

FTT: A System to Refactor Traditional Forms into Ajax-Enabled Forms

Ming Ying, University of Alberta, Canada

James Miller, University of Alberta, Canada

ABSTRACT

Forms are a common part of web applications. They provide a method for the user to interact with the web application. However, forms in traditional applications require entire web pages to be refreshed every time they are submitted. This model is inefficient and should be replaced with Ajax-enabled forms. Ajax is a set of web development technologies that enables web applications to behave more like desktop applications, thus allowing a richer, more interactive and more efficient model for interactions between the user and the web application. This paper presents a refactoring system called Form Transformation Tool (FTT) to assist web programmers refactor traditional forms into Ajax-enabled forms while ensuring that functionality before and after refactoring is preserved.

Keywords: Ajax, Efficiency, Form, jQuery, Refactoring

INTRODUCTION

Web 2.0 is responsible for the growth of Rich Internet Application (RIA) technologies. Rich Internet Application (RIA) (Driver, Valdes, & Phifer, 2005) technologies now dominate Web 2.0 applications to provide richer, faster and more interactive experiences by only changing updated information instead of reloading the entire web page every time when a user requests a change (Paulson, 2005). One of the most popular RIA technologies is Ajax (Asynchronous JavaScript and XML) which is a set of web development technologies that allows web

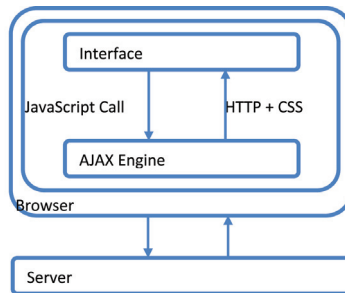
applications to be more interactive. It enables web applications to be updated without reloading the entire page; thus, improving the user experience (Zakas, McPeak, & Fawcett, 2007).

Ajax is comprised of five different technologies (Garrett, 2005):

1. XHTML and Cascading Style Sheets (CSS) as the presentation layer.
2. Document Object Model (DOM) trees being used for dynamic display and interaction.
3. XML for data interchange and manipulation.
4. XMLHttpRequest to asynchronously retrieve data.
5. JavaScript to bind everything together.

DOI: 10.4018/jssoe.2011100101

Figure 1. Ajax web application model



In traditional form based web applications, users on the browser (client) side trigger an HTTP request to the server side by either clicking on a link or a submit button. The server side processes this request and returns an entire HTML page back to the user. With the Ajax model (Figure 1), an Ajax engine runs within the browser to communicate with the server. This engine performs interactions and updates selected elements within the web page. The engine sends an asynchronous request to the server to retrieve new data if required. This asynchronous request does not interfere with the user's interaction with the application (Zakas, McPeak, & Fawcett, 2007). Some examples on Ajax technology applications include: Google Maps (<http://maps.google.ca/>), Google Suggest (<http://www.google.ca/>), Gmail², and Facebook (<http://www.facebook.com/>). Ajax applications are designed to significantly reduce waiting time. Card's research (Card, Robertson, & Mackinlay, 1991) argues that for a system to be seen as responsive, it has to respond in 0.1 seconds or less. The users will perceive a delay if the system takes one second to respond; and after ten seconds of waiting, they will abandon the task and move on. However, the Web still has many traditional forms, and transforming these forms into Ajax-enabled forms is not straightforward.

To aid with this transformation process, refactoring can be used. Refactoring is a technique to transform the source code's internal structure without changing the application's functionality (Fowler, 2002). A small behavior

preserving change to the source code is made with each refactoring. A significant structural change to the code can be seen once a sequence of these refactorings is applied. Clear, cleaner, simpler and more reusable codes are the main benefits of refactoring (Thompson & Reinke, 2002). A variety of refactorings and catalogs of various refactorings are widely available (Fowler, 2002; Thompson & Reinke, 2002; Kerievsky, 2004). Refactoring is very popular with programmers because the programming process can be divided into two steps with the technique: (1) write the code to meet the functional requirements; and (2) modify the written code to improve the quality of the code. Due to the tedious, error-prone and omission-prone features of manual refactoring, this paper extends the refactoring idea to allow programmers to refactor traditional web forms into Ajax-enabled forms using a semi-automated refactoring tool, Form Transformation Tool (FTT). The purpose of our refactoring is to improve the efficiency of web forms.

This paper makes the following contributions:

- (1) A method is introduced to refactor traditional forms into Ajax-enabled forms.
- (2) It produces a proof of concept tool named FTT to demonstrate that the transformation can be implemented using an automated approach. The aim of the automated approach is to save developers effort and time while guaranteeing a defect-free transformation.

18 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/article/ftt-system-refactor-traditional-forms/61313

Related Content

An Integrated Methodology for the Evaluation of Electronic Port Services

Nikolaos Marianos and Maria Lambrou (2010). *International Journal of Service Science, Management, Engineering, and Technology* (pp. 63-83).

www.irma-international.org/article/integrated-methodology-evaluation-electronic-port/45930

Why Controls Are Used in Platform Ecosystems?: An Analytic Hierarchy Process Approach to Rank Roles of Control Mechanisms in Platform Ecosystems

Sandip Mukhopadhyay, Shahrokh Nikou and Harry Bouwman (2016). *International Journal of E-Services and Mobile Applications* (pp. 1-19).

www.irma-international.org/article/why-controls-are-used-in-platform-ecosystems/160109

Service Economics

Adamantios Koumpis (2010). *Service Science for Socio-Economical and Information Systems Advancement: Holistic Methodologies* (pp. 198-217).

www.irma-international.org/chapter/service-economics/36095

Revisiting the Extended IoT Use Behavior Model Among Senior NCD Patients for Smart Healthcare in Malaysia

Chinnasamy Agamudai Nambi Malarvizhi, Shamima Raihan Manzoor and Rasheedul Haque (2024). *International Journal of Service Science, Management, Engineering, and Technology* (pp. 1-25).

www.irma-international.org/article/revisiting-the-extended-iot-use-behavior-model-among-senior-ncd-patients-for-smart-healthcare-in-malaysia/349911

Understanding Business Strategy

Emanuel Gomes and Paul Jackson (2012). *Service Science Research, Strategy and Innovation: Dynamic Knowledge Management Methods* (pp. 325-341).

www.irma-international.org/chapter/understanding-business-strategy/61883