A New Broker-Based Architecture for TQoS Driven Web Services Composition

Kavya Johny
Department of Computer Science,
Viswajyothi College of Engineering & Technology, Vazhakulam,
Kerala -686670
kavyajovita@gmail.com

Abstract—Web services are self-contained modular units that support synchronous and asynchronous message exchanges between business partners. There exist a huge number of web services that can handle particular requests. Web service composition consists of combining web services that supports business-to-business or enterprise application integration to offer more complex services. While performing web service composition, the selection of appropriate web service for each activity in the workflow from the discovered services that satisfy a given requirement has become an important problem. However, all the existing broker based architectures consider only QoS factors for web service composition. They do not consider the transactional constraints during the composition process. This paper proposes a broker-based framework for web service composition not only according to their QoS characteristics but also to their transactional properties and thereby facilitates dynamic integration of atomic web services.

Index Terms—web services, quality of service, transactional web service, broker based architectures, workflow, web service composition.

I. INTRODUCTION

Web services provide ready-to-use functionalities through fixed interfaces for other applications by hiding the implementation details and they are commonly used in real world applications. With the increasing number of available web services, maintaining these services and searching for the ones that satisfy a given requirement has become an important problem. In order to handle a complex request, a combination of more than one service is required. This process of combining web services to achieve complex tasks is called Web Service Composition (WSC). Generally, when a client application invokes operations in a composite WS [1], the composite WS executes these operations by serving as a broker that delegates the operations to some atomic WSs. The operation invocations between the application, the composite WS, and the atomic WSs are shown in Figure 1. The web services composition performs the service selection, that can be done based on quality factors and transactional properties. Previously, a framework for quality-driven web service composition [2] was proposed that selects the web services based upon the QoS requirements of the requestors. Different approaches for various optimal web service selection problems have been proposed in the previous years [3][4]. However, none of these approaches takes into account the transactional behavior of the composite WS. Bhiri et al. [5] extend workflows patterns in order to consider the transactional behavior in case of failures and recovery. A new TQoS selection algorithm is proposed [6]. This paper proposes a new broker-based architecture focuses on the selection of WSs based on their QoS and transactional behavior using the TQoS selection algorithm.

II. THE TQoS BROKER ARCHITECTURE

The new TQoS broker based architecture performs with an objective of selecting the best web service that satisfies transactional properties as well as the requester’s QoS constraints and preferences. The Broker [7] contains six main components as shown in Figure 2.

A. WF-Modeller

The First step is done by the WF-Modeller. The requestor inputs a service description to the Broker referring to the required final Web service [8]. The WF-Modeller returns a workflow as a result. It is a set of activities where each activity is complemented by a set of semantic annotations, to describe its functionalities and capabilities.

B. WS-Locator

Based upon the workflow that has been generated, the WS-Locator will identify one or more Web services for each workflow activity. It searches the UDDI registry to find similar services.

C. Transactional WS-Adapter

The Transactional WS-Adaptor selects the appropriate
services that satisfying the transactional property from the available services for each activity given by the WS-Locator. The required transactional property depends upon that of the immediate previous activity.

D. QoS WS-Adaptor

From the set of services for each workflow activity obtained from the Transactional WS-Adaptor it will select the best web services satisfying requestor’s end-to-end QoS constraints and preference for every task to take part in the composition.

E. QoS Manager

The QoS property values provided by service providers are finally verified by the QoS-Broker. The QoS-Manager will refine these values in QoS DB according to the user feedbacks to reflect more accurate values.

F. WSBPEL generator

This is the last step. The results returned by the QoS Adaptor are translated into a WSBPEL document.

III. QOS COMPOSITION

QoS issues are considered as an important factor in web service selection. The new broker architecture uses a global QoS optimization selection algorithm whereas the existing TQoS algorithm considers only local optimization. The Quality assessment of web service is used for obtaining high-quality results [9][10]. The Quality of service (QoS) is a combination of several qualities or properties of a service, specified in Table I.

