He KQ, Wang J, Liang P. Semantic interoperability aggregation in service requirements refinement. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 25(6): 1103–1117 Nov. 2010. DOI 10.1007/s11390-010-1088-1

Semantic Interoperability Aggregation in Service Requirements Refinement

Ke-Qing He (何克清), *Senior Member, CCF, IEEE*, Jian Wang (王　健), *Member, CCF* and Peng Liang (梁　鹏), *Member, CCF, ACM, IEEE*

State Key Lab of Software Engineering (SKLSE), Wuhan University, Wuhan 430072, China

E-mail: {hekeqing, jianwang, pengliang}@sklse.org

Received July 15, 2009; revised January 28, 2010.

Abstract Semantic refinement of stakeholders' requirements is a fundamental issue in requirements engineering. Facing with the on-demand collaboration problem among the heterogeneous, autonomous, and dynamic service resources in the Web, service requirements refinement becomes extremely important, and the key issue in service requirements refinement is semantic interoperability aggregation. A method for creating connecting ontologies driven by requirement sign ontology is proposed. Based on connecting ontologies, a method for semantic interoperability aggregation in requirements refinement is proposed. In addition, we discover that the necessary condition for semantic interoperability is semantic similarity, and the sufficient condition is the coverability of the agreed mediation ontology. Based on this viewpoint, a metric framework for calculating semantic interoperability capability is proposed. This methodology can provide a semantic representation mechanism for refining users' requirements; meanwhile, since users' requirements in the Web usually originate from different domains, it can also provide semantic interoperability guidance for networked service discovery, and is an effective approach for the realization of on-demand service integration. The methodology will be beneficial in service-oriented software engineering and cloud computing.

Keywords connecting ontologies, requirements refinement, semantic interoperability aggregation

1 Introduction

With the rapid development of Web service and SaaS (Software as a Service), software engineering is transforming into a discipline of software service engineering in the network age. The "Service-Oriented Requirements Engineering (SORE)" becomes a key research topic in the user-centric on-demand service software paradigm[1], and is the research focus of "Service-Oriented Software Engineering (SOSE)".

Since Web service is becoming a new software paradigm to realize requirements, it is necessary to consider how services can affect requirements refinement, the key among which is the interoperability problem brought by the loosely coupled characteristic of services. Ontology, as an explicit information modeling method, can be viewed as a fundamental approach to support semantic interoperability. In SORE, ontology can be used as the semantic representation mechanism of requirements, the semantic encapsulation mechanism of services, as well as the modeling mechanism of domain assets. During the refinement process of service requirements, how to connect these different kinds of ontologies and provide a unified semantic carrier for service composition driven by requirements, are key issues in SOSE.

To address these issues, we propose a semantic interoperability aggregation method for requirements refinement based on connecting ontologies. This approach provides a dynamic semantic representation mechanism for domain-oriented and user requirements-dominated service composition.

The research on connecting ontologies is receiving much attention recently. The initial research on connecting ontologies usually focuses on how to generate a collaborative ontology to cover the original heterogeneous ontologies. Karim *et al.* propose to use connecting ontology as a mediator that does not cover the original ontologies[2]. Their research focuses on defining the transformation rules, intermediate concepts between original ontologies. Cregan *et al.* propose to implement semantic interoperability by connecting ontologies, and present an example of connecting ontologies on gene domain[3]. In our opinion, connecting ontologies are

Regular Paper

This work is supported by the National Basic Research 973 Program of China under Grant No. 2007CB310801, the National Natural Science Foundation of China under Grant Nos. 60970017 and 60903034.

neither new ontologies which cover the original ontologies, nor mediator ontologies. Connecting ontologies can be generated by matching semantically with existing ontologies based on users' requirements, and loosely coupled connecting these ontologies based on the control structure of users' requirement sign ontology. It is significant to investigate the theory and method of connecting ontologies for the research of domain-oriented and service community-oriented networked software.

The rest of this paper is organized as follows. Section 2 introduces the refinement problem of service requirements, and discusses the role of semantic interoperability in service requirements refinement. The measurement framework and calculation methods are presented in Section 3. Section 4 investigates the semantic interoperability in connecting ontologies. A case study is presented to demonstrate the application of semantic interoperability aggregation in Section 5. We conclude our work with future research directions in Section 6.

2 Refinement of Service Requirements

2.1 Ontological Encapsulation of Services

With the development of service computing technology, thousands of Web services are disseminated throughout the Web. Generally, these services are structurally heterogeneous, non self-possession, and significantly different in underlying implementation techniques (e.g., programming languages, running environment). It is difficult to study the interaction and collaboration between them in the implementation level.

Considering the situation mentioned above, we investigate the issue of semantic interoperability aggregation of services based on the ontology and metamodeling theory[4]. Focusing on the existing service description in WSDL, we provide the ontological encapsulation (description) of services using MFI-7 (ISO/IEC 19763-7: Metamodel framework for service registration), which is in the working draft edition. The ontological description of services is composed of entity concepts, operation concepts and the semantic relationship between them. By semantically annotating the business models and interfaces of services, the ontological description of services can be generated.

The service ontology provides the capability for semantically describing services' behavior. This ontological description of services makes a separation of the concern between what and how services' behavior can be realized, supports the management and control of services through the strategy and contract, and supports communication through coarse-grained messages. Consequently, the ontological description makes a service not only a calculation entity, but also a logical entity. The ontological description of services can be used for semantically classification, registration and publication of heterogeneous services.

2.2 Service-Oriented Domain Assets Modeling

In the age of network, users' requirements with the characteristics of diversity and personalization bring much challenge to software development. In order to provide on-demand service for users timely, at a low cost and with a reliable quality, it is necessary to develop software services in a form of mass customization. The basis of mass customization is to develop reusable domain core assets by domain modeling. Faced with the common domain requirements, we propose a domain modeling method based on O-RGPS (Ontology based Role-Goal-Process-Service) metamodel[1,5], which can be used to guide developing reusable domain assets (or solutions) for domain requirements. Domain assets can provide fundamental support for domain-oriented service requirements refinement.

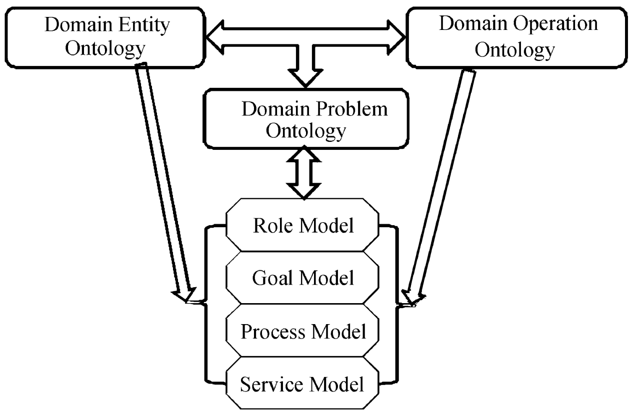


Fig.1. Organizing domain assets based on DPO.

In O-RGPS domain modeling method, the first step is to construct domain ontologies: domain entity ontology that describes the entity concepts in a domain and the relationships among these concept, domain operation ontology that describes the operation concepts in a domain and the relationships among these concepts. Based on domain ontologies, we can construct domain models including role model, goal model, process model and service model, and semantically annotate the information in domain models. Then, domain models can be classified according to specific domain problem, and a corresponding domain problem ontology (DPO) can be constructed (see Fig.1), so as to realize the classification and index of domain models based on DPO. On one hand, DPO defines the relationship with the domain ontologies. On the other hand, DPO can cover the general information of RGPS domain models, so as to relate the domain ontologies with domain models, and provide integrated solution for domain problems. In this way, the domain assets can be modeled and managed semantically in order to improve the reuse efficiency and quality of domain assets.

2.3 Refinement Process of Requirements

The purpose of service requirements refinement is to select and compose appropriate services to satisfy users' requirements based on users' requirements elicitation. The refinement process is described as follows:

A. *Description of Service Requirements*

To elicit users' personalized requirements, we define a requirements description language SORL (Service Oriented Requirements Language)[1] to elicit users' requirements. The users can propose their requirements based on the language patterns defined in SORL. The elicited requirements can be parsed by finite automata, and initial requirements models can be created automatically.

