Refactoring Flash Embedding Methods

Ming Ying, Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada

James Miller, Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada

ABSTRACT

Flash and Ajax are currently two popular Rich Internet Application (RIA) technologies, integrating Flash and Ajax will further enhance Internet users’ experiences. To communicate Flash written in ActionScript with Ajax written in JavaScript, the first step is to embed Flash content into a web page. Two methods can be used: markup-based Flash embedding methods and JavaScript-based Flash embedding methods. However, the drawbacks of markup-based Flash embedding methods make JavaScript-based Flash embedding methods a better solution. To automatically convert markup-based Flash embedding methods into a JavaScript-based method, this paper presents a refactoring tool, called FlashembedRT, to assist programmers with the transformation. This tool refactors the five different markup-based Flash embedding methods to the JavaScript-based Flash embedding method called flashembed.

Keywords: ActionScript, Asynchronous JavaScript and XML (Ajax), Flash, JavaScript, Refactoring, Rich Internet Application (RIA)

INTRODUCTION

With the advent of Rich Internet Application (RIA) technologies which are crucial to Web 2.0 sites, Internet users’ experiences have moved from the click-and-wait mode to a richer, faster and more interactive mode. Adobe Flash and Ajax (Asynchronous JavaScript and XML) are two popular RIA technologies. Adobe Flash is a multimedia platform for creating interactive and animated web sites (Ying & Miller, 2011). Flash games in Facebook¹ are examples of Adobe Flash applications. Ajax, a set of web development technologies, is an alternative to Flash. Ajax makes web-based applications feel like desktop applications by only changing updated information without refreshing the entire page² (Paulson, 2005).

Though both Flash and Ajax allow programmers to build dynamic websites, they have different focuses: Flash stresses on high-quality graphics and animations; it supports sound effects, video and audio playback and capture; whereas Ajax emphasizes actions on a website that involves the web server and the browser. Integrating Flash with Ajax to enhance the users’ experiences is highly popular. In practice, Flash is usually served as a partial substitution for the interface to provide many graphical tasks.

DOI: 10.4018/jssoe.2012070102
that cannot be accomplished by JavaScript. Currently, there are many innovative websites using Flash and Ajax technologies together, such as Google Finance and Yahoo Finance.

Figure 1 shows the combined Ajax and Flash working model. Flash content is integrated into the user interface as a component. The Ajax engine runs within the browser to communicate, interact and display information between the server and the browser. If more data is requested from the Flash component, the Ajax engine sends an asynchronous request to the server to retrieve extra data for the Flash component to display without causing the entire web page to be refreshed.

Figure 1. The combined ajax and flash working model

Flash is written in ActionScript; whereas Ajax is written in JavaScript. To implement the integration of Flash and Ajax technologies, interaction between ActionScript and JavaScript follows two steps:

- Embedding Flash content into a web page; and
- Communicating between actionscript and javascript.

Flash can be embedded through markup-based embedding methods or javascript-based embedding methods. Markup-based embedding methods use pure hypertext Markup Language (HTML) tags: <object> or <embed> to include Flash content, while javascript-based embedding methods utilize javascript to load Flash content. Flash embedding methods should meet the following criteria (Braunstein et al., 2007):

- **Cross-browser support**: the method should have the ability to support all web browsers;
- **Standards compliance**: the method should be in compliance with the World Wide Web Consortium (W3C) standards;
- **Support for alternative content**: alternative content should be supported. That is, the content should be accessible when no Flash Player is installed. This content can also be used for search engine indexing purposes;
- **Support for plug-in detection**: The version of the Flash Player should be detected before Flash content is displayed. Mismatches between the Flash content and the Flash Player may result in errors or broken content.

In general, markup-based Flash embedding methods provide no Flash content and plug-in version detection, whereas JavaScript-based Flash embedding methods meet all the criteria (the detail will be discussed in the Markup-Based Flash Embedding Method and JavaScript-Based Flash Embedding Methods Sections). However, most of the time, the average programmer simply applies the <object> and <embed> tags to embed Flash content (Schmitt, 2005). This is because: (1) many programmers have limited knowledge about the pros and cons of markup-based and JavaScript-based Flash embedding methods; (2) programmers are more familiar with the <object> and <embed> tags than the new JavaScript libraries. Although most of the JavaScript libraries for embedding
Related Content

Design-Based Research with AGILE Sprints to Produce MUVES in Vocational Education
www.irma-international.org/chapter/design-based-research-with-agile-sprints-to-produce-muves-in-vocational-education/188226/

Neural Network Control of a Laboratory Magnetic Levitator
www.irma-international.org/chapter/neural-network-control-of-a-laboratory-magnetic-levitator/79673/

Software Engineering at Full Scale: A Unique Curriculum
www.irma-international.org/chapter/software-engineering-full-scale/29603/

A Survey to Nature Inspired Soft Computing
Deepak Kumar, Sushil Kumar, Rohit Bansal and Parveen Singla (2017). International Journal of Information System Modeling and Design (pp. 112-133).
www.irma-international.org/article/a-survey-to-nature-inspired-soft-computing/199006/

Functional Method Engineering
www.irma-international.org/article/functional-method-engineering/75465/