Service Composition Engine Based on Service-ontology

Liquan Han¹², Shufen Liu¹, Lu Han¹, Zhilin Yao¹

¹College of Computer Science and Technology, Jilin University, ChangChun, P.R. China
²College of Computer Science and Engineering, ChangChun University of Technology, ChangChun, P.R. China

asehan@163.com; liusf@jlu.edu.cn; yaozl@jlu.edu.cn

Keywords: service-ontology, service dynamic composition, smart transcripts, Service-Oriented Architecture (SOA).

Abstract

With the rapid increasing of the number of services in Web environment, single or simple composition of the service is unable to provide high-quality service for requesters. Although dynamic service composition provides the automatic composition methods for machines, it cannot handle complex service requests. The main objective of researching Semantic Web is to extend the current World-Wide-Web, which makes all the information in Internet semantic. This paper arms to analyzes the main features of the Web Service. Then, with the definition and application of ontology, we propose the concept and definition format of service-ontology, and with the construction of service composition and corresponding algorithm, we apply it in the dynamic Web service composition application. Combined with intelligent smart transcript repository and ontology knowledge repository, the architecture provides the static and dynamic composition environments to facilitate knowledge accumulation, logic inference, binding web service and creation of process driven model. The approach considers the semantics of services along with the quality and the efficiency of the composition. Finally, based on the prototype system to validate the approach, the results of experiment prove that the Service Composition Engine is feasible and effective.

1 Introduction

Web Service is a new model of distributed Web applications, and an effective mechanism of integrating data and information. From a technical perspective, Web services can be considered to be the objects deployed on the Web, and is the self-contained, self-described and modular program. With standard XML message technology encapsulating information, the Web service can be visited, found and called to complete specific tasks through the Internet[1,2].

With the increasing maturity of technical services, more and more stable, easy-to-use Web services are shared on the network. However, single service provides limited service, and it is necessary to synthesize shared services to increment services, thus provide powerful service function and better meet the needs of service requesting. Web service composition refers to the process of combining several Web services to provide a value-added service. It is emerging as the technology of choice for building cross-organizational applications on the Web[1,3]. Web service composition has recently taken a central stage as an emerging research area. Several techniques have been proposed[4,5,6]. Standardization efforts are under way for supporting Web service composition (e.g., BPEL4WS[7]). However, these techniques and standards provide little or no support for the semantics of participant services, their messages, and interactions. Additionally, they generally require dealing with low level programming details, which may lead to unexpected failures at run-time. The main objective of researching semantic web is to extend current World-Wide-Wed, and thus makes all the information in the Internet semantic, and able to be understood and handled by computer and facilitate the interaction and cooperation between people and computers. Ontology is a formal, explicit specification of a shared conceptualisation[8]. It greatly accelerates the research of semantic web. Meanwhile, we found that it would greatly promote the development of web service dynamic composition technology, with the integration of semantic idea to web service.

Based on the analysis of web service reference model on the structure of SOA and the Semantic Web Ontology model, this paper proposes the definition of service ontology model and applies the service ontology to web service dynamic composition technology, build Service-Composition-Approach, which achieve the service composition, service detection and service matching and improved the precision and the recall to some extent.

2 Framework of the Service Composition Based on Service-ontology

With the release and gradual adoption and realization of core web service standards (for example: SOAP, WSDL, UDDI), the interoperability has progressed to some extent between heterogeneous software systems. Despite some restrictions of SOA based on web services, it is the promising composition of structure and technology. With consistently robust and reusable function, such service will not only meet business needs nowadays, but can be applied to ever-changing business needs [9].

Combined with the service meta-model framework and the definition of service-ontology model, we propose the whole framework architecture of web service dynamic composition on the basis of service-ontology (e.g. Figure 1). This framework can support any programming languages and the running platforms. It has two participants, one is service provider and the other is service requester. Service
 providers can provide services for the composition, and service requester makes use of the service and information from the service provider. Apart from these two participants, the general framework consists of three management parts: service Ontology Manager, service composition manager and intelligent transcript manager.

1. Service Ontology Manager: It is the key part to represent knowledge. There are three tiers in service ontology. With corresponding reasoning machine to realize analysis and inference instead of manual work on each tier, it can meet the needs of service ontology dynamic composition. The first tier ontology are known as the Common Ontology, which includes the general knowledge of human daily life and are used to understand and translate the requests from the service requester (such as: human language). In the achieving process, FOL are used to express knowledge. The knowledge base contains simple declaration, reasoning rules and reasoning control rules, and thus can generate new inference with the reasoning machine. The second tier ontology are known as the Domain Ontology, which are used to express the concept, taxonomy, relationship and rules in some top domain. The domain ontology is used to response applications from different fields and thus meet their requirements. The third tier ontology is service ontology. They are used to provide the shared knowledge representation of concepts and attributes in the specific areas of the Web service.