B. *Matching Requirements Description Model with Domain Models*

Since users' requirements tend to target specific problems in a domain, it is needed to semantically match users' requirements into the domain problem ontology. Thus, requirement sign ontology (RSO), which is an ontological description of user requirements, can be generated by conducting the domain-oriented requirements analysis. The matching result from the users' requirements to DPO is usually sub-ontology of DPO, denoted as O_0. There is a tightly coupled relationship between O_0 and RSO. In service-oriented software systems, users' requirements are difficult to be fully covered by a DPO due to the diversity and personalization of users' requirements. In such a situation, we have to discover service ontologies available in the Web for these unmatched users' requirements, so that the service ontologies from other domains can satisfy the semantics necessity of users' requirements. The matching results from the RSO to the service ontologies of other domains are also the sub-ontologies of DPOs, denoted as O_i $(i = 1 \sim n)$, there is a loosely coupled relationship between O_i and RSO. In this way, connecting ontologies (CO) consist of RSO, the domain-oriented ontology O_0 that is tightly coupled with RSO, and service ontologies O_i that are loosely coupled with RSO (see Fig.2).

Then, we formally describe these three kinds of ontologies: DPO, RSO, and CO. Generally speaking, ontology can be formally described as a 4-tuple (C, R, A, I)[6], where C denotes a set of concepts, R denotes a set of relations, A denotes a set of axioms, and I denotes a set of instances. Since I is optional in the definition of ontology, and instances are not considered in domain modeling, so DPO can be described as a 3-tuple ($C_{\rm DPO}$, $R_{\rm DPO}$, $A_{\rm DPO}$), where $C_{\rm DPO}$ denotes the domain concept set, $R_{\rm DPO}$ denotes the relation set between the domain concepts, and $A_{\rm DPO}$ denotes the axiom set among the domain concepts and relations.

RSO can be described as a 4-tuple ($C_{\rm RSO}$, $R_{\rm RSO}$, $A_{\rm RSO}$, $I_{\rm RSO}$), where $C_{\rm RSO}$ denotes the requirements concept set, $R_{\rm RSO}$ denotes the relation set between the requirements concepts, $A_{\rm RSO}$ denotes the axiom set among the requirements concepts and relations, and $I_{\rm RSO}$ denotes the requirements instance set.

CO can be described as a 4-tuple ($C_{\rm CO}$, $R_{\rm CO}$, $A_{\rm CO}$, $I_{\rm CO}$), where $C_{\rm CO} = C_{\rm RSO} \cup (\cup_i C_{{\rm DPO}_i})$, that is, the union of the concept set in RSO and all related DPOs; $R_{\rm CO} = R_{\rm RSO} \cup (\cup_i R_{{\rm DPO}_i})$, that is, the union of the relation set in RSO and all related DPO; $A_{\rm CO} = A_{\rm RSO} \cup (\cup_i A_{{\rm DPO}_i})$, that is, the union of the axiom set in RSO and all related DPO; $I_{\rm CO} = I_{\rm RSO}$, that is, the instance set of RSO.

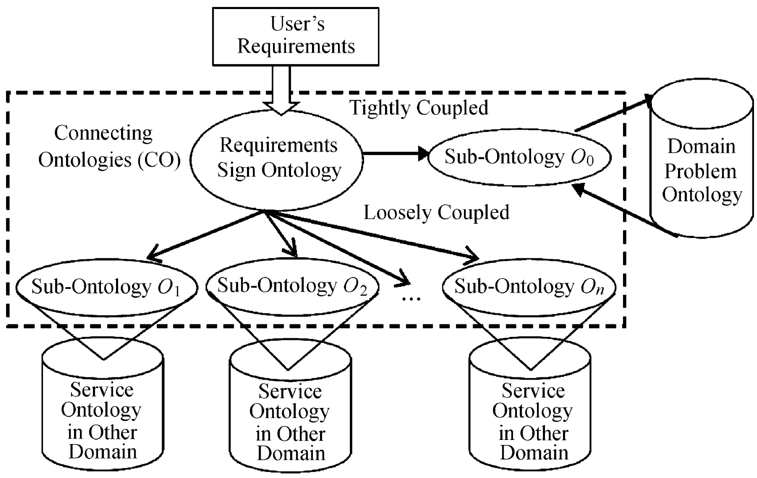


Fig.2. Generation of connecting ontologies.

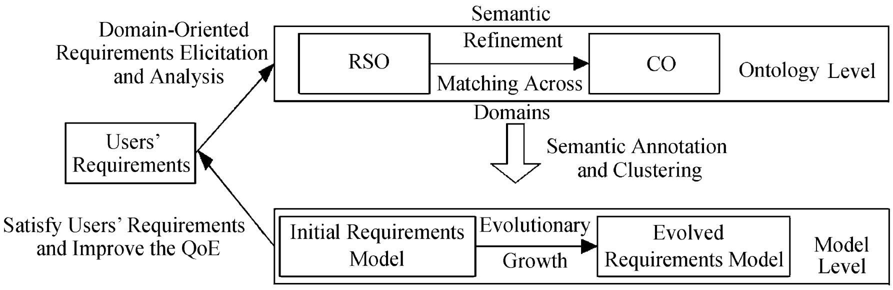


Fig.3. Requirements analysis and modeling based on CO.

C. *Requirements Analysis Based on Domain Model*

When the requirements are matched into DPO, the role and goal information of users' requirements can be mapped to the goal models in the domain models based on the relationship between DPO and corresponding domain models. Then according to the goal decomposition and restriction relationship defined in goal model, the requirements model can be supplemented, and the upper goal can be decomposed into operational goals step by step. Furthermore, a composite process chain can be generated based on the corresponding relationship between the operational goal and the process model.

D. *Mapping Refined Requirements Model Into Services*

According to the processes in the process chain generated in Step C, corresponding services can be discovered and selected. The CO serves as the common semantic agreement during discovering and matching services. Finally, based on the control structures defined in the process chain, the selected service can be connected and users' requirements can be realized by invoking and composing these services.

2.4 Role of Semantic Interoperability in Requirements Refinement

It is obvious that the CO provides a unified semantic representation mechanism for the development of user-driven software services. The process of requirements refinement is composed of the phases of requirements analysis, service discovery, and service composition, where CO acts as the semantic reference, and ensures the consistency and completeness of requirements refinement.

Generally, the domain-oriented RSO is also dynamically evolved due to the diversity and dynamicity of users' requirements, so a single domain ontology cannot fully cover RSO, which requires to match to services in other domains dynamically in the Web. In one perspective, the factors of dynamic changes result in the dynamicity of generated CO. As shown in Fig.3, the requirements models are also dynamically evolved in the model level; in the other perspective, the generated CO always acts as the guidance and basis for requirements evolution modeling in the dynamic evolution.

During the construction of the CO, it is necessary to realize the semantic interoperability aggregation among RSO and DPO together with the service ontologies in other domains. Semantic interoperability aggregation means the different kinds of ontologies can be connected together so as to provide a unified semantic carrier for the interoperation and aggregation of models. The key issue for semantic interoperability aggregation is the measurement of the semantic interoperability capability.

3 Semantic Interoperability Capability and Measurement

3.1 Semantic Interoperability Capability

Semantic interoperability depends on the system's application domain, and different application domains have different semantic interoperability objectives. In general, it refers to the capability that two software modules or systems can exchange the data with precise meaning, and the receiving party can accurately translate or convert the information carried by the data, including the knowledge, i.e., information and knowledge that can be understood, and ultimately produce an effective collaborative results. The validity of the collaborative results of the interoperability is determined and negotiated by the two parties involved[7].

The capability of semantic interoperability of services is essentially the capability of the interaction and behavior collaboration between software services. The levels of semantic interoperability are generally classified as follows[4]:

1) *Meaning Interoperability*: also named as full semantic interoperability. The software service entities can fully share the same semantic agreement when the meaning interoperability between software service ontologies is achievable. Namely, the two parties can realize the ideal interaction and collaboration, and smooth connection in the end.

