Best Practices for Rapid Application Development of AJAX based Rich Internet Applications

Nalaka R. Dissanayake[#1], G. K. A. Dias[#2]

[#]University of Colombo School of Computing, UCSC Building Complex, 35, Reid Avenue, Colombo 7, Sri Lanka.

[1]nalakadmnr@gmail.com, [2]gkad@ucsc.cmb.ac.lk

Abstract— **AJAX is a powerful script based approach for Rich Internet Application development, yet AJAX is suffering from various complexities which make the AJAX based Rich Internet Application engineering is difficult and Rapid Application Development is supported inadequately. If we can simplify the AJAX based Rich Internet Application engineering process by providing standards, the Rapid Application Development methodology can be used effectively to raise the productivity and the quality. In this paper, we propose a set of best practices to facilitate the project management of AJAX based Rich Internet Applications – hence utilize Rapid Application Development methodology – in the initial development and post-deployment maintenance, modification and expansion phases. These best practices are meant to increase the realization of the Rich Internet Application and the AJAX adoption – therefore reduce the complexities – by providing a standard structural arrangement for the development project. We plan to design a general architecture for AJAX based Rich Internet Applications and combine with the best practices discussed in this paper to formulate a new platform to support the Rapid Application Development methodology adequate, by increasing the realization, which causes to minimize the complexities.**

Keywords— **Rich Internet Applications, AJAX, Rapid Application Development, Best Practices.**

I. Introduction

Rich Internet Applications (RIAs) provide an interactive and responsive experience [1] to the user, with their rich Graphical User Interfaces (GUIs) and fast responses [2]; which are not available in classical page sequential web applications, with limited interactivity and slower in response [1]. RIAs resemble desktop applications, yet deliver the web based functionalities. With the introduction of HTML 5 and CSS 3, the RIAs have become more prominent with increased portability and availability [3]. Even though the demand for RIAs grow higher, there is a lack of full development methodologies and standards for RIA engineering [4]. There are distinct approaches to develop RIAs [5]; among them the script-based Asynchronous Javascript And Xml (AJAX) approach has become popular [6] over its competitor – the proprietary plugin based approach, which uses technologies like Adobe's Flash/Flex or Microsoft's Silverlight.

AJAX is a powerful technique [7] – which combines matured technologies like JavaScript (JS), http request-respond model, XML/HTML DOM and CSS – for developing RIAs [8]. It is capable of communicating with the server asynchronously and update the GUI by rendering the web page partially – without refreshing the whole page – which makes the AJAX is fast and responsive [9] [10]. Even the AJAX general architecture is not difficult to realize, the AJAX adoption within the RIA is suffering from various complexities [9] [11], hence AJAX based RIA engineering is considered difficult [6]. Solutions have being proposed by tools/frameworks venders and researchers – with their own pros and cons – to reduce these complexities and simplify the development. However, the AJAX technology is still immature [11]; and it lacks of standards [12] and architectural formalism [8].

As the customer demand for RIAs goes high [13], various Computer Aided Software Engineering (CASE) tools, frameworks and libraries for AJAX have being introduced and used by the RIA engineers, to get a rapid output with increased quality by incorporating the Rapid Application Development (RAD) methodology. RAD combines the CASE tools, best practices and some other factors into reliable formula for top-notched quality and productivity [14]. If the complexities can be reduced by increasing the realization of the RIA and AJAX adoption, we can expect a better support for RAD in AJAX based RIA development.

RIA designing embraces a completely different paradigm than tradition HTML web sites or web applications [15] and brings an added layer of complexity into the development. Properly employed set of best practices can help to make RIAs easier to develop [16]. The lifetime of software projects does not end with the deployment of the product after the development; it may lengthen with the requirements for maintenance, modifications and expansions; and the application of the RAD is continued to these post-deployment phases too. The assistance of best practices is necessary, when the RAD methodology is applied [17] in these post-deploy phases as well; instead of being limited to the initial development. Hence the identification and incorporation of set of best practices is significant, as same as other factors in RAD of AJAX based RIAs.

