Operative Semantic Web Services Oeuvre

N. Danapaquiame, E. Ilavarasan, Neeraj Kumar

Abstract

Web service composition is one of the challenging issues that have been studied over the past years. It consists of combining existing web services to best suit new user requirements. This paper suggests an UML profile to compose web services based on their behavioral aspects. The Web Service Description Language (WSDL) files are first transformed to Unified Modeling Language (UML) models; then the profile is used to integrate UML models and this will be transformed into Business Process Execution Language (BPEL) process using Model Driven Architecture (MDA) approach. Our approach consists of three main steps. In the first step, existing simple web services are discovered and situated in the web service registry. The developer significances the WSDL files of the candidate services and translates them into UML diagrams (class diagram, use case diagram and sequence diagrams). In the second step, the sequence diagrams are in the form of scenarios, are integrated into a single sequence diagram that describes the behavior of the composite web service, by using the interaction operators of UML 2.0. In the third step, the MDA approach is adopted to transmute the consequential sequence diagram into a BPEL process. Proposed method has the advantages of being self-governing of the web service composition language and the UML modeling (Visual Paradigm) tool.

Keywords: Web services composition, UML Profile, MDA, BPEL, ATL and SOA.

1. Introduction

Nowadays, disseminated applications are increasingly being developed in the context of Service Oriented Architecture (SOA), where the basic unit of computation is called a service. According to W3C [2], a Web service is defined as a software system designed to support interoperable machine-to-machine interaction over a network. The technology basically articulates around the following three components:

Simple Object Access Protocol (SOAP) provides the definition of an XML document, which can be used for exchanging structured and typed information between service peers in a decentralized distributed environment [3].

Web Services Description Language (WSDL) is an XML format for recounting network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to concrete network protocols and message formats to define an endpoint [4].

Universal Description, Discovery and Integration (UDDI) focuses on the definition of a set of services supporting the description and discovery of the web services available for clients, and the technical interfaces, which may be used to access those services [5].

Moreover, web services have functional, non-functional, behavioral, and semantic characteristics. The functionality of Web services is described using interfaces with input and output parameters. The quality of services like performance is described by the non-functional specification usually given as cost, response time, availability, security, reliability, and reputation. The behavior states, how to interact with the Web services, in

* Corresponding author. E-mail: n.danapaquiame@gmail.com
terms of sequences of input/output interactions, for
instance.

Web service semantics describe the meaning of the
services generally through the usage of ontology. The
description of Web services exposes the main aspects that
enable them to be published, found, and used by other Web
services. They are also the key elements in composing the
Web services into new ones [6].

The composition of Web services is one of the key
features advanced by the technology. It consists of
combining existing services to provide a richer
newcomposite service to meet some user requirements.

Such composition requires methods and languages for
basic Web services integration, such as: XML, XLANG,
BPML, WSFL, WSCI, BP4WS and WSCDL.

The composition techniques can be classified into two
categories, namely: static service composition and dynamic
service composition. Static composition allows the
requestor to create an abstract model that should be
respected during the execution of the composed web
service. While the dynamic composition enables selecting
the atomic web services automatically and combines them
to create an unlimited number of new web services.

2. Related Works

A constraint satisfaction based web service composition
algorithm that combines tabu search and simulated
annealing meta-heuristics.

In [5] semantic-based composition approaches have
brought some comprehensive advantages such as higher
precisions and recalls, they are far from the real practice
and hard to be applied in real-world applications due to the
several challenging issues such as performance issues of
time-consuming ontology reasoning, exponentially
expanded searching time in large service repositories, lack
of available and consensus ontologism, and higher using
thresholds for users who do not have much semantic
knowlVerge1. Traditional semantic-based composition
approaches have not yet well addressed several challenging
issues like performance issues, exponentially expanded
searching time, serious lack of available and consensus
ontologism, and higher using thresholds. An innovative
composition technique have been presented by combining
WorldNet with multiple heterogeneous ontologism together
to build ESQL for supporting automated service
composition. Based on ESQL, the computational
efficiency can be significantly improved while
guaranteeing the semantic composition quality.