2) *Partial Semantic Interoperability*: the partial semantic interoperability between software service entities refers to partial satisfaction of the semantic agreement of shared ontologies. The degree of partial semantic interoperability is normally represented by the percentage of shared ontologies. 100% represents the meaning interoperability, and lower percentage means weaker semantic interoperability.

3) *No Semantic Interoperability*: it refers to the situation when the degree of partial semantic interoperability is lower than certain threshold. For example, the syntactic (or structural) interoperability belongs to this type.

The research objective of SIM-TBASSC project (Semantic Interoperability Measures: Template-Based Assurance of Semantic Interoperability in Software Composition)[8] is to discover appropriate mapping of functions and data by evaluating the semantic interoperability of functions and data in software components, and predict the performance of software composition. They introduce the concept of "semantic gauge" to evaluate the matching degree of the functional semantics and data semantics of software components. The results of this project can help users select appropriate components composition solution based on the measurement results, but it does not provide the level classification for the semantic interoperability.

The partial semantic interoperability, as the interoperability capability that most systems have, has been a hot research topic and focus. The SemanticHEALTH[9] project funded by Europe Union FP6 program classifies the semantic interoperability in 4 levels and 3 dimensions in e-Health information system, in which the second level is about partial semantic interoperability.

Compared to the SemanticHEALTH project, our method is based on the CO, and investigates the partial semantic interoperability aggregation between software services. We also study the semantic distance and threshold in partial semantic interoperability aggregation.

3.2 Capability Measurement of Semantic Interoperability Between Ontologies

It is an unsolved issue on how to measure the capability of semantic interoperability between ontologies. We argue that the capability measurement of semantic interoperability is composed of two parts: semantic similarity measurement of ontologies, which is the basic capability for semantic interoperability; and capability measurement of semantic interoperability as a sufficient capability, which is an agreement on semantic interoperability capability between the two parties. Based on the analysis above, we propose a measurement framework for the semantic interoperability capability based on ontology (as shown in Fig.4).

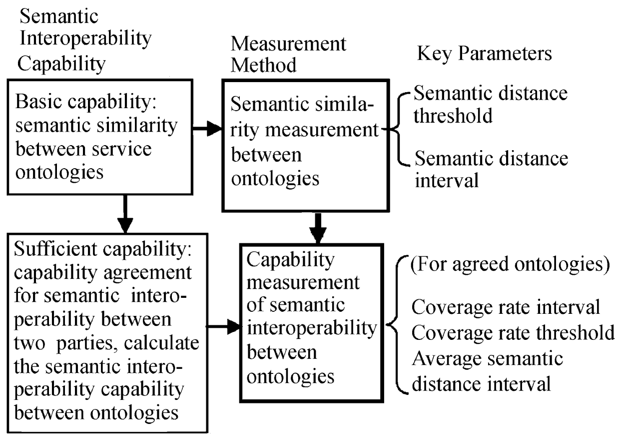


Fig.4. Measurement framework for the semantic interoperability capability based on ontology.

The detailed introduction of the measurement framework and measurement solution is presented below:

Step 1. *Semantic Similarity Measurement Between Ontologies*

Semantic distance is a key method to calculate the semantic similarity between ontologies. The shorter the semantic distance, the higher degree the semantic similarity. This subsection will present the calculation method of the semantic distance between ontologies based on the precision and recall rate, and another semantic distance calculation method based on mediator ontologies.

1) *Semantic Distance Calculation by Precision and Recall Rate*

The concepts of precision and recall rate originate from the Information Retrieval (IR) theory[10]. In [11-12], the authors employ the precision and recall rate on quality evaluation of ontology mapping and matching. Their work focuses on the quality evaluation of the mappings between ontologies at the conceptual level without considering the effect to the instance level. In this part, we will present the semantic similarity and distance measurement based on the ontology instance level.

(a) *Semantic Distance in Computational Linguistics*

The concept of semantic distance originates from computational linguistics[13]. In computational linguistics, the concept of semantic relatedness involves two lexemes in a lexical resource. Its inverse is the semantic distance, which formalizes and quantifies the intuitive notion of similarity and dissimilarity between two lexically expressed concepts. An extensive survey and classification of measures of semantic distance are presented in [14]. A domain model is composed of lexical concepts and the relationships between them. Thus the similarity and dissimilarity between models can be represented by the aggregation of the similarity and dissimilarity between their compositional concepts, i.e., the semantic distance between concepts. Similarly, we try to define the similarity between the ontological models based on the similarity between the concepts.

(b) *Semantic Distance Between Ontological Models*

The concept mapping between ontological models provides a theoretical base for ontology similarity. The semantic distance between two models depends on two parts mentioned as follows:

Semantic Difference of Concept Mapping: semantic mapping is a basic method for calculating the semantic distance. The definition of concept mapping between ontologies has a direct impact on the precision of semantic distance calculation. There are different concepts mapping relationships being proposed. The structural concept mapping proposed in this paper is composed of several semantic mapping relationships, e.g., *EquivalentClassOf*, *SubClassOf*, *AttributeOf*, *PartOf*, *InstanceOf*, *NoMatchingPair*, which can increase the precision of semantic matching.

The semantic distance between two concepts can be different with different mapping relationships. The semantic distance between two concepts with *EquivalentClassOf* mapping relationship is shorter (more similar) than those with *SubClassOf* mapping relationship.

Model Coverage Difference: due to the unavoidable heterogeneity between different models, the model mapping cannot enforce that all the concepts can find their counterparts in other models, i.e., the *NoMatchingPair* mapping relationship, which increases the semantic distance (more dissimilar) between models.

These two kinds of difference can be both measured by precision and recall rate.

(c) *Calculation of the Semantic Distance*

The ontology similarity and matching activity can be viewed as an information retrieval task[15] and the matching quality can be quantified in terms of precision, recall, and the F-measure (the metrics for quantifying integrated information retrieval quality).

The example below acts as the dataset for calculating precision and recall rate. Fig.5 illustrates the ontology matching in the perspective of instance classification based on concept mappings. The two bigger circles x and y represent two concepts in different ontological models, the small circles inside each concept circle represent the instances belonging to them, and the directed arrow between the small circles are instance classifications. If concept x (e.g., Human) is *SuperClassOf* y (e.g., Man), three instance classification scenarios may exist:

- an instance of concept x is classified correctly as an instance of concept y, e.g., instance a (John is a human and he is also a man);
- an instance of concept x is classified as instance of concept y, but it is not a correct mapping, e.g., instance b (Tom is a human, but he is not a man, he is a boy). This instance classification is possible in practice by an instance classifier since concepts man and boy are similar;
- an instance of concept x cannot be classified as an instance of concept y, e.g., instance c (Mary is a human but she is not a man, she is a woman).

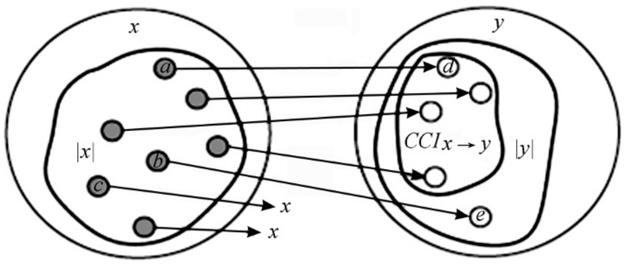


Fig.5. Instance classification based on ontology matching.

Based on the instance classification scenarios, three instance sets can be retrieved (see Fig.5) originating from the IR theory for the calculation of precision and recall rate:

- $|x|$: all instances to be classified in concept x regardless whether they are classifiable or not, e.g., a, b, $c \in |x|$. Relevant data in IR theory;
- $|y|$: all instances classified to concept y regardless whether they are correctly classified or not, e.g., d, $e \in |y|$, retrieved data in IR theory;
- $CCI_{x \to y}$: all correctly classified instances from concept x to y, e.g., $d \in CCI_{x \to y}$, relevant retrieved data in IR theory.