2. Service Composition Manager: it is the center of the framework. The core part is service composition engine. It is used to receive the requests from service requester, to translate and transform them to the internal identifiable format of the service composition manager. With the service from service ontology manager and intelligent transcript manager, it aims to provide the best service for the requester. If the service combined successfully, the engine would transmit the result of static or dynamic composition to the service composition module. The service composition module is in charge of the interaction with the service base. It will complete the final service matching and send the description of service matching to the service requester and thus provide services. The working process and designing methods of the composition engine are depicted in 3.3.

3. Smart Transcript Manager: it is responsible for generating and maintaining “Smart Transcripts”. Transcripts are not necessary “scripts”. They can be database entries as well. Transcripts are used to “record” information about the activity of the service composition, such as states, action items, important milestones, etc.. Transcripts could be used to make references to past events. All past service composition documents and files are stored in the transcript knowledge base. The transcript management module is responsible for managing the transcript, including: Build the transcript files; Store the transcript files into database; Export the transcript from database.

3 Definition Model

In the framework structure illustrated in figure 1, there are three main aspects: service ontology concept model, dynamic composition concept model and dynamic service engine based on service ontology. With the three mechanisms of communication, cooperation and coordination, the engine integrates the service ontology concept model and dynamic composition concept model into a whole. It is the center of main framework structure, and is the center of service ontology dynamic composition model designing and working.

3.1 Service-ontology Conceptual Model

The central task of service modelling on the basis of ontology is to establish shared collection of services vocabulary. Service ontology can be regarded as a simple description of service terms and their management. As for the concept and role in ontology, the concept is an abstract class and can be described with the FOL. Services are categorized as atomic service, the services that are able to decompose into more fine-grained, and composite service, the services that are combined through a number of services [9]. The composite service is formed by the atomic service, role and construct identifier, according to certain rules. The role is a relationship class of abstract class, and can be described by SOL. The role itself can also be composed.

Definition 1(Ontology Model): Ontology model is a triple in the form of \( O = \langle T, H, X \rangle \), which is referred as the ontology. \( T \) is the collection of terminology, and the term in \( T \) is
referred as the atomic term, including atomic term C
(Shortened form: atomic class) and atomic attribute term P
(Shortened form: atomic attribute), represented as T=<C,P>.
According to the value range, there are two kinds of the
ontology attributes: class attributes and data attributes. Class
attributes represent the relationship among classes and data
attributes are used to present the attributes of class. H is the
inherited relationship collection of term T, including class
inheritance and attribute inheritance, namely the subClassOf
and subPropertyOf. X are ontology rule set, or ontology
constraint set, and can be formally expressed with First-
Order Predicate Logic or Description Logic.

Definition 2(Service-ontology Model): Service ontology
model are often referred as the service ontology. It is a triple,
expressed as S=<E, O, R>. E is event or action; O is the
ontology model referred by the service ontology; R is the
description of rules. The relationship of E and O is a kind of
reference. The basic ontology concept is illustrated in O.

Definition 3(Concept Model): In the service ontology, if
two concepts Ci and Cj, and if Ci is the equivalentClass of Cj,
then Ci and Cj are semantic equivalent, noted as Ci=Cj; if Ci
is the subClassOf of Cj, then Ci semantic includes Cj, noted as
Ci⊂Cj.

Definition 4(Semantic Similarity): semantic similarity of
service ontology concepts is described as follows:
\[ S(C_i, C_j) = \frac{a}{d + a} \]  
(1)

In the formula, a is an adjustable parameter, and d is an
integer. For the convenience of computing, we set the rules: if
Ci=Cj, then d=0; if Ci⊂Cj or Cj⊂Ci, then d=1. If the concept
collection CA(CA1,CA2,....,CAm) and
CB(CB1,CB2,....,CBn) are CA=CB, then the semantic
similarity are expressed as:
\[ S(CA, CB) = \frac{1}{n} \sum_{i=1}^{n} \max_{j=1}^{m} S(CA_i, CB_j) \]  
(2)

3.2 Dynamic composition conceptual model

The web service composition uses existing web service
module to form large-grained and complex functional web
service, through the prescription of their way of execution, in
order to meet the business requirements [10]. The interface of
service is used to transmit the information of control and data
between services. Therefore, service composition based on
interfaces is illustrated in [11, 12]. It is mainly used to match
the input and output parameters between services, so as to
realize dynamic service composition. Based on the thought of
expressing a service with the input and output entities, and
with the needs of this article, the web service is formalized as
follows:

Definition 5(Service): A web service can be expressed as
follows: WSi(li, Oi). WSi is the name of web service; li is the input set of
the service and Oi is the output set of the service.

