

How to cite this paper:

R Kanesaraj Ramasamy, Noor Ain Rosly, Illiana Azizan, & Fang-Fang Chua. (2017). Cloud-based web service composition using action script in Zulikha, J. & N. H. Zakaria (Eds.), Proceedings of the 6th International Conference of Computing & Informatics (pp 406-411). Sintok: School of Computing.

CLOUD-BASED WEB SERVICE COMPOSITION USING ACTION SCRIPT

R Kanesaraj Ramasamy¹, Noor Ain Rosly², Illiana Azizan³ and Fang-Fang Chua⁴

¹Linton University College, Malaysia, kanes@ktg.edu.my

²Pertama Institute of Technology, Malaysia, noorain@ktg.edu.my

³Pertama Institute of Technology, Malaysia, illiana@ktg.edu.my

⁴Multimedia University Cyberjaya, Malaysia, ffchua@mmu.edu.my

ABSTRACT. In this paper, we introduce the use of ActionScript 3.0 with web service composition for complex Flash Applications. ActionScript 3.0 is an object-oriented programming language which is used for arithmetic operation and presenting interactive output in Adobe Flash platform. Nevertheless, implementation of all the processes within the flash file itself consume too much of resources in the implementing device. In this paper, we have proposed to use web service composition in ActionScript 3.0 implementation to reduce the development time. Web service composition can provide a flow of system by invoking multiple web services. By using web service composition, we can produce a complex flash application without increasing the size of an independent flash file. Finally, by implementing web service composition in ActionScript 3.0, we can develop a complex web service application with smaller size and portable.

Keywords: web service composition, web service, ActionScript 3.0

INTRODUCTION

Web services are can be supported by many platforms and environment although it was established in one platform. Web services that are available in service repositories can be utilized in a single system and being invoked sequentially, which is called as web service composition. In other words, few web services can be composed in single application (Samwell, 2013). In fact, web service interfaces are similar with remote procedure call and the interaction protocols are manually written (Srivastava, 2003). Applications can be developed by invoking a couple of elementary or composite services using the composition technique and thus reducing development effort and time. During the composition activities, the sequence of events that will take place within any business process will be implemented by few web services in workflow management, where the logic is comprehended by composing autonomous applications. These composite services can also be in turn recursively composed with other services into higher level solutions (Lublinsky, 2007). Furthermore, this recursive composition feature allows the development of the new solution rapidly based on the existing business services and become one of the most important features of Service-oriented Architecture (SOA).

To create an interactive application, the common use of Adobe Flash is being very popular but later it was made even more interactive with the application of ActionScript 3.0. The Java

look like script is widely used to create responsive application such as buttons which react on button clicks, contents which can move based on the mouse movement and many more (Moock, 2003). However, one major drawback of this approach is that the repetition of code is required during the implementation of flash on different platforms for instance on web, mobile and desktop on the same system or process. Once there is an occurrence of change, all the .fla files are required to be updated. Significantly, the most serious disadvantage of this method is that the implementation of all the processes within the flash file itself consume too much of resources in the implementing device. Therefore, to address this problem, we have proposed a composition model for Adobe Flash that is flexible across different platform devices. Our proposed mechanism aims to reduce the implementation device resource usage for complex Adobe Flash application and to reduce the developer's effort when the changes are required. Based on the benefits provided by ActionScript 3.0, our proposed model can create complex application or even high processing application being presented in a highly responsive manner. This can be achieved by implementing the web service composition workflow in ActionScript 3.0.

The remainder of this paper is generally organized as follow: Related works includes the existing works on Web Service Composition and ActionScript 3.0. This is followed by the proposed model and its explanation on each of the proposed phase. Discussion on the implementation of the proposed model is included and finally the conclusion which presents the contribution and future works.

