Orchestration as a Service (OaaS)

Samar Mohamed Taha Elkeiey, Ehab Ezzat Hassaneen, Osman Hegazy

Information Systems Department,
Cairo University, Faculty of computers and information,
Egypt

Abstract

A lot of business organizations need to use third party web services that lead to composing different web services to fit an organization business process model. This composition can be achieved through orchestration or Choreography. In this research, orchestration will be used as a mean of web service composition. Current framework does not guarantee the correctness of the orchestration process and will tailor the process and workflow each time according to developers' skills which may lead to faulty development. A new concept orchestrator as a service (OaaS) is proposed to avail orchestration in a form as a web service which works as a repository of all services instead of using each service alone, also the orchestrator will work as an engine to perform certain pre-defined workflows. The architecture of the OaaS is provided and used to achieve and solve the problems of web service orchestration and experiments show the correctness of the proposed solution and the effecting of using it in terms of a number of lines written and development time without any noticeable overhead.

Keywords: Web services orchestration, OaaS, Centralized orchestration, Decentralized orchestration, third party web services.

I. INTRODUCTION

Recently most of the medium and large sized business organizations develop and host only their main functions and outsource many other functionalities from the web services available on the Internet. However, the concern here is how to correctly choose the right combination of web services that will be used. If the user requirements need more than web service so the organization should integrate web services to meet the requirements [5], [3], [4], [24].

Web services that are published on the Internet will use protocols and interfaces that combine software and information [1]; when a number of web services are integrated together they are called a composite web service. This process may use Business Process Execution Language for Web Service (BPEL4WS) [7], [2], [9], [13], [14], [15] or Web Services Integration and Processing Language (WS IPL) [10], [16] to perform some business process that represents organization workflow [17], [11], [8], [6]. Subsequently, the composite services can be orchestrated either in a centralized or in a decentralized way [19] [20].

II. NEED FOR ORCHESTRATION:

A lot of web services are found on the Internet, but when using one alone it does not meet all the needs of some business processes or user requirements. This can be achieved by using the concept of web service orchestration. However, the composite services are considered as a single web service from the user point of view [10], [26], [27], [28] [29], [30], [31]. Nevertheless, there were some difficulties had been mentioned in [10] in using such composite services so; they stated that discovery is a prerequisite for selection, but selection is the main problem. Thus, services composition can be considered as a multi-objective optimization problem.

Web Services Composition Framework:

This framework is adapted by [18] shown in Fig.1 which shows the method of composition regardless of how this composition is done.

![Figure 1. A Survey of Automated Web Service Composition Methods][18]

III. ORCHESTRATION VERSUS CHOREOGRAPHY

The concept of composing web services in which one central web service is responsible for controlling all of the involved web services (those web services did not know that they are integrated together each of them just knows its role) is called orchestration[2], [9], [25], [32]. This central web service is called coordinator makes an orchestration in a centralized way as in the following Fig.2.
In the other side the concept of composing web services by integrating them together without needing for the central coordinator is called choreography [2], [9], [25], [32]. Choreography depends on making each web services in the composition know what to do exactly when to work and with who will exchange messages. This concept will be described in the following Fig.3.

IV. METHODS/APPROACH (CONTRIBUTION)

As web services orchestration is a very interesting area of research so it is useful to take much caring in which implementing it as a separate web service that orchestrates all the web services (on the same cloud) needed to perform some business logic [21], [22], [23] either in a centralized or decentralized way. This will solve the problem of using the third party web services as the user will use the orchestrator as a service (OaaS) as a repository of all services instead of using each service alone, Also the orchestrator will work as an engine to perform a certain pre-defined workflows which will save a lot of hard code as the users can easily use these workflows and easily update them by only changing in the orchestrator instead of changing the workflow and tailor it each time performing it, the following Fig.4 represents the architecture of OaaS:

V. RESULTS AND DISCUSSION

As a sample average of 10 projects each composed of 5 to 7 web services was taken to make the study on the effect of using OaaS on the number of written lines in these projects in the following three cases:

- The 1st case is that the user will use the OaaS to access all the web services in the projects either they were the third party or not.
- The 2nd case is that the user will use the OaaS to access half of the web services used in the projects and access the other web services by another way.
- The 3rd case is that the user will not use the OaaS at all and accessing all web services by another way.