More prominently, our approaches can be more scalable
to a large scale of service sets and alleviate users or
developers from the burden of using complicated semantic
knowlVerge1 for service composition, thus making service
composition more realistic and easier.

Cheng Zeng, Weijie Ou, Yi Zheng, Dong Han [1], as the
applications become more and more complicated, the
requirement of Web service composition will be more
urgent. The paper puts forward a new storage strategy for
web services which can be flexibly extended in relational
database. It also present a matching algorithm SMA
between web services of multiple input/output parameters,
which considers the semantic similarity of concepts in
parameters based on Word Net. Moreover, a service
composition algorithm Fast-EP based on the above storage
strategy is presented. Because utilizing the characteristic
and using index mechanism in relational database, an
highly efficient web service composition is obtained.

It extracts the feature vector of each web service or
composite service, and create dynamic linear hash index on
these vectors so that the results of each search could be
hierarchical classified. QOS and service price are utilized
to rank the result set.

The difference of service composition algorithm is
semantic similarity between multiple concepts based on the
characteristic of dynamic classifying of search results. QOS
and rank the result set. At last, a service search, service
composition ratio between our approach and research the
high efficiency results based on semantic.

In [2], Proposes a Petri net-based algebra, used to model
time flows, as a necessary constituent of reliable Web
service composition process. This algebra is expressive
each to capture the semantics of complex Web service
combinations. Many organizations are putting their core
business competencies on the Internet as a collection of
Web services. An important challenge is to integrate them
to create new value-added Web services in ways that could
never be foreseen forming what is known as Business-to-
Business (B2B) services. Therefore, there is a need for
modeling techniques and tools for reliable Web service
composition. Any service expressed using the algebra
constructs can be translated into a Petri net representation.
By means of a set of algebra properties, we are able to
transform and optimize Web service expressions
guaranteeing the same semantics of initial expressions. In
addition, the use of a formal model allows the verification
of properties and the detection of inconsistencies both
within and between services.

In [4] this paper introduces the modeling, mapping and
transformation of behavioral aspects of interacting Web
services, within the context of Model Driven Architecture
(MDA). There are certain systems, such as Web services,
where the dynamic aspects are of high importance and need
to be considered during the modeling and transformation
process, in order to create accurate representations in their
target domains. To demonstrate the approach, a realistic
example is presented, involving a number of Web services,
participating in a business process expressed as
choreography of exchanged messages.

Initially, the behavior of one of the participants was
modeled with a UML activity diagram. Then the UML
activity and WSCI met models are presented and a
mapping among their equivalent Meta modeling elements
is defined.

Afterwards, a set of transformation rules in OCL was
introduced to perform a transformation from the original
UML model to a WSCI compliant MOF model and code.
As a result, the whole process was successful and
highlighted the applicability of MDA in transforming Web services standards, representing business processes, message interchanges and workflow. In [3], present an approach to support and semi-automate this transformation process of composing services by transforming the models created by designers at a final executable workflow. This approach produces good quality applications in less time. The idea is to capture existing knowledge of the company and consider them as patterns parameterized reusable. These patterns are used for tool based model transformations of the business processes. To support their approach, the authors have used domain specific modeling languages (DSMLs) designed for each company to capture the business process models in different levels of abstraction. These authors have shown that a general modeling language such as activity diagrams or UML Business Process Modeling Notation (BPMN) does not model the business processes of companies using their vocabularies and terminologies as the DSML. In this approach MDA provides mechanisms for defining DSMLs and a conceptual framework for defining transformations between different DSMLs.

The survey gives us the different methods and approaches for web services composition. Each method gives us the better understanding of the web service composition and designing patterns. The survey also used to analyze different web service composition techniques for web services composition. Our method has several advantages. It is graphical, which allows the developer to import descriptions of exiting Web services (WSDL) and represent them clearly in class diagrams. Furthermore, our approach is based on UML but it is independent of the UML modeling tools and the WSC language used.

Another advantage is that our method gives a semi-automatic way to generate a BPEL process by adopting the MDA approach. These features are the keys that make our method simple and understandable by even non-experienced users.