RELATED WORKS

Developers uses Adobe Flash as an authoring tool to develop Web-based animations, games and interactive materials. Usually, Flash is installed on Web-facing computers such as Java or Flash Player that runs as a client-side sandboxed virtual machine. There are number of benefits of Adobe Flash that include the ubiquity of its plugin; the complexity of the programs that can be implemented since the introduction of ActionScript 3.0. Moreover, Flash has been designed mainly for authoring interactive materials on the Web. However, the limitations of adobe flash include its absence from Apple touchscreen devices such as iPhones and iPads, and its own potential security issues (Reimers *et al*, 2015). A simple content manager and coding approach are well-known security steps in Flash applications (Ostroumov, 2016). Development process is oriented into people who do not want to spend a lot of time for coding. However, action-script is a popular choice for development due to low cost and ease to use (Pimentel, 2001). A Flash testing framework patent was developed indicating Flash program code is written in the ActionScript language, which supports object-oriented programming (OOP), exceptions, events, and has a native user interface component library. A Flash application is deployed from an application server as a SWF file that is referenced from web pages using <object> or <embed> tags that are interpreted by a browser plugin (Haugh *et al* 2015).

A modern web application uses web service to build complex application. There are rapid growth of applications which are built from services; that provide data and business logic through generic interfaces or APIs. In future, web-based applications can be developed merely based on services that provide either business logic, data, or user interfaces (Pietschmann, 2009). Pietschmann, (2009) proposed a method to use the web service as the user interface as there are already sufficient web services available to use. On the other hand, Ramasamy *et al* (2015a) implemented web service composition in cloud-based mobile application that is able to reduce the resource usage for mobile devices.

PROPOSED MODEL

Based on the related work, we have acknowledged the benefit of using Adobe Flash and how it can be further improved with the implementation of web services. We have proposed to use web service composition in ActionScript 3.0 implementation and the overall architecture is shown in Figure 1.

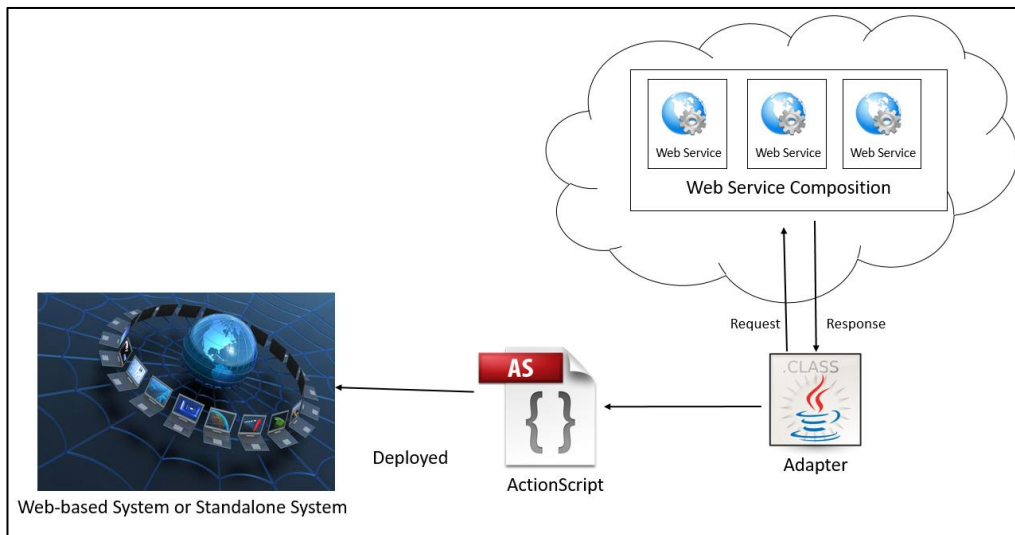


Figure 1: Overall Architecture for Cloud-based Web Service Composition Implementation in ActionScript 3.0

Based on Figure 1, we are proposing ActionScript 3.0 to be implemented using the Cloud-Based web service composition. Using this model, the design and the responsive user interface will be placed within the Flash file itself and the process will be designed using the web service composition workflow. This will reduce the size of the application and we can still run complex Flash application. Meanwhile, web service composition which is being placed in cloud environment is using cloud resources for processing. This allows the Flash application developers to build more complex application and run the application in any platform without taking into consideration of the installation platform resources. The adapter is required in between ActionScript 3.0 and Web Service Composition workflow as the ActionScript does not need to include the SOAP libraries. Moreover, the ActionScript 3.0 has also stopped supporting SOAP libraries. Web Service classes have been removed for ActionScript 3.0 projects which are being created in Flash. With the adapter, we can run the Flash application in new or old devices while using web services.