This paper discusses a set of best practices identified in our ongoing research – designing an architectural solution to address the difficulties in AJAX based RIAs, hence increase the support for RAD. In the next section we specify the methodology we used in our ongoing research, and in the discussion section we present the findings of the surveys briefly and discuss the set of best practices for AJAX based RIAs. The perspective of these best practices is to increase the assistance in RAD by supporting the management of the project, not only in the initial development life cycle but also in post-deployment phases. In the last section we conclude our discussion and specify the future work.

II. Methodology

978-1-4799-7732-1/14/$31.00 ©2014 IEEE

We conducted a literature survey, followed by a cross-sectional survey, parallel to a series of experiments.

The literature survey was conducted to gain the background knowledge related to the RIAs, AJAX and RAD; and to learn the interactions among these fields. Important ideas of the experts in the field of AJAX based RIA engineering, and the related researches, were examined during the literature survey.

Cross-sectional survey was targeted on the population of individual engineers (mainly designers and developers) of AJAX based RIAs. Data were gathered using random sampling technique, using a structured questionnaire with close end questions, and analyzed with statistical methods. The results derived from the cross sectional survey were useful to confirm the findings of the literature survey and to learn the up to date nature of the environment.

The series of experiments was helpful to experience the knowledge attained in the surveys, hence identify the areas to be improved to reduce the complexities in AJAX based RIA engineering. The experiments were prototype based and continued in incremental development. The facts discovered in early cycles were tested with different solutions to identify and extract the optimized techniques, to address the difficulties in AJAX based RIAs, in later cycles. To be specific in back-end development and limit the scope for evaluation purposes, we selected PHP as the server-side scripting language. HTML5 and CSS3 were used to develop the views, along with jQuery as the client-side JavaScript library.

Combining the knowledge of the surveys and the experience gained from the experiments, we derived a set of best practices, to provide an adequate support in the management of the AJAX RIA development projects therefore increase the support for RAD; in initial development, as well as in the maintenance, modifications and expansions.

III. Discussion

In the analysis of the literature survey, we identified several facts related to the difficulties and complexities engaged in RIA engineering and AJAX adoption. The major conclusion of the literature survey is that all the difficulties and complexities are related to the fact, that there is a lack of architectural formalism for AJAX enabled RIAs [8]. We have presented the analysis of the literature survey in our paper "An Analysis of Rapid Application Development of AJAX based Rich Internet Applications" [18].

We have discussed the analysis of the cross-sectional survey in our paper "What does the AJAX Rich Internet Applications need to Support the Rapid Application Development" [19]. The main finding of the cross-sectional survey is that the difficulty level of adopting AJAX features in a web page is increasing with the number of the features per page. The higher realization of AJAX general architecture or higher usage of CASE tools are not capable of reducing the development complexities.

In our ongoing research we propose that an architectural formalism and standards will help to increase the realization, hence reduce the complexities in RIA engineering. While continuing the experiments we have identified a set of architectural features to be expected from a standard architectural formalism for RIAs. We have presented these features in the paper "Essential features a general AJAX Rich Internet Application architecture should have in order to support Rapid Application Development" [20]. Throughout experimenting we noted some patterns, which can be utilized as best practices, when the proposed architectural features are used.

We propose that the following best practices will provide sufficient support for AJAX based RIA project management, hence RAD, in initial development and post-deploy maintenance, modifications and expansions.

A. *Architectural patterns*

Architecture is a vital part of software development and a proper architecture can be considered as a system framework [16]. And architecture plays an important role in RAD [21], while it actually brings down the cost and reduce development risks [16]. This general software engineering rule is valid for RIAs too, hence its crucial to select appropriate architectural patterns – like 2 tier/3 tier/multi-tier, Model-View-Control (MVC), Service-Oriented Architecture (SOA), etc. – for the project to provide a solid architectural foundation. Since we have identified the AJAX adoption complexities are directly related to lack of architectural formalism in AJAX base RIAs [8], we propose that engineers must identify suitable architectural patterns for the RIA, before jump directly into the design or development. We recommend the mandatory use of MVC for AJAX based RIAs.

