 2) similarity calculating based on two thesaurus.

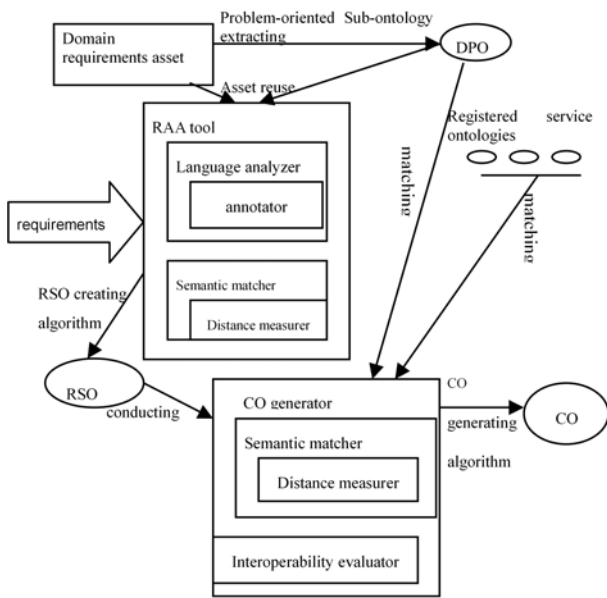


Figure 5. Requirements semantics interoperability extending integrating environment based-on CO

Adopted approach is: calculating two concepts similarity from words similarity; calculating ontologies similarity based on concepts similarity; calculating services similarity based on concepts similarity.

To generate CO, the function of CO generator is driven and conducted by the control structure of RSO, and it will use semantic matcher and interoperability level evaluator. It can automatically complete the task for looking up reusable resources with CO generating algorithm purposed in the above part to the more extent.

After received CO, interoperability level evaluator, which will evaluate semantic interoperability level, able to decide the preference grade for candidate services and forecast the QoE of users.

We have designed and implemented a series of tools for supporting service identifying and composition based on CO and DPO. Relative prototype and validation of the proposed approach have also partly achieved. Experiment has demonstrated that the proposed approach is useful for service finding and integrating. The snapshot of primary tools and Prototype system for context of traffic travel problem domain can see from Figure 6.

3. Related Work

Application of ontology in RE starts from domain engineering. As reusable core resources in product line, domain requirements [17] mainly solve requirements modeling issue for component-oriented software system.

Dr. Jerome Euzenat from INRIA Grenoble Rhone-Alpes in France has studied semantic interoperability issues based ontology mapping [18,19] and acts as principal in NeOn project of EU FP6 plan. In June 2008, In-

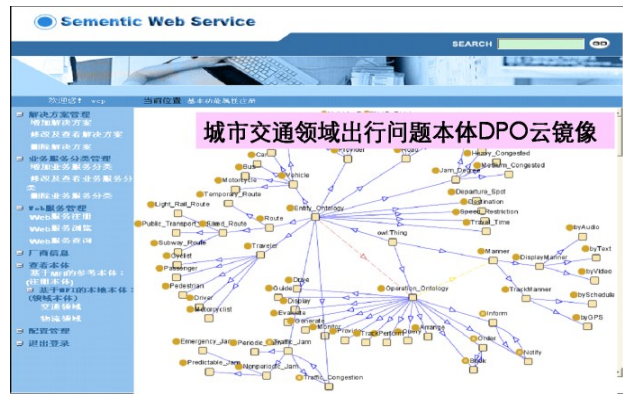


Figure 6. Prototype context and tools of traffic travel problem domain

formatics of EU startup semantic interoperability central plan for Europe and set up first session in Brussels aiming at realizing semantic data interoperability for E-government in Europe. Open source SILIME project of MIT-Semantic Interoperability of Metadata and Information in unLike Environments attempts to semantic interoperability for data resources (such as data library).

The studying of connecting ontologies is new direction in the world. Initial investigation studies original domain-level ontology for heterogeneity and explores how to create new ontology for covering original ontology with collaboration and consistence, and also containing ontology grouping technology (for example ontology mapping, ontology aligning, ontology merging etc.). In 2007, the paper by Shuaib Karim [20] presented a CO application framework that need not cover original ontology and focus on studying transfer principle and intermediate concept among original ontologies. Cregan Anne [21] proposes to build semantic interoperability by CO and gives some CO examples of gene ontology in 2008. However, in Cregan Anne's paper, connecting manners of CO, incentive of connecting, method and critical content of building semantic interoperability are absent. We also notice that Linked Open Data [22] initiative has become the existing foundation for federal Web of Data.

Now, together with CO and RE, the investigation of

requirements semantic interoperability extending for networked software with respect to service-oriented computing just begins to proceed, and a great deal of theoretical and technological issues will require to solve.

4. Conclusions

This paper explores ontology-based RE, for interoperability extending of requirements semantics; we present CO approach to improve requirements modeling under the condition of distributed services aggregation with loosely coupling and different domain. Some formal definition and generating algorithm of CO are given. With the novel approach, a integrating environment and measurement system based on CO is designed and implemented.