Definition 6(Service Request): A web service request
can be expressed as follows: WSRk(Ik, Ok). WSRk is the name
of service request; Ik is the input set of service request and Ok
is output set of service request.

Definition 7(Semantic Associated1): For two service
WSi(li, Oi) and WSj(lj, Oj), if Oi⊂lj, then WSi and WSj are
semantic associated, noted as WSi↔WSj, and their semantic
association degree are expressed as AD(WSi, WSj)=S(li, Oi),
where WSi is the precedent service of WSj , and WSj is the
subsequent service of WSi.

Definition 8 (Semantic Associated2): For the service
request WSRk(Ik, Ok) and service WSi(li, Oi), if li⊂Ik, then
the WSRk is semantic associated with the service WSi(li, Oi),
noted as WSRk→WSi, and its semantic association degree are
expressed as AD(WSRk, WSi)=S(li, Oi), where WSi is the
subsequent service of WSRk; if Oi⊂Ok, then the service WSi
are semantic associated with the service request WSRk, noted as
WSi→WSRk, and the semantic association degree are
expressed as AD(WSj, WSRk), where WSi is the precedent
service of WSRk.

Definition 9 (Service Composition): A service
composition can only meet a service sequence of service
request WSRk : WSI, WSi2, ..., WSn. The sequence satisfies
the following three conditions:
(1) WSRk→WSi
(2) Any two adjacent service WSi and WSi+1 are
WSi→WSi+1
(3) WSn→WSRk

Definition 10 (Service Composition Satisfaction
Degree): The satisfaction degree refers to the extent of
service composition satisfy some service request, noted as
SAT. For the service request WSRk, a service composition
satisfaction degree are expresses as:
\[ SAT = AD(WSR_k, WSI) \times \prod_{i=1}^{n-1} AD(WS_i, WS_{i+1}) \times AD(WS_n, WSR_k) \]  
(3)

3.3. Execute Process of Composition Model

On the basis of analysis automatic service composition
framework, and with the service ontology model and service
composition model defined in this paper, we implement the
composition engine as figure 2:
In figure 2, the service composition engine enhances the existing service (service composition) operation mechanism with the service ontology semantic inference capability and intelligent transcript, which can integrate existing web service into the framework seamlessly. After defining some semantics, the framework can provide the ability of dynamic service composition based on service ontology. The composition engine has the following components: service converter, service ontology process generator, service assessment recommendation module, and service execution engine. Service converter can translate the external language of service requester to the internal language used in service ontology process generator. For each request, service ontology process generator will provide a set of service scheme, which could be dynamic or static. If there are more than one service scheme, the service assessment and recommendation module will evaluate these schemes and select the one which best meets the requirements of requester. The service execution engine will carry out the selected scheme and return the outcome to the requester.

1. The diversity of service release: the service provider releases their service in a global service center. A number of different languages can be used, for example, UDDI and OWL-S ServiceProfile. In the process, the main attributes of Web service description are required, such as signature, state and non-functional value.

2. Adaptation of service request format: The composition engine uses the service converter to translate the external descriptive language to internal specification language. Service ontology process generator applies formal language to describe service, including descriptive logic to express FOL, and carry out service ontology reasoning and application. The service requester can employ their familiar language to express service request. At present, popular Web service standards are WSDL and OWL-S.

3. The generation of composition process model: The service requester applies a kind of service specification language to express their requirements. The process generator search available service composition from service storage, based on knowledge inference and service dynamic matching composition algorithm, trying to resolve these requirements. Functions are often used as the input of the process generator, and process model describing composite service are outputs. The process model consists of a set of selected atomic service, and the control flow and data flow of these services.

4. Assessment of composite service: There are a number of same of similar functions in the service storage, thus it is possible that more than one composite service are produced by the process generator, to meet the requirement. In such case, the service assessment and recommendation module choose the process by evaluating the non-functional attributes information. Generally speaking, the requester designates a non-functional attributes power, and then assessor select the best option according to the weight sequence.

4 Main Implementation Algorithm

Based on the general framework, we use Java to implement the service engine, and provide the service storage key testing algorithm with the UDDI service registration center, which achieve certain effect. We mainly describe the knowledge inference process and service dynamic matching algorithm here.