Then the precision (P) and recall (R) rate based on concept mapping from x to y can be defined as follows:

$$P_{x \to y} = \frac{\textit{relevant retrieved data}}{\textit{retrieved data}} = \frac{CCI_{x \to y}}{|y|}, \quad (1)$$

$$R_{x \to y} = \frac{\textit{relevant retrieved data}}{\textit{relevant data}} = \frac{CCI_{x \to y}}{|x|}. \quad (2)$$

The precision (MP) and recall (MR) rates of the model mapping from S to T can be calculated based on the aggregation of precision and recall rate for the individual concept mappings. Some symbols for the calculation of MP and MR are defined: x_i denotes a concept of model S, y_i denotes a set of concepts of model T due to multiple mapping relationships from x_i to y_i, W_{X_i} denotes the weight of concept x_i in model S

(i.e., the percentage of amount of instances in concept x_i denoted by $|x_i|$ in relation to the whole amount of instances in model S denoted by $|S|$), and $NoC(S)$ is a function to get the number of concepts in model S. The formulas for the calculation of MP and MR are defined as follows:

$$W_{x_i} = \frac{|x_i|}{|S|}, \tag{3}$$

$$MP_{S\to T} = \sum_{i=1}^{NoC(S)} (P_{x_i\to y_i} \times W_{x_i})(x_i \in S,\ y_i \subset T), \tag{4}$$

$$MR_{S\to T} = \sum_{i=1}^{NoC(S)} (R_{x_i\to y_i} \times W_{x_i})(x_i \in S,\ y_i \subset T). \tag{5}$$

F-measure (F), the combination of precision and recall rate in IR theory, and its inverse semantic distance (SD) are defined as follows:

$$F_{S\to T} = \frac{2 \times MP_{S\to T} \times MR_{S\to T}}{MP_{S\to T} + MR_{S\to T}}, \tag{6}$$

$$SD_{S\to T} = \frac{1}{F_{S\to T}}. \tag{7}$$

We explain the meaning of these formula results in the context of semantic similarity between ontology models:

- $P_{x\to y}$ is the percentage of correctly classified instances in concept x to all the classified instances in concept y, and $R_{x\to y}$ is the percentage of correctly classified instances in concept x to all the instances of concept x;
- $MP_{S\to T}$ is the percentage of correctly classified instances in model S to all the classified instances in model T, and $MR_{S\to T}$ is the percentage of correctly classified instances in model S to all the instances of model S;
- $F_{S\to T}$ is a combined criterion to quantify the sharing performance: the greater the F value is, the more similar the two models are, and vice versa;
- $SD_{S\to T}$ is the multiplicative inverse of F, the greater the SD value is, the more dissimilar the two models are, which matches the notion of semantic distance in computational linguistics.

2) *Semantic Distance Calculation Method Based on Mediator Ontologies*

Most of semantic distance calculation methods[16] calculate the semantic distance between the nodes within the same ontology model. In this paragraph, we present the semantic distance calculation method based on mediator ontologies. First, O_1 and O_2 are the semantic mapping results to the mediator ontology MO by two interactive parties, i.e., the ontologies by correct understanding of mediator ontology, then we can calculate the semantic distance and similarity between O_1 and O_2 using the calculation methods presented above for calculating precision and recall rate.

It reminds that it is necessary to set up the semantic distance interval to address the practical problem in semantic interoperability, i.e., $SD \in [1, K]$, in which $(K > 1)$ is the semantic distance threshold. The semantic distance within this interval has certain degree of semantic similarity, which can satisfy the basic condition of semantic interoperability.

Step 2. *Capability Measurement of Semantic Interoperability Between Ontologies*

This step provides the method for calculating the semantic Interoperability Capability (IC) between ontology models. The objects of this calculation is the ontology models whose semantic distance got in Step 1 is within the semantic distance interval; and the condition of this calculation is that there is an agreement of semantic interoperability between the ontology models involved in semantic interoperability.

1) *A Measurement Method Based on Percentage*

This method calculates the semantic IC based on the calculation of semantic distance between ontologies, and the coverage percentage of ontologies involved with the interoperation parties to the agreed mediator ontologies.

Measurement method based on the ontology weight in mediator ontologies: define the weight "agreement" of important sub-ontologies in the mediator ontologies by ontology partition and weight method, and apply the weighted percentage to measure the semantic IC.

2) *A Measurement Method Based on Ontology Graph Coverage*

Besides the simple measurement method based on percentage, we also present another measurement method based on ontology graph overlapping coverage and semantic depth to quantify the IC "agreement" between the two parties involved in semantic interoperation.

The semantic depth of concept C is denoted as $Depth(C)$, which represents the semantic distance between concept C and the top concept in the ontology classification graph. O_1 and O_2 are the semantic mapping results to the MO (equivalent to the semantic IC "agreement") by two interoperation parties. First, we use the semantic distance calculation method based on MO in Step 1 to check whether the O_1 and O_2 are qualified to the basic condition of semantic interoperability. The semantic mapping results to the MO by O_1 and O_2 as the agreement of partial semantic interoperability is denoted as $O_1 \cap O_2$, and then we can calculate the semantic IC between these two ontologies by (8):

$$IC = \frac{\sum_{i=1}^{m} Depth(C_i)}{\sum_{j=1}^{n} Depth(C_j)}. \quad (8)$$

In (8), C_i ($i = 1 \sim m$) denotes all the concepts in $O_1 \cap O_2$, and C_j ($j = 1 \sim n$) denotes all the concepts in *MO*. Apparently the semantic *IC* calculated by (8) represents the common understanding degree (semantic depth) of two interoperation parties to the semantic *IC* "agreement". If the calculated semantic depth is equal to the semantic depth of "agreed" ontology, then it suggests that the two interoperation parties can realize deep semantic interoperation, i.e., meaning interoperability.

4 Semantic Interoperability Aggregation in Connecting Ontologies

Based on the measurement of semantic interoperability capability, we discuss the semantic interoperability of CO, which includes three aspects: 1) the semantic matching between RSO and DPO, and between RSO and other domain service ontologies; 2) the semantic interoperability aggregation within the coupled ontologies; 3) the behavior collaboration (meaning interoperability aggregation) at CO level. These three aspects are further discussed below.

Aspect 1: *Semantic Matching*

Semantic matching is a research issue on semantic similarity for the realization of semantic substitution. It is a simple semantic interoperability context. The semantic similarity between models is a necessary (but not sufficient) condition for the realization of semantic interoperability. If the semantic similarity between two models is low, it implies that the capability of exchanging meaningful data (e.g., the information, knowledge carried by the data) between two interoperation parties is also quite weak, which consequently produces ineffective collaboration results.

The process of semantic matching between ontology models is composed of the following steps: first, perform the semantic matching for all the concepts; second, calculate the semantic matching capability between ontology models. The detailed process is described as follows: M_1 and M_2 denote the two ontology models to be matched, and C_1 and C_2 are two concepts in M_1 and M_2 respectively. The names of these two concepts are strings S_1 and S_2. First, these two strings are processed by lexical analysis in order to remove the meaningless words (e.g., conjunctions, pronouns, interjections), and retain the meaningful words. The lexical analysis results of S_1 and S_2 are tuples, e.g., $\langle S_{1w1}, \ldots, S_{1wn} \rangle$ and $\langle S_{2w1}, \ldots, S_{2wm} \rangle$. For each word S_{1wi} ($i = 1 \sim n$), let S_{2wj} be the most similar word in S_2, which is calculated by looking up the words similarity table, and this table is pre-constructed by domain experts based on WordNet. The similarity degree between S_{1wi} and S_{2wj} is denoted as $matchScore(S_{1wi}, S_{2wj})$, then the similarity degree between C_1 and C_2 is $matchScore(C_1, C_2) = \sum_i matchScore(S_{1wi}, S_{2wj})/n$, the average similarity degree of all the words in S_1.

In the next step, calculate the matching degree of all the concepts, and select the concept C_k, which has maximum matching degree, as the matching part from C_1 to M_2. Let s represent the threshold of matching degree, if $matchScore(C_1, C_k) \geqslant s$, then $\langle C_1, C_k \rangle \in Matched(M_1 \cap M_2)$. $Matched(M_1 \cap M_2)$ denotes the matched concepts pair.