<table>
<thead>
<tr>
<th>Table I</th>
<th>SOME POSSIBLE QWS METRICS THAT CONSIDER WHEN DISCOVERING RELEVANT SERVICES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>1</td>
<td>Response Time</td>
</tr>
<tr>
<td>2</td>
<td>Availability</td>
</tr>
<tr>
<td>3</td>
<td>Throughput</td>
</tr>
<tr>
<td>4</td>
<td>Likelihood of success</td>
</tr>
<tr>
<td>5</td>
<td>Reliability</td>
</tr>
<tr>
<td>6</td>
<td>Compliance</td>
</tr>
<tr>
<td>7</td>
<td>Latency</td>
</tr>
</tbody>
</table>

IV. TQOS SELECTION ALGORITHM

In the TQoS-driven selection algorithm, a workflow is given as the input and a TCWS is the output. When a WS is assigned to an activity of the workflow, its transactional property influences the selection of the WS for the next activity. It considers both atomic transactional Web Service (AT) i.e. AT = \{op1, op2, ..., opi, ..., opn\}, where opi denotes service operation which is the component service of ATS, and composite transactional web services where CT = \{twsi, tws2, ..., twsi, ..., twsn\}, where twsi denotes atomic transactional Web service, twsi ~ CT, i=1, 2, ..., n. The transactional properties of a CWS highly depend on the transactional properties of its component W.Ss such as pivot, compensatable and retrievable and on the structure of the workflow [11]. TQoS define two notions of execution risk in a transactional system: Risk 0: The system guarantees that if the execution is successful, the obtained results can be compensated by the user. The user can choose another application that can be used to undo the previous effect. Risk 1: The system does not guarantee that the result can be semantically undone by the user in case of successful execution. TQoS algorithm considers both the sequential pattern and parallel pattern [12] while assigning web services to the activities in the work flow.

A. Sequential Pattern Assignation

Proposition 1: In a sequential pattern, if the web service assigned to the first activity of the pattern is pivot (p), pivot retrievable (pr), atomic (a), or atomic retrievable (ar), then the WS assigned to the second activity should be pivot retrievable (pr), atomic retrievable (a), or compensatable retrievable (cr) in order to obtain a TCWS.

Proposition 2: In a sequential pattern, if the WS assigned to the first activity of the pattern is compensatable (c) or compensatable retrievable (cr), then WS of any transaction property can be assigned to the second activity to get the resulting CWS as always transactional (TCWS).

B. Parallel Pattern Assignation

Proposition 3: If a pivot (p) or an atomic (a) WS is
assigned to one activity of a parallel pattern, to obtain a TCWS, and then the WS assigned to the other activity should be compensatable retriable (cr). The transactional property of the resulting TCWS is atomic (≥).

**Proposition 4:** If a pivot retriable (pr) or an atomic retriable (≥) WS is assigned to one activity of a parallel pattern, to obtain a TCWS, the WS assigned to the other activity should be pivot retriable (pr), atomic retriable (), or compensatable retriable (cr). **Proposition 5:** If a compensatable (c) WS is assigned to one activity of a parallel pattern, to obtain a TCWS, the WS assigned to other activity should be compensatable (c) or compensatable Retriable (cr).

**Proposition 6:** If a WS is assigned to one activity of a parallel pattern is compensatable retriable (cr) then the resulting CWS is independent of the WS transactional property assigned to the other activity.

V. CONCLUSIONS

With the emerging role of web services in business processes, the requirement of composing and executing them has begun to draw high attention, and today the need to find the optimal web services composition for the business processes is a challenging issue. This paper proposes a new broker based architecture that performs an optimal web service composition according to the transactional properties and user QoS requirements which is an added advantage than existing brokers. This will dynamically generate a workflow based on user’s request and make a better web service selection for the composition. As the new broker based architecture makes use of the modified TQoS algorithm, it considers both sequential and parallel patterns and provides global optimization.

REFERENCES