3. Proposed Work

This paper proposes a new method for composing Web services by defining a “WSComposition profile” which is an extension of the UML2.2 sequence diagrams metamodel. The proposed method applies the MDA [8] to generate the code of the composite Web service in Business Process Execution Language (BPEL) [9]. Therefore, this profile allows us to represent the behavioral characteristics of Web services, and provides an easy way to design and compose Web services based on their behavioral aspect.

Basically, our approach consists of three main steps. In the first step, existing simple Web services are discovered and located in the Web services registry. The developer imports the WSDL files of the candidate services and translates them into UML.

3.1 Our new method for web services composition is based UML Profile

The key ideas of our method to compose Web services. In our approach, interactions with a services are modeled as scenarios, thus the composition of web services is nobody other than the composition of those scenarios. Knowing that the scenario describes the behavior of a system, our approach is based on the behavioral characteristics of web services to compose them. In our approaches proposals of two graphical ways:

The interaction between the user and any requested service can be clearly represented by sequence diagrams (scenarios) in UML. Accordingly, the services composition amounts composing these scenarios to meet the client needs. Thus, the idea is to model the WSC by web service composition profile which is an extension of the UML2.2 sequence diagram metamodel and to accept the MDA approach to automate the generation of BPEL process execution code corresponding to the resulting composite Web service. Our method is independent of the WSC language and the UML modeling tool. Thus, the user can choose the UML design tool, the language of composition and the execution engine to make has composition. In the current version, our method supports the BPEL language and the EclipseUML tool for UML design.

3.2 The Metamodel of UML2.2 Sequence Diagrams

Here, the sequence diagrams metamodel which constitute a type of the interaction package of UML2.2 metamodel. To define our profile use the Lifeline, Message and Combined Fragment classes as defined in OMG UML2.2. A lifeline is a Named Element that represents an individual participant in the Interaction. While Parts and Structural Features may have multiplicity greater than 1, Lifelines represent only one interacting entity. Message, however, is a Named Element that defines one specific kind of communication between lifelines of an interaction. The message specifies not only the kind of communication, but also the sender and the receiver.

Sender and receiver are normally two occurrence specifications (points at the ends of messages). Finally, combined fragment is an interaction fragment which defines a combination (expression) of interaction fragments. A combined fragment is defined by an interaction operator and corresponding interaction operands. Through the use of combined fragments the user will be able to describe a number of traces in a compact and concise manner.

3.3 System Architecture

Web services composition method for interaction between the user and any requested service can be clearly represented by the sequence diagrams in UML. Consequently, the services composition amounts composing the scenarios to meet the client needs. The web service composition and to adopt the MDA approach to
automata the generation of BPEL process execution code corresponding to the resulting composite web service as shown in the figure 1.

In our method is independent of the web service composition language and the UML modeling tool. Thus, the user can choose the UML design tool, the language of composition and the execution engine to make the composition.

In the current version, our method supports the BPEL composition language and the Eclipse tool for UML design. The various steps of the composition method in the following figure.

![Web service composition method](image)

Figure 1. Web service composition method

3.4. Algorithm

Algorithm 1: Testing of web services

For each CES and FCES (Faulty Complete Event Sequence) with UML SD (Sequence Diagram) DO

Generate data expanded CES/FEP (Completer Event Sequence /Faulty Event Pair) on the basis of SD

For each service of root control construct

{Invoke function Testpath generation recursively using current service component as a parameter store the result into TestPathList.}

For each CES DO

Execute CES

If final event is not reachable OR time constraint elapsed

   Remark test as failed

Else

   Remark test as passed

For each FCES DO

Execute FCES

If FCES leads to a Faulty Response

   Remark test as failed

Else

   Remark test as failed

Using SD, it generates the data for each CES and FCES. It executes the statements for each CES. If final event is not reachable or time period is completed, it shows that test has been failed. Similarly the statements are executed for each FCES, if response comes in terms of faulty, test is treated as pass otherwise failed.

Since vicissitudes to the composition graph, produced by new web services being added to the system, occur often, the incremental algorithm is also main. The services compositions are computed in advance and stored into the tourism table.

3.5. Implementation Result and Performance metrics

It consists of the manual selection of the candidate web services involved in this composition. These services are represented as UML diagrams (class diagram, use case diagram and sequence diagrams) based on their WSDL.
files.