4.1 Knowledge Reasoning Process

Based on the three-tier ontology in service ontology manager, service ontology process generator would start the service, combined with the service ontology model and dynamic service composition model. Its main service process is (1) the requester (user of program) submits the service request to the composition engine, and the engine could analyze and translate the request into the formally representation for targeted areas. (2) Combining the relevant domain ontology to reason, aimed at the request objectives and possible constraints, and to generate a request logic in the form of certain criterion; may request binding, said the request generated a certain standard logic (3) With the request logic and Web service domain knowledge, use the service ontology inference mechanism, detection mechanism and recommendation mechanism to search and choose the web service able to complete the request, and then form the ultimate composition logic with the service dynamic selection, return the result after binding.

In the service process, the ontology-based knowledge inference is the core of service management and is the basis of web service dynamic composition. We use Jena as the reasoning machine to implement the knowledge reasoning in the process.

4.2 Algorithm of Service Dynamic Matching and Composition

Preparations:

According to the UDDI Yellow Pages to describe Web Services, and with the improved UDDI model and information of entrance parameters ontology, the Web Services are divided into different regions. The example of web service release directed graph are illustrated in figure 3.

**Algorithm 1: To construct the web service release demonstration picture**

(1) Establish root node $WS$, and assign a node $WS_i$ for each released web service (Node number can be assigned in sequence and figure 3 numbered according to levels for convenience);

4.3 Figure 3 demonstration of web service release
(2) According to definition 7, check the $WS_i(I_i, O_i)$ and $WS_j(I_j, O_j)$, if $WS_i \supseteq WS_j$, then draw a complete matching directed line from $WS_i$ to $WS_j$; if $WS_i \supseteq WS_j$, then draw an ontology semantic matching line from $WS_i$ to $WS_j$.

(3) Check all the remaining service where input degree is zero(isolated single node), if it is the service has not been checked then go to step (2), if not, go to step (4);

(4) Establish the directed line from root node $WS$ to all the nodes with zero input degree.

According to the algorithm above, figure 3 is constructed with the registered service in accordance with the service ontology composition. It is the foundation to run service detection matching algorithm.

Assuming the current service requests is $WSR_i(I_i, O_i)$.

Algorithm 2 : dynamic detection and matching algorithm based on semantic service ontology

(1) Check if there are composite service $WSR_j$ meet $WSR_i(I_i, O_i)$ in the service log base. If so, and service composition satisfaction feedback degree is greater than or equal to calculation results, then return $WSR_j$ as the service composition. If not, go to step (2);

(2) Figure 4 is the demonstration of the result of running algorithm 2, the web service release picture.

(3) Check figure 3 to find if there are services where $WSR_i(I_i, O_i)$ meet $WSR_j(I_j, O_j)$, if so, return the description information of $WSR_j$, or else go to step (4);

(4) Check figure 3 to find all the service composition $WSC = (WSc_1, WSc_2, \ldots, WSc_n)$, where $WSR_i(I_i) \supseteq WSR_j(I_j)$ as well as $WSR_i(O_i) \supseteq WSR_j(O_j)$ or $WSR_i(I_i) \supseteq WSR_j(I_j)$ and register the Dijkstra algorithm (not described in detail here) and select the service composition $WSR(WSR_1, WSR_2, \ldots, WSR_n)$ with the shortest path from the service node $WSc_1$ to $WSc_n$, then go to steps (5); If there is not any service composition meet the conditions, return the information of request failed.

(5) According to formula 3, to calculate the satisfaction degree of service composition $WSR$, and register the descriptive information and calculated satisfaction degree of service composition. Eventually, return the service composition $WSC$.

5 Conclusions and Future Works

Web Service is considered as the new generation information exchange technology based on web after XML, with excellent interoperability and loosely coupled characteristics of the industry's concerns. Combined with service-oriented architecture, Web Services is gradually playing a more important role. This article is based on deeply study of SOA and Web Services, this paper focuses on the research of Web service composition, propose the service ontology concept model, and apply it to the web service composition technology. A service composition engine and main algorithm are also constructed and introduced. The results show that scientific and effective service composition can be achieved through this technique and dynamic service composition satisfaction degrees are greatly increased.

Illustrated and realized in this article, the service composition engine can adjust to the static as well as dynamic service composition needs. However, dynamic composition mechanisms of realizing complex services are not achieved. Therefore, the main work followed is to, in the domain of service ontology, construct adaptive inference algorithm, improve the capability of service inference, enhance the ability of service registration and semantic expression in service community (storage), and increase the satisfaction degree of complex dynamic composition.

References