The Fig. 5 represents the effect of using OaaS in the number of written lines to access these web services, the chart shows that the number of written lines increased when not using the OaaS that is because of writing the code many times with every workflow and tailoring it each time with respect to user needs and requirements.

By using the OaaS engine you will only need to write the code and perform the workflow once when the user designs his/her own composition.

Here the study will show the effect of using OaaS on development time, as shown in the previous chart 1 number
of written lines increased by not using OaaS so, definitely, the development time also will increase as in Fig. 6.

![Figure 6. The effect of using OaaS in development time](image)

From Fig. 6 it is obvious that the time decreased about 1/3 when using the OaaS, so, a lot of time and efforts will be saved in case of using the OaaS engine as we did not need anymore to tailor the workflow of theses web services.

The main shortage in the OaaS is that the execution time will increase in a linear manner with the increase of a number of web services in OaaS as shown in Table 1 and the Fig. 7 below starting with one web service the execution time spent by using the OaaS engine is 0.063 seconds and the time spent without using OaaS is 1.859 seconds.

<table>
<thead>
<tr>
<th>Number of web services</th>
<th>Execution time spent by using OaaS in seconds</th>
<th>Execution time spent without using OaaS in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.063</td>
<td>1.859</td>
</tr>
<tr>
<td>10</td>
<td>0.626</td>
<td>3.017</td>
</tr>
<tr>
<td>100</td>
<td>3.257</td>
<td>6.539</td>
</tr>
<tr>
<td>1000</td>
<td>24.834</td>
<td>22.297</td>
</tr>
<tr>
<td>10000</td>
<td>238.694</td>
<td>190.952</td>
</tr>
</tbody>
</table>

Increase number of web services to 10 web services will show that the execution time spent by using OaaS engine is 0.626 seconds and without using the OaaS will be 3.017.

Increase the number of web services by 10 times until reaching to 10000 web services resulting in the execution time spent by using OaaS engine is 238.694 seconds or 3.97 minutes and without using the OaaS engine will be 190.952 seconds or 3.1 minutes which means that the difference is about 1 minute in 10000 web services.

TABLE 1. The effect of using OaaS on execution time in seconds

As shown, by comparing the advantages and disadvantages of OaaS, the saving of development time and a number of written lines plus the efforts done in coding and tailoring the workflow each time; is important than the increase in execution time in a very large sample(10000 web services).

The difficulties of choosing the correct and suitable workflow do not exist in OaaS since it allows the user to choose his/her own workflow on the run time and can be flexibly changed for each user.

VI. CONCLUSION AND FUTURE WORK

The concept of Orchestration as a Service (OaaS) can be represented as a separate web service which leads to faster software development through integration of third party web services and lower cost of development since less code is written and maintained.

There is an overhead for using this centralized orchestration and the OaaS nevertheless the overhead is minimal and does not present noticeable decreasing of the performance.

Since this research relied on centralized orchestration, it would be preferable if decentralized orchestration was used to avoid all the drawbacks of centralized orchestration. This can be achieved by using proxy web services that represent a virtual container for each of the used web service and that will manage independently the message flow between web services which will be assigned to them by OaaS at deployment time.

Another important future addition is to manage web service availability at the time of accessing them as it may be down or busy.

Alternative plans should be used in such condition through using different web services or using retrial time out.

REFERENCES


Orchestration as a Service (OaaS)

[22] [Online].Available: https://www.priv.gc.ca/resource/fs-fs/02_05_d_51_cc_c_e.pdf