The adapter in Figure 1 uses the Algorithm 1, to get the inputs from the ActionScript 3.0; which are the user inputs and the adapter send the inputs to web service composition workflow to go through the process flow and to return the value. Using this implementation, we can have one java class with multiple methods to call different web services or web service composition workflows. For example, if we are developing a registration page which is expected to be highly responsive to the movement of mouse, the registration information can be sent to the adapter. Once the web service composition is invoked, the composition workflow can have web service to store the data in database and to send verification Short Messaging System (SMS) or email to the user. Once all the processes are completed beyond the ActionScript 3.0, the final output of Flash file will be smaller in size and performs faster.

```
Input: Input from ActionScript 3.0
Output: Result from Web Service Composition
Begin
Begin Method 1
    Receive Input from ActionScript 3.0
    var result = CallWebService(ReceivedInput)
    return result
End Method 1
End
```

Algorithm 1: Adapter to Call and Invoke Web Service Composition

IMPLEMENTATION AND DISCUSSION

For the implementation phase, we have used Adobe Flash Professional CS6 to develop the user interface as shown in Figure 2 and Visual Studio for development of web service composition workflow as proposed by (Ramasamy *et al*, 2015b). As for the adapter, we have developed using Eclipse as we publish it as a JAR file.

Based on the model proposed in Figure 1, we have developed a simple registration form for a Virtual Run as shown in Figure 2. The registration is done online. As we wanted to have an interactive system, so we have used flash to develop the form to be responsive meanwhile upon clicking the registration button we have invoked the web service composition workflow as shown in Figure 3. The workflow saves the user data to database and send email or SMS based on user choice. We have developed only the 'store data' web service meanwhile 'send SMS' and 'email' web services was taken from a public repository. We have reduced our development effort and time as we are reusing a web service published for public use.

