

Integrating REST with RIA-Bus for Efficient Communication and Modularity in Rich Internet Applications

NR Dissanayake^{1#}, T Wirasingha² and GKA Dias²

¹University of Colombo School of Computing, Colombo 7, Sri Lanka

²Informatics Institute of Technology, Wellawatta, Sri Lanka

#nalakadmnr@gmail.com

Abstract— *Rich Internet Applications give a rich set of features and an enhanced user experience, but the engineering of Rich Internet Applications comes with complexities, mainly due to poor realization caused by the lack of architectural formalism. If we can realize the architectural properties of Rich Internet Applications well, it might reduce the engineering complexities and the development of Rich Internet Applications may too offer a good experience. In this paper we discuss enhancing the structure and modularity of our proposed concept RIA-Bus, by integrating the customized version of the REST style. We expect to evolve this hybrid style further to come up with an abstract architectural style for Rich Internet Applications.*

Keywords— *Rich Internet Applications, REST, RIA-Bus*

I. INTRODUCTION

Rich Internet Applications (RIAs) have become popular among users; and have gained the attention of the Web2 engineers over last decade (Lawton, 2008). The rich User Interfaces (UI) and the faster responding nature can be seen as main factors for the attraction for RIAs (Piero, et al., 2010). The asynchronous communication mechanism is the key, which enables the development of the rich features in RIAs (Busch & Koch, 2009). There are multiple approaches to implement the asynchronous communication and the script-based approach has become the main choice with its open-source, free and plug-in-less features (Farrell & Nezelek, 2007). The techniques and technologies used for script-based approach have been evolved from the Asynchronous JavaScript and Xml (AJAX) (Garrett, 2005) to Websockets (Fette, 2011) throughout the last decade. However – despite the popularity – the engineering of RIAs is still suffering from complexities (Piero, et al., 2010) (Li & Peng, 2012), hence the designing and developing the RIAs are still complicated tasks (Preciado, et al., 2007).

REpresentational State Transfer (REST) architectural style (Fielding, 2000) has become the de facto standard for the

Web2 applications, especially for designing and developing the web services. REST can be used to maintain the modularity of the applications – mainly for the pattern based on Create, Read, Update, Delete (CRUD) operations – applying its simple principle to the Unified Resource Locator (URL) using the GET, POST, PUT and DELETE form methods.

In our ongoing research we try to identify the architectural complexities in RIAs and design an architectural style for RIAs. We have introduced a concept – in our ongoing research – named RIA-Bus (Dissanayake, et al., 2015), to minimize the complexities engaged in the asynchronous communication in AJAX based RIAs. RIA-Bus helps to centralize the asynchronous request handling in the server, thus increases the simplicity and modifiability of the RIA.

In this paper we propose and discuss how the REST can be integrated in to the RIA-Bus, to increase the modularity of the server-side code. We expect that this integration will provide adequate support for various architectural and non-functional quality attributes, such as evolvability, extensibility and customizability.

II. METHODOLOGY

A literature survey was conducted to gain the domain knowledge of RIAs, web architectures, AJAX and other asynchronous communication techniques and technologies.

A cross-sectional survey was conducted to understand the current state of the facts identified in the literature survey. Targeted population was the individuals engaged in RIA development; the data were gathered using a structured questionnaire with closed end questions; and the gathered data were analysed using statistical methods, to derive knowledge.

Parallel to the surveys, we did run a series of experiments to realize the RIA development and gain some empirical