The matching rate of M_1 and M_2 in the matching process is defined as:

$$MatchRate(M_1, M_2) = \frac{Num(Matched(M_1 \cap M_2))}{Num(M_1)} \quad (9)$$

where $Num(Matched(M_1 \cap M_2))$ represents the number of matched concept mapping pairs between M_1 and M_2, and $Num(M_1)$ represents the number of all the concepts in M_1.

Aspect 2: *Semantic Interoperability Aggregation in Coupled Ontology O_i*

For the coupled sub-ontologies O_i ($i = 0 \sim n$) in CO, we focus on the semantic interoperability aggregation of service ontologies annotated by O_i. The interoperability capability agreement between services is the semantic constraints to O_i, namely the dynamic aggregation should satisfy and cover O_i at a semantic level. The semantic interoperability aggregation is interoperability semantic aggregation based on O_i semantics between service ontologies, and the service composition is the composition modeling of service models annotated by service ontologies based on semantic interoperability aggregation.

The semantic distance and similarity between models are commonly used to measure the semantic interoperability capability. However, as presented in Section 3, the semantic distance and semantic similarity are both used to address the issue of semantic similarity between models. The semantic interoperability capability relies on not only the semantic similarity between models for interoperation, but also (more importantly) the interoperability agreement between the models. As a result, the interoperability agreement becomes essential criteria for measurement of semantic interoperability capability. We can realize agreement-based measurement of semantic interoperability capability, and judge the semantic interoperability capability between models.

It is necessary to include the behavior collaboration between several service models to satisfy the agreement of semantic interoperability capability, and consequently we regard this common "agreement" as the standard for the semantic interoperability capability among multiple service models. The result of interoperation between two models can only partially cover the common "agreement", which is called partial semantic interoperability. The greater the "agreement" coverage between two models, the stronger the semantic interoperability capability between these two models. Obviously, the semantics of the multiple models is needed to cover and satisfy the common "agreement" for the realization of semantic aggregation, which we called partial semantic interoperability aggregation.

The semantic interoperability capability of the coupled sub-ontologoy O_i $(i = 0 \sim n)$ in CO refers to the semantic interoperability capability between m service models annotated by O_i. Fig.6 shows the service software encapsulated by ontology and metamodeling techniques[3]. O_i is the common agreement of semantic interoperability between m services. The commonly covered area (partial semantic interoperability) of O_{i_1} and O_{i_2} is the percentage (with weight) compared with O_i. For O_i, the combination of partial semantic interoperability capability $(O_{i_1} \sim O_{i_m})$ of the m service models should fully cover O_i. The partial semantic interoperability aggregation from multiple services to O_i can realize the agreed semantic interoperability capability of behavior collaboration.

For example, the ontology graphs of O_{i_1} and O_{i_2} are totally overlapped with each other, namely the semantic distance between them is 1. But the overlapped area is only a small part of O_i, hence the partial semantic interoperability capability between them is very low.

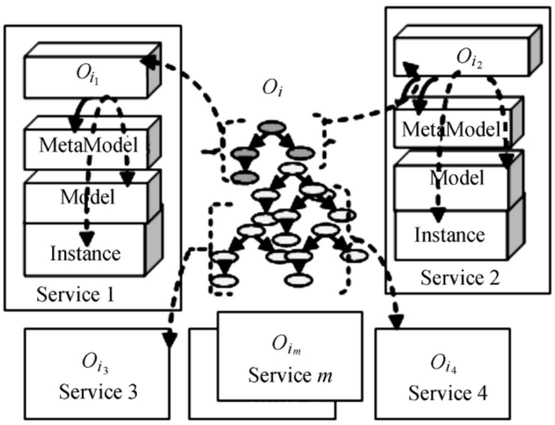


Fig.6. Partial semantic interoperability aggregation based on coupled sub-ontologies O_i.

The semantic distance and similarity between models are necessary and fundamental methods for the measurement of semantic interoperability capability, but they are not sufficient.

There are two key parameters in the process of mining and selecting potential agreement for the realization of semantic interoperability of O_i:

1) *Threshold of Semantic Distance* K_i

For the coupled ontology O_i, it requests that two service ontologies are similar semantically. The threshold of semantic distance K_i $(K_i \geqslant 1)$ needs to be set. If the semantic distance of two service ontologies is within the interval $[0, K_i]$, then it implies that these two service ontologies satisfy the basic condition for semantic interoperation, and can act as candidate services for semantic interoperability aggregation.

2) *Threshold of Semantic Interoperability Capability* $IC_{K)i}$

For the candidate services for semantic interoperability aggregation identified in 1), we should set the threshold of semantic interoperability capability between two service ontologies IC_{K_i} $(IC_{K_i} \in [0, 100\%])$ based on the agreement of semantic interoperability capability, the quality of aggregation modeling and OoS requirements (e.g., increasing the number of services in aggregation will decrease the system quality), etc. If the semantic interoperability capability of two service ontologies is within the interval $[IC_{K_i}, 100\%]$, then it implies that these two service are capable to realize the agreement of O_i semantic interoperability, and can be the service participants of semantic interoperability aggregation.

Aspect 3: *Meaning Interoperability Aggregation at CO Level*

The meaning interoperability aggregation at the CO level is based on the control structure in RSO, which can realize the behavior collaboration between the coupled O_i by acquiring behavior of O_i under the constraints of context and users preference. We consider following factors in the meaning interoperability aggregation of the CO level in order to improve the QoE (Quality of Experience) of users, and simulate the group behavior emergence in complex software systems: the collaboration by users preference and context constraints; the intensity of interoperability aggregation which denotes the degree of interoperability aggregation among the process units under different control structures.

RSO can invoke and coordinate the behavior among service ontologies O_i $(i = 0 \sim n)$ based on the control structures in RSO. Different control structures result in the difference in the intensity of interoperability collaboration. We consider several basic control structures: *sequence*, *choice*, *split-join*, *any-order*, and *loop*. *Sequence* refers to the sequential execution of a series of collaboration units; *choice* refers to the selection of one collaboration unit from a series of collaboration units; *split-join* refers that a series of collaboration units execute in parallel, and the control structure completes when all the collaboration units complete; *any-order* refers to a random execution sequence of a series of collaboration units, in which all the collaboration units will be executed in a none-parallel order; *loop* refers to the repetitive execution of a collaboration unit when a condition is hold. For these control structures, the services collaboration by *loop* control structure requires a restrictive condition dependency between services, which has a highest intensity of interoperability collaboration; the services collaboration by *sequence* control structure also requires data and condition dependencies between services, which has a comparably higher intensity of interoperability collaboration; the intensity of interoperability collaboration for *choice* and *split-join* control structure decreases in turns; the services collaboration by *any-order* control structure has no any data and condition dependency between services, which has the lowest intensity of interoperability collaboration.

We configure the intensity coefficient of semantic behavior collaboration by different control structures, and classify the collaboration modules based on the intensity. This step can assist the calculation of the intensity value of semantic behavior collaboration in CO.

First, define the intensity coefficient of semantic behavior collaboration of different control structures.

Suppose that the intensity coefficient for the *sequence* control structure is 1, and then the intensity coefficient for the *loop* control structure is 2 since *loop* control structure is concerned with bidirectional condition dependency, and *sequence* control structure is only related with unidirectional control and data dependency. The intensity coefficients for the *choice*, *split-join* and *any order* control structure are set to 0.5, 0.5 and 0.1 respectively.

Second, classify the collaboration modules based on control structures. The classification criterion is that there is only one type of control structure in a collaboration module. Then calculate the collaboration intensity of collaboration units in collaboration modules, in which one collaboration module can be an atomic process or a compositional process. For an atomic process, the collaboration intensity is 1; for a compositional process, the collaboration module also includes control structures and other collaboration units, then it is regarded as a collaboration module. The collaboration intensity of this collaboration module should be calculated recursively.

The behavior (meaning) collaboration intensity value at CO level is:

$$S_{\mathrm{CO}} = \sum S_i \tag{10}$$

$$S_i = p_i \sum S_{ij} \tag{11}$$

in which, S_i represents the COLLABORAtion intensity of the i-th collaboration module, S_{ij} represents the collaboration intensity of the j-th collaboration unit in the i-th collaboration module, p_i represents the collaboration intensity coefficient of the i-th collaboration module, which depends on the type of control structure in this collaboration module.