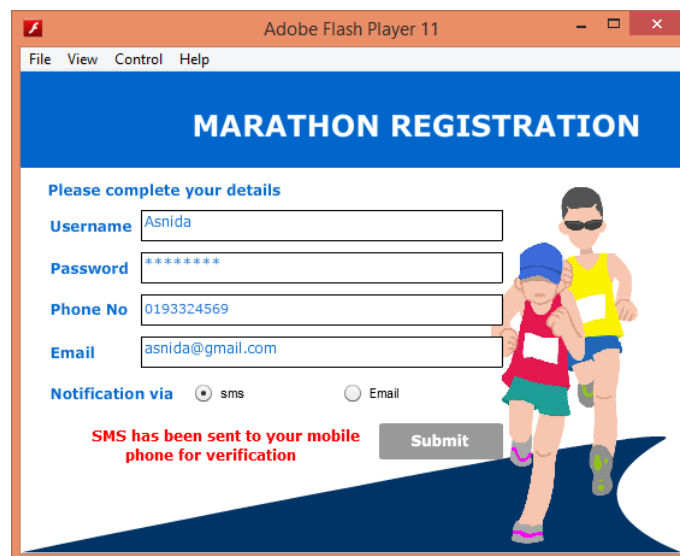


Figure 2: Registration System Development Using Adobe Flash

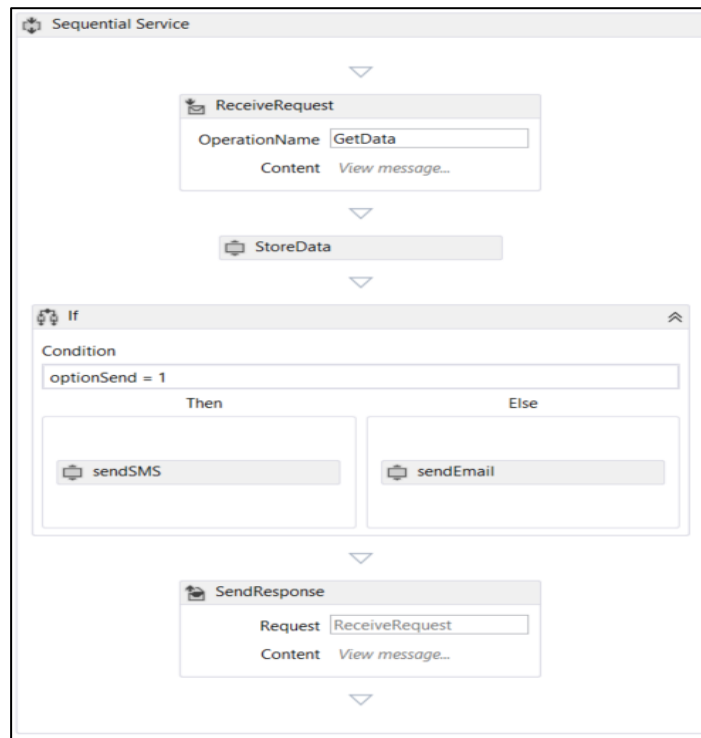


Figure 3: Web Service Composition Workflow

CONCLUSION AND FUTURE WORKS

In this paper, we discussed the implementation of cloud-based web service composition for ActionScript 3.0 implementation. The ultimate aim of this paper is to propose a model which allows implementation of complex processes for ActionScript 3.0 without having high dependency on its own library. Using the proposed solution, we can increase reusability of the codes especially in implementation across platform or various types of devices. Besides being highly reusable, we have also reduced the resource usage in implementation device and more complex implementation can be catered by cloud resources. As for future enhancement, we would like to test the reliability and efficiency of this model. Besides, since we have implemented a static web service composition workflow, thus we are keen to improve the methods to become dynamic and adaptive.

REFERENCES

- Da Graca Pimentel, M., Ishiguro, Y., Kerimbaev, B., Abowd, G. D., & Guzdial, M. (2001). Supporting educational activities through dynamic web interfaces. *Interacting with computers*, 13(3), 353-374.
- Haugh, J., Kennedy, R., Schemmel, M., & Group, I. V. (n.d.). Patent US9009674 - Flash testing framework. Retrieved October 08, 2016, from <https://www.google.com/patents/US9009674>
- LUBLINSKY, B. (2007). Service Composition. [Online] Available from: <http://www.infoq.com/articles/lublinsky-soa-composition> [Accessed: 25th October 2013]
- Mooch, C. (2003). *ActionScript for Flash MX: the definitive guide*. " O'Reilly Media, Inc."
- Ostroumov, I. (2016). *Elearning Development*. Proceedings of the National Aviation University, (1), 85-90.
- Ouyang, J., Lin, S., Jiang, S., Hou, Z., Wang, Y., & Wang, Y. (2014). SDF: software-defined flash for web-scale internet storage systems. *ACM SIGPLAN Notices*, 49(4), 471-484.

- Pietschmann, S., Voigt, M., Rümpel, A., & Meißner, K. (2009, June). Cruise: Composition of rich user interface services. In *International Conference on Web Engineering* (pp. 473-476). Springer Berlin Heidelberg.
- Ramasamy, R. K., Chua, F. F., & Haw, S. C. (2015a). Adaptive and Dynamic Service Composition for Cloud-Based Mobile Application. In *Advanced Computer and Communication Engineering Technology* (pp. 963-973). Springer International Publishing.
- Ramasamy, R. K., Chua, F. F., & Haw, S. C. (2015b). Web Service Composition Using Windows Workflow for Cloud-Based Mobile Application. In *Advanced Computer and Communication Engineering Technology* (pp. 975-985). Springer International Publishing.
- Reimers, S., & Stewart, N. (2015). Presentation and response timing accuracy in Adobe Flash and HTML5/JavaScript Web experiments. *Behavior research methods*, 47(2), 309-327.
- Sanwell, J. (2013) What is web service composition. [Online] Available from: <http://stackoverflow.com/questions/1244425/what-is-web-service-composition>[Accessed: 16th October 2013]
- Srivastava, B., & Koehler, J. (2003). Web Service Composition. *Current Solutions and Open Problems. ICAPS 2003* (1). p.23