The use case diagram in Figure.1 describes the interactions between the user and the various services.

During WSC profile and MDA approach, the following requirements have to be fulfilled: response time, throughput, reliability and cost.

Response time: the time required to complete a web service request.

Availability:

3.6. Computation of metrics provided by Glass Fish

1. Availability: is the probability that a service is up and running.
   Successful invocations / total invocations

2. Reliability: is the ratio of the number of error messages to total messages and can be calculated in the following way.
   \[
   \text{Reliability} = \left( \frac{\text{total fault}}{\text{total messages}} \right) \times 100
   \]

Table 1-formula for calculating performance metrics

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Response Time</th>
<th>Throughput</th>
<th>Reliability</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence (for existing)</td>
<td>$\sum_{i=1}^{n} R_T(S_i)$</td>
<td>$\frac{1}{1} \sum_{i=1}^{n} T(S_i)$</td>
<td>$\prod_{i=1}^{n} R(S_i)$</td>
<td>$\sum_{i=1}^{n} C(S_i)$</td>
</tr>
<tr>
<td>Parallel (for proposed)</td>
<td>$\max(R_T(S))$</td>
<td>$\min T(S)$</td>
<td>$\prod_{i=1}^{n} R(S_i)$</td>
<td>$\sum_{i=1}^{n} C(S_i)$</td>
</tr>
</tbody>
</table>

The calculation of performance metrics in accordance with the QOS parameters for both sequence and parallel services execution is shown in the table 1

Table 2-UML diagram metrics

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Response Time</th>
<th>Throughput</th>
<th>Reliability</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flight</td>
<td>40ms</td>
<td>10 req/s</td>
<td>95%</td>
<td>20</td>
</tr>
<tr>
<td>Train</td>
<td>40ms</td>
<td>10 req/s</td>
<td>95%</td>
<td>20</td>
</tr>
<tr>
<td>Hotel</td>
<td>40ms</td>
<td>10 req/s</td>
<td>98%</td>
<td>5</td>
</tr>
<tr>
<td>Car</td>
<td>40ms</td>
<td>10 req/s</td>
<td>98%</td>
<td>5</td>
</tr>
</tbody>
</table>

Using the table 1 the following parameters are calculated.
Response Time = Latency (LT) + Processing Time (PT)
RT = 10ms + 30ms = 40ms
Throughput = min (response time)
T(s) = min {40, 40, 10, 10} = 10 req/s
Cost(C) = cost of composed service
C = TicketReservation + TicketCancellation + Acceptance
C = 5 + 5 + 10 = 20. The values are shown in table 2.

The following table shows the comparison between the existing web services composition system and the proposed system. The difference in the response time and cost proves that the proposed system is more efficient than the existing system.

The response time for existing approach and the proposed approach is shown in the table 3.

<table>
<thead>
<tr>
<th>Scaled Value</th>
<th>UML Response time</th>
<th>RDB Response time</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>20</td>
<td>15</td>
<td>30</td>
</tr>
<tr>
<td>30</td>
<td>20</td>
<td>40</td>
</tr>
<tr>
<td>40</td>
<td>25</td>
<td>50</td>
</tr>
<tr>
<td>50</td>
<td>30</td>
<td>60</td>
</tr>
<tr>
<td>60</td>
<td>35</td>
<td>70</td>
</tr>
</tbody>
</table>

Figure 2: Plot for UML response time and RDB response time

The comparison between the UML response time and RDB response time is shown in figure 2. We conclude that the UML response time is better than RDB response time.

4. Conclusion

Web service is one of the emerging trend in internet. Composition of web service is the challenging research issues. In this paper, we are using MDA approach for representing the UML scenarios. Web service is created using net beans and integrated into sequence diagram using visual paradigm tool and has been proposed for evaluating the web services are fast and efficient in terms of cost effective

4.1 Future Enhancement

The possibility to extend our WSC profile to take into consideration the user needs in terms of QOS. If any one of the possibility of using semantic web services to automata web services finder and to offer a better precision on existing services. Our mapping method is unidirectional, to study the possibility to make it bidirectional in order to enhance the automation degree.

References