5 Case Study

In our opinion, domain-orientation and user requirements domination are the basic guidelines in SORE. The requirements engineering process is a user-centric process, and is composed of several activities including requirements elicitation, analysis, validation and modeling based on domain knowledge. In this section, we provide a simple requirements use case to show the semantic interoperability aggregation in requirements refinement based on CO.

The user requirements described in the use case presented in this section is about the route planning from the domain of urban transportation. The detailed information of this use case is described below: *Smith plans to watch the Olympic men's football final match at Olympic main stadium (the Bird Nest) in the noon of 23rd, August. He needs to query the location of the Bird Nest, and weather information of that day to decide whether or not to take rain gear, reserve a place at a western restaurant for the lunch, arrange the departure time and an economic route to the Bird Nest according to the match time and duration on the road, and display the route notification in a short message by mobile.*

5.1 Prerequisites of Experiment

Two prerequisites should be met in order to use this requirement use case to describe the dynamic construction of CO and semantic interoperability aggregation:

1) The development of core assets for the route planning problem from the domain of urban transportation. The customized domain-oriented core assets in the problem domain include: route planning DPO (as shown in Fig.7), and corresponding services with semantic annotation.

2) There are a large number of existing service ontologies, and service resources in the Web, e.g., service ontology for scenic spots, service ontology for weather forecast.

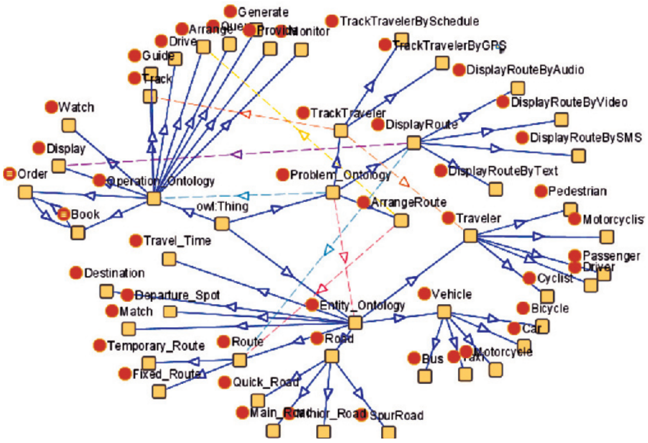


Fig.7. Route planning DPO in transportation domain.

5.2 Construction of RSO

Construction of RSO can be achieved by the matching to the route planning DPO from urban transportation domain based on the requirement use case described above. The detailed steps are specified as follows.

1) *Syntactic Analysis.* First, perform the syntactic analysis to the requirement, identify the entity concepts (nouns) and operation concepts (verbs), and annotate them (in this subsection, we use bold font to represent the entity concept, and italic font for the operation concept). "**Smith** plans to *watch* the **Olympic men's football final match** at **Olympic main stadium (the Bird Nest)** in the noon of **23rd, August**. He needs to *query* the **location** of the Bird Nest, and the **weather information** of that day to decide whether or not to take rain gear, *reserve* a place at a **western restaurant** for the lunch, *arrange* the **departure time** and an economic **route** to the Bird Nest according to the **match time** and **duration on the road**, and *display* the **route notification** in a short message by mobile."

2) *Semantic Annotation and Matching.* The DPO of route planning problem from urban transportation domain is employed to semantically match and annotate the operation and entity concepts identified in the requirement. For example, "Smith" is annotated as an instance of "Person", "the Bird Nest" is annotated as an instance of "Stadium". We can get a more understandable ontology model using domain ontology and the matched core services based on this semantic annotation.

3) *RSO Refinement.* Generally, only part of entities of the requirement can be matched to the problem domain ontology. The system analyst needs to further customize the use case using requirements elicitation and analysis tools, and add the concepts and their associations, which are not matched in the previous step (as shown in the dashed box in Fig.8), in order to construct the RSO as shown at the bottom of Fig.8.

5.3 Dynamic Construction of CO Based on RSO

CO is dynamically constructed by connecting the

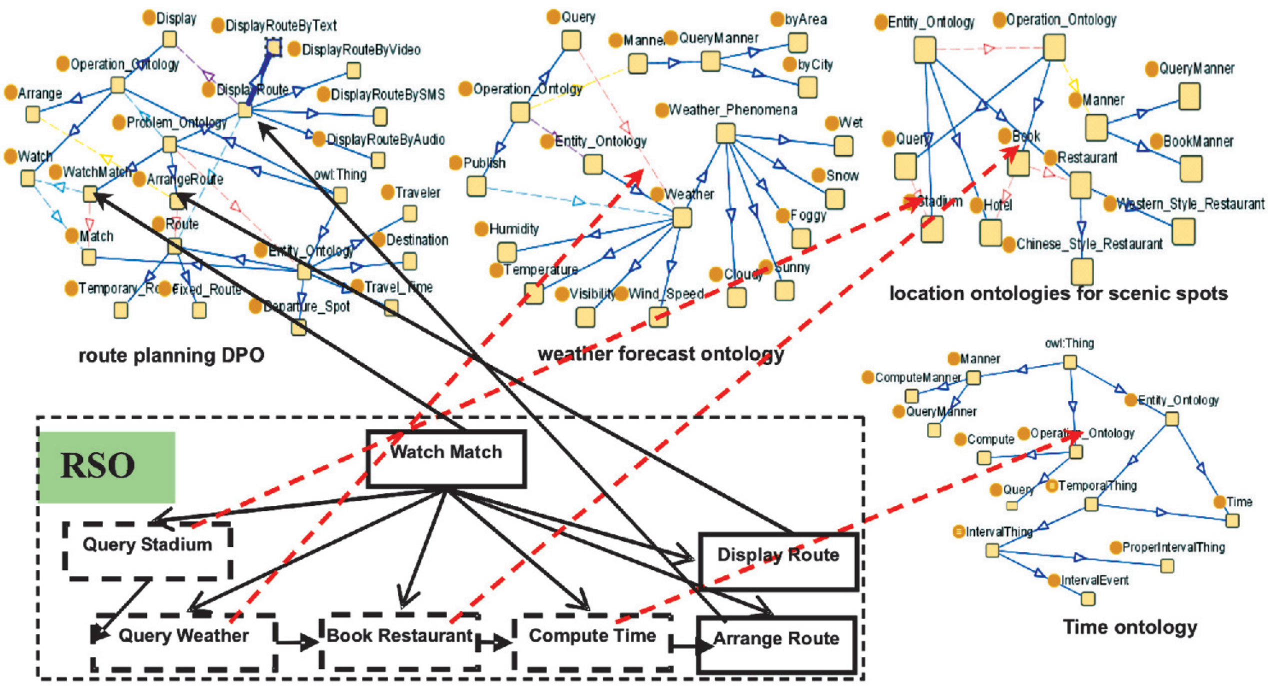


Fig.8. Dynamically generated CO based on RSO.

sub-ontology O_0 of DPO and service ontologies O_i ($i = 1 \sim n$) in other domains and using RSO as a core ontology. During the process of selecting O_i ($i = 1 \sim n$), all matched concepts and their directly related nodes are retrieved starting from the root node in the ontology graph based on the concept of ontology partition[17].

1) *Tightly Coupled Connection Between RSO and DPO*

Since software requirements are usually domain-oriented, thus the tightly coupled connection between RSO and DPO can be predefined, as shown by the solid arrow in Fig.8. In this route planning example, arranging and displaying route both can be matched directly from the route planning DPO.

2) *Loosely Coupled Connection Between RSO and Service Ontologies in Other Domains*

The part of RSO, which cannot be matched directly by DPO (as shown by the dashed box in Fig.8), has a loosely coupled connection to the service ontologies in other domains, and this connection is a dynamic binding as shown by the dashed solid arrow. In this example, the concept of querying stadium, querying weather, reserving restaurant, and calculating departure time requires the matching to the service ontologies in other domains, such as service ontologies for weather forecast, location service ontologies for scenic spots.