Further work can be classified as follows: studying partial meaning of semantic interoperability for networked software requirements; build CO based on Linked Open Data infrastructure; empirical testing for integrating environment with multi-domain, such as financial risk assessment, environment protection and so on.

5. Acknowledgments

This research has been partly supported by the National Basic Research Program of China (Grant No. 2007CB-310801) and the National Natural Science Foundation of China under Grant No.60970017 and 60903034.

REFERENCES

- [1] Rajkumar Buyyaa, Chee Shin Yeoa, Srikumar Venugopala, James Broberg, and Ivona Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, Vol. 25, No. 6, pp. 599–616, June 2009.
- [2] K. Q. He, R. Peng, W. Liu, *et al.* "Networked Software," Science Press, Beijing, 2008.
- [3] Z. Jin, L. Liu, and Y. Jin, "Software Requirements Engineering: Principles and Method," Science Press, Beijing, 2008.
- [4] J. Mylopoulos, L. Chung, and E. Yu, "From object-oriented to goal-oriented requirements analysis," *Communications of ACM*, Vol. 42, No. 1, pp. 31–37, January 1999.
- [5] A. V. Lamsweerde and E. Letier, "From object orientation to goal orientation: A paradigm shift for requirements engineering," *Radical Innovations of Software and System Engineering in the Future*, pp. 325–340, 2004.
- [6] R. Q. Lu, Z. Jin, and G. Chen, "Ontology-oriented requirements analysis," *Journal of Software*, Vol. 11, No. 8, pp. 1009–1017, August 2000.
- [7] Z. Jin, "Ontology-based requirements elicitation," *Chinese Journal of Computers*, Vol. 23, No. 5, pp. 486–492, May 2000.
- [8] R. A. Falbo, G. Guizzardi, and K. C. Duarte, "An ontological approach to domain engineering," In *Proceedings of the International Conference on Software Engineering and Knowledge Engineering (SEKE02)*, Ischia, Italy, pp. 351–358, 2002.
- [9] J. Wang, K. He, P. Gong, *et al.* "RGPS: A unified requirements meta-modeling frame for networked software," In *Proceedings of Third International Workshop on Advances and Applications of Problem Frames (IWAAPF'08) at 30th International Conference on Software Engineering (ICSE'08)*, Leipzig, Germany, pp. 29–35, May 2008.
- [10] K. Q. He, F. He, and B. Li, "Research on service oriented ontology meta modeling theory and methodology," *Chinese Journal of Computers*, Vol. 28, No. 4, pp. 524–533, April 2005.
- [11] K. Q. He, Y. F. He, and C. Wang, "International standard: Information technology-metamodel framework for interoperability (mfi)-3: metamodel for ontology registration," (ISO/IEC19763-3), online at: http://www.iso.org/iso/iso_catologue/catalogue_tc/catalogue_detail.htm?csnumber=38637. ISO, 2007.
- [12] Y. Mao, Z. Wu, and H. Chen, "Sub-ontology based resource management for web-based e-learning," doi: <http://doi.ieeeecomputersociety.org/10.1109/TKDE.2008.127>, 2008
- [13] B. Hu, K. Q. He, H. F. Chen, and J. Wang, "Requirements driven web service composition based on RGPS domain assets: Approach and realization," *Journal of Chinese Computer System*, Vol. 30, No. 5, pp. 859–862, May 2009.
- [14] K. Q. He, "Semantic interoperability refining and clustering theory and its application in on demand service aggregation," *Science in China, F: Information Science* (unpublished).
- [15] L. Lin, "Text clustering research based on semantic distance," Master's thesis, Xiamen University, April 2007.
- [16] W. Liu, "Research on services-oriented software requirements elicitation and analysis," PhD thesis, Wuhan University, June 2008.
- [17] M. Mikyong and Y. Keunhyuk, "An approach to developing domain requirements as a core asset based on commonality and variability analysis in a product line," *IEEE Software Engineering* (unpublished), Vol. 31, No. 7, pp. 551–569, July 2005.
- [18] J. Euzenat, "An api for ontology alignment," In *Proceedings of 3rd International Semantic Web Conference (ISWC)*, Hiroshima, Japan, *Lecture Notes in Computer Science*, Vol. 3298, pp. 698–712, 2004.
- [19] J. Euzenat and P. Shvaiko, "Ontology matching springer," Heidelberg, Germany, 2007.
- [20] Shuaib Karim, Khalid Latif1, and A. Min Tjoa1, "Providing universal accessibility using connecting ontologies: A holistic approach," *Lecture Notes in Computer Science* 4556, pringer-Verlag, Berlin Heidelberg, Vol. 3, pp. 637–646, S 2007.
- [21] Cregan Anne, "W3c semantic web ontology languages: Owl and rdf tutorial," Technical Report, ISO/IEC JTC1 SC32 11th Open Forum on Metadata Registries, Sydney, Australia, May 2008. Tutorial.ppt.
- [22] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data: The story so far," *International Journal on Semantic Web and Information Systems*, Vol. 5, No. 3, pp. 1–22, 2009.