The objective of the connection between RSO and the service ontologies in other domain is to get the behavior provided by the service composition of matched service ontologies, for example, the behavior provided by service ontologies for weather forecast, location service ontologies for scenic spots. The compositional service behavior of user requirement use case can be satisfied by the dynamic construction of CO through the RSO control structures.

5.4 Analysis of Semantic Interoperability Aggregation in CO

We describe semantic interoperability aggregation in this route planning example in three aspects.

Aspect 1: *Semantic Matching Between Ontologies*

First, perform the semantic matching for all the structure concepts using the semantic association of the structural concept mapping between ontologies presented in Aspect 1 of Section 4, and then calculate the semantic matching capability between ontology models.

We use natural language to specify the requirements in this example. The syntactic/semantic matching rate is 30%, and the 30% matched elements act as a basis for the construction of RSO. The syntactic/semantic matching rate between the non-matched part in RSO and the service ontologies for weather forecast is 40%. The syntactic/semantic matching rate between the non-matched part in RSO and the location service ontologies for scenic spots is 15%. The syntactic/semantic matching rate between the non-matched part in RSO and the time service ontologies is 20%.

The semantic matching between ontology models is a simple semantic interoperability context. It focuses on the semantic similarity of two parties without the agreement on the collaboration capability of both parties. The semantic similarity is the basic condition for the realization of semantic interoperability.

Aspect 2: *Semantic Interoperability Aggregation in Coupled Ontology O_i*

We focus on the partial semantic interoperability aggregation between multiple service ontologies in the coupled ontology O_i of CO based on the semantic matching of Aspect 1. The partial semantic interoperability aggregation is the key element for the dynamic aggregation modeling of multiple service models annotated by service ontologies.

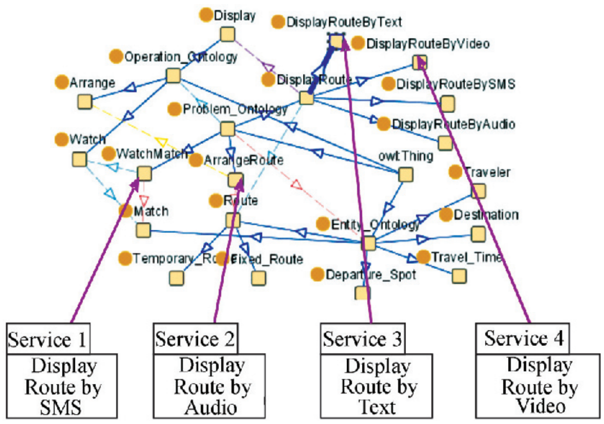


Fig.9. Partial semantic interoperability aggregation.

In this example as shown in Fig.9, we get the route planning sub-ontologies, the semantic interoperability aggregation between annotated service ontologies, and aggregation between service models by semantic matching in Aspect 1.

The upper of Fig.9 is the route planning sub-ontologies by semantic matching, and bottom half of Fig.9 is the service models annotated by these sub-ontologies. The service models in the Fig.9 are the service assets customized for the route planning problems based on O-RGPS[5]. We set $K_i = 3$, $IC_{ki} = 55\%$. Using the formulae for calculating semantic distance and semantic interoperability capability defined in Aspect 2 of Section 4, we can select the services whose semantic distance between service ontologies is within the interval [0, 3], and whose semantic interoperability capability between service ontologies is within the interval [55%, 100%] among all the available services. The resulting services are *Display Route by SMS*, *Display Route by Audio*, *Display Route by Text*, *Display Route by Video*, the four services can participate the semantic interoperability aggregation of the route planning sub-ontologies. In other words, the four services can satisfy the semantic agreement of the route planning sub-ontologies, and can realize the capability of semantic interoperability and aggregation service for the requirements semantically represented by the route planning sub-ontologies.

For other coupled ontologies O_i $(i = 1 \sim n)$ in CO, we can also get partial semantic interoperability aggregation. For example, the sub-ontology for weather forecast O_1 is aggregated by the partial semantic interoperability aggregation of four services (*Query Weather*, *Query Weather_Phenomena*, *Query Temperature*, *Query Humidity*); the sub-ontology for scenic spots O_2 is aggregated by the partial semantic interoperability aggregation of four services (*Query Stadium*, *Query Restaurant*, *Book Hotel*, *Query Restaurant*); the time sub-ontology O_3 is aggregated by the partial semantic interoperability aggregation of three services (*Query Time*, *Query IntervalTime*, *Compute IntervalTime*).

Aspect 3: *Meaning Interoperability Aggregation at CO Level*

Meaning interoperability aggregation at the CO level is the behavior got from the coupled ontologies O_i ("*query weather*" ontology O_1, "*query stadium*" and "*query restaurant*" ontology O_2, "*time*" ontology O_3), and the service aggregation in "route planning sub-ontologies" O_0 as shown in Fig.10. These behaviors construct the service behavior which satisfies the user requirements based on the behavior (meaning) aggregation of RSO at the CO level.

The RSO control structures for this use case is shown in Fig.10: "*query weather*", "*query stadium*", "*reserve restaurant*" are associated by control structure *any-order*; and then they are sequentially connected with "*calculate departure time*", then sequentially connected with "*plan route*", and finally sequentially connected with "*display route*".

Then we can calculate the semantic behavior collaboration intensity of CO by (10) and (11). The RSO in this example includes two types of control structures: *any-order* and *sequence*. The process of the use case is divided into two process modules by these two types of control structures. The first process module is composed of three process unites "*query weather*", "*query*

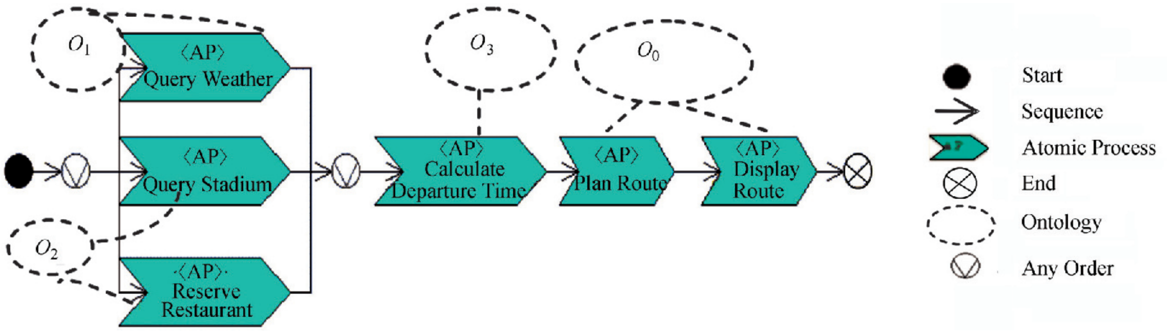
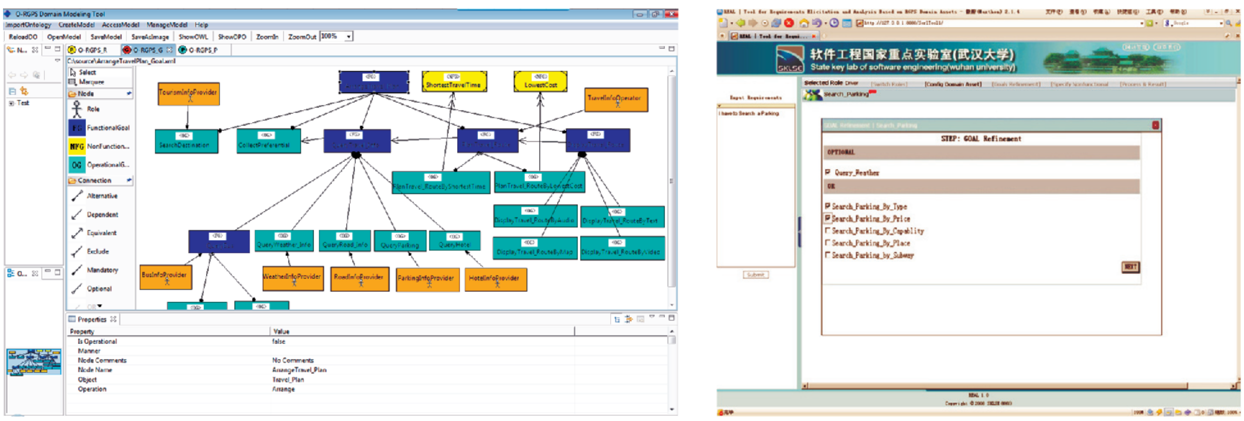


Fig.10. Meaning interoperability aggregation based on RSO.



(a) (b)

Fig.11. Snapshots of corresponding tools. (a) Snapshot of domain modeling tool. (b) Snapshot of requirements elicitation and analysis tool.

stadium" and "*reserve restaurant*" connected by *any-order* control structure, and the three process unites are all atomic processes, hence their collaboration intensity value are all 1. The second process module is composed of three process unites "*calculate departure time*", "*plan route*", and "*display route*" connected by *sequence* control structure, and the three process unites are also atomic processes, hence their collaboration intensity values are also 1. The semantic behavior collaboration intensity value of the CO, with which the RSO is related, is: $S_{\rm co} = 0.1 \times 3 + 1 \times 3 = 3.3$.

In addition, we have developed corresponding toolkits to support service requirements refinement, including ontology-based domain modeling tool[5], service registration tool[18], and requirements elicitation and analysis tool[19]. Figs. 11(a) and 11(b) show the snapshots of domain modeling tool and requirements elicitation and analysis tool, respectively.

6 Conclusion and Future Work

This paper proposes a method of semantic interoperability aggregation for requirements refinement based on interoperability method of connecting ontologies. We present the method for creating requirements-driven CO, which is the semantic guidance for the dynamic construction of service-oriented software systems, and the semantic agreement for service semantic interoperability. We also present a framework and a calculation method for the measurement of semantic interoperability capability. Finally, we discuss the method and process for creating CO, and applications of semantic interoperability aggregation based on CO through a concrete case study.

We are in an age of collaborative software engineering[20]. In this era, the interoperability theory and method will be the core issue in the software engineering research. With the rapid growth of large software systems in scale and complexity in the network environment, the self-organized behavior and the interaction of software units become stronger and more intensive. In our future work, we will further investigate the theory of semantic interoperability aggregation in the self-organization of networked software complex systems, and the adjustment mechanism of semantic interoperability aggregation; the theory and method for stimulating or restraining the swarm behavior of software units. All these researches will ultimately contribute to the development of software engineering in the network age.

References

[1] He K, Peng R, Liu W *et al.* Networked Software. Science Press, China, 2008. (in Chinese)

[2] Karim S, Latif K, Tjoa A M. Providing universal accessibility using connecting ontologies: A holistic approach. In *Proc. the 4th Int. Conf. Universal Access in Human Computer Interaction*, Beijing, China, Jul. 22-27, 2007, pp.637-646.

[3] Cregan A. W3C semantic Web ontology languages: OWL and RDF tutorial. In *ISO/IEC JTC1 SC32 11th Open Forum on Metadata Registries*, Tutorial, Sydney, Australia, May 19-22, 2008.

[4] He K, He Y, Wang C *et al.* Ontology and Metamodeling Theory, Method and Applications. Science Press, China, 2008. (in Chinese)

[5] Wang J, He K, Gong P *et al.* RGPS: A unified requirements meta-modeling frame for networked software. In *Proc. the 3rd International Workshop on Advances and Applications of Problem Frames* (*IWAAPF 2008*) *at 30th International Conference on Software Engineering* (*ICSE 2008*), Leipzig, Germany, May 10-18, 2008, pp.29-35.

[6] Stabb S, Studer R (eds.). Handbook on Ontologies. International Handbooks on Information Systems, Springer, 2004.

[7] Heflin J, Hendler J. Semantic interoperability on the Web. In *Proc. Extreme Markup Languages 2000*, Alexandria, USA, Aug. 15-18, 2000, pp.111-120.

[8] SIM-TBASSC. http://www.isi.edu/tbassco/index.html, Oct. 15, 2008.

[9] Pierre L, Jean M R *et al.* A Deployment and Research Roadmap for Semantic Interoperability: The EU SemanticHEALTH Project. eHealth Beyond the Horizon — Get IT There, Andersen S K *et al.* (eds.), 2008, pp.635-640.

[10] Cleverdon C. The Cranfield Tests on Index Language Devices. Morgan Kaufmann Multimedia Information and Systems Series, Morgan Kaufmann Publishers Inc., 1997, pp.47-59.

[11] Ehrig M, Euzenat J. Generalizing precision and recall for evaluating ontology matching. In *Proc. the 4th International Semantic Web Conference* (*ISWC*), Poster Session, Galway, Ireland, Nov. 6-10, 2005, PID-54.

[12] Euzenat J. Semantic precision and recall for ontology alignment evaluation. In *Proc. the 20th International Joint Conference on Artificial Intelligence* (*IJCAI*), Hyderabad, India, Jan. 6-12, 2007, pp.348-353.

[13] Budanitsky A. Lexical semantic relatedness and its application in natural language processing. Technical Report CSRG390, University of Toronto, 1999.

[14] Budanitsky A, Hirst G. Semantic distance in Word-Net: An experimental, application-oriented evaluation of five measures. In *Proc. the NAACL 2001 Workshop on WordNet and Other Lexical Resource*, Pittsburgh, USA, Jun. 3-4, 2001, pp.29-34.

[15] Liang P, Jansen A, Avgeriou P. Sharing architecture knowledge through models: Quality and cost. *The Knowledge Engineering Review*, 2008, 23(4): 225-244.

[16] Ganesan P, Garcia-Molina H, Widom J. Exploiting hierarchical domain structure to compute similarity. *ACM Transactions on Information Systems*, 2003, 21(1): 64-93.

[17] Seidenberg J, Rector A. Web ontology segmentation: Analysis, classification and use. In *Proc. the 15th International Conference on World Wide Web*, Southampton, UK, May 11-14, 2006, pp.13-22.

[18] Zeng C, He K, Yu Z *eta al.* Towards improving Web service registry & repository model through ontology-based semantic interoperability. In *Proc. the 9th International Conference on Grid and Cooperative Computing*, Shenzhen, China, Oct. 24-26, 2008, pp.747-752.

[19] Chen H, He K, He Y *et al.* An automated approach for requirements driven model selection. In *Proc. the 1st International Workshop on Data and Semantic Engineering for Emerging Technology and Services*, Busan, Korea, Nov. 11-13, 2008, pp.1052-1057.

[20] Ivan M *et al.* (eds.). Collaborative Software Engineering. Springer, 2009.

Ke-Qing He received his B.S. degree in mathematics from Wuhan University in 1970 and Ph.D. degree in computer science from Hokkaido University of Japan in 1995. He is a professor in State Key Lab of Software Engineering (SKLSE) and the director of Software Engineering Institute at Wuhan University. His research interests include service-oriented software engineering, networked software engineering based on complex system, service-oriented requirements engineering, and software engineering technical standards for interoperability. He is now a senior member of IEEE and CCF, and Vice Director of CCF TCSC (Technical Committee on Service Computing), and an editor of ISO/IEC JTC1 SC32. Until now, he has published over 120 papers and 4 monographs, and developed ISO standard ISO/IEC 19763-3.

Jian Wang received his B.S. degree from University of Jinan in 2002, M.S. degree from Guizhou University in 2005, and Ph.D. degree from Wuhan University in 2008. He is a lecturer in the State Key Lab of Software Engineering (SKLSE) at Wuhan University. His main research interests include requirements engineering, service registration, and semantic interoperability. He is a member of CCF.

Peng Liang received his B.S. degree in computer science from Wuhan University of Technology in 2000, and M.S. degree in computer science (software engineering) from Vrije Universiteit Brussel in 2002, and Ph.D. degree in computer science (software engineering) from Wuhan University in 2006. He is an associate professor in the State Key Lab of Software Engineering (SKLSE) at Wuhan University. His research interests include software engineering, semantic-enabled requirements engineering, software architecture, and software interoperability. He is a member of CCF, IEEE and ACM. He has published over 40 publications, including five book chapters and one co-authored book.