With the increased volume of JS and CSS in modern RIAs, maintaining the code base has become significantly difficult [15]; therefore a standard directory and file management mechanism is essential. We propose best practices to facilitate the directory and file management, by providing a directory/file structure and naming standards which may increase the realization of the both RIA and project structure; hence assist the project administration and maintenance.

B. *Directory Structure*

Unlike in desktop applications, the file path is important in RIAs. Since lots of files in an RIA are connected via navigational hyperlinks, external JavaScript/CSS file linkage elements, server-side include/import statements and HTML/CSS image linkages; it is essential to have a simple and firm directory structure. It will increase the realization of the file organization, therefore minimize the confusion of the linkage and connectivity between the files. Efficient file organization in an effective directory structure will help to reduce the unnecessary computations of the file paths, based on the current location too.

We propose that all the web pages – referred as the Views – including HTML and PHP pages, should be saved directly in the root directory. The CSS, JavaScript and image files used for the presentation of the Views, should be contained in a directory named "Views" in the root directory. Images can be organized in sub directories as needed, inside the "Images" directory contained in the "Views" directory.

The JavaScript files which contain the event handlers, used to implement the behaviors, and AJAX engine for asynchronous communication, should be contained in a directory named "Controllers" inside the root directory.

The PHP files (not web pages) which include the pure business models, should be saved in a directory named "Models" inside the root directory. This directory structure gives a hint of the MVC pattern and provide a clear separation of the MVC modules as well.

To avoid the complexities in file linkages, we recommend not to use any sub directories in the Views, Controllers or Models directories – other than for the Images or any third party frameworks/libraries which have their own directory structures. If there are multiple modules in the RIA and a separation of the files between these modules is needed, we advise to use the File Naming practice specified below.

C. File Naming

Always use meaningful names for the files; class name is preferred, where applicable. We recommend to use Camel-case or Pascal-case when naming the files, to avoid the unnecessary spaces or/and length in the file names, yet maintain the readability. When a clear separation between the files in separate modules is needed, use the module name as a prefix to the file name, and use "grouping files by name" feature in the development system for easy navigation.

Refer the figure 1 for the directory structure and the file naming practices.

D. Object Oriented Programming

Use Object Oriented Programming (OOP) concepts as much as possible. For server-side development, always use OOP class definitions and implement each class in a separate file. We advise to use the class name as the file name – as mentioned in the previous practice – along with the module name prefix, if needed.

Even though the client-site JavaScript has no class definitions, consider the file itself as a class. In a single file, implement only the properties and actions belong to the specific class.

E. RIA Bus

An asynchronous call from the AJAX engine in the client-side is pointed to a server-side script file, and for each and every AJAX request, there supposed to be a separate dedicated server-side script file to handle the request. When the number of AJAX calls increases, the number of server-side files – dedicated to handle the AJAX requests – also increases. The increasing number of files will create an environment which the project file structure become complex and can introduce new layer of difficulty in the project management.

Usually these server-side script files who handle the AJAX requests are small, contains structured scripts, and not aligned to the OOP concepts. When there are many files which does not follow the OOP concepts, it may tend to violate the best practices we discussed above too.

To manage this situation we propose to use a technique called "RIA Bus", which contains the code to handle all the AJAX requests in a single file. All the AJAX requests should be pointed to this RIA Bus file and the requests can be distinguished using an argument passed into the RIA Bus from the AJAX engine. This RIA Bus file will not contain any code for business models, instead it will use the classes developed in the Model files. If the RIA is large and complex we recommend to implement RIA Bus per module and use the naming practice we propose to save them. Since this RIA Bus use the classes in the Model, the RIA Bus file should be saved inside the Model directory. Figure 2 illustrates the architecture of the RIA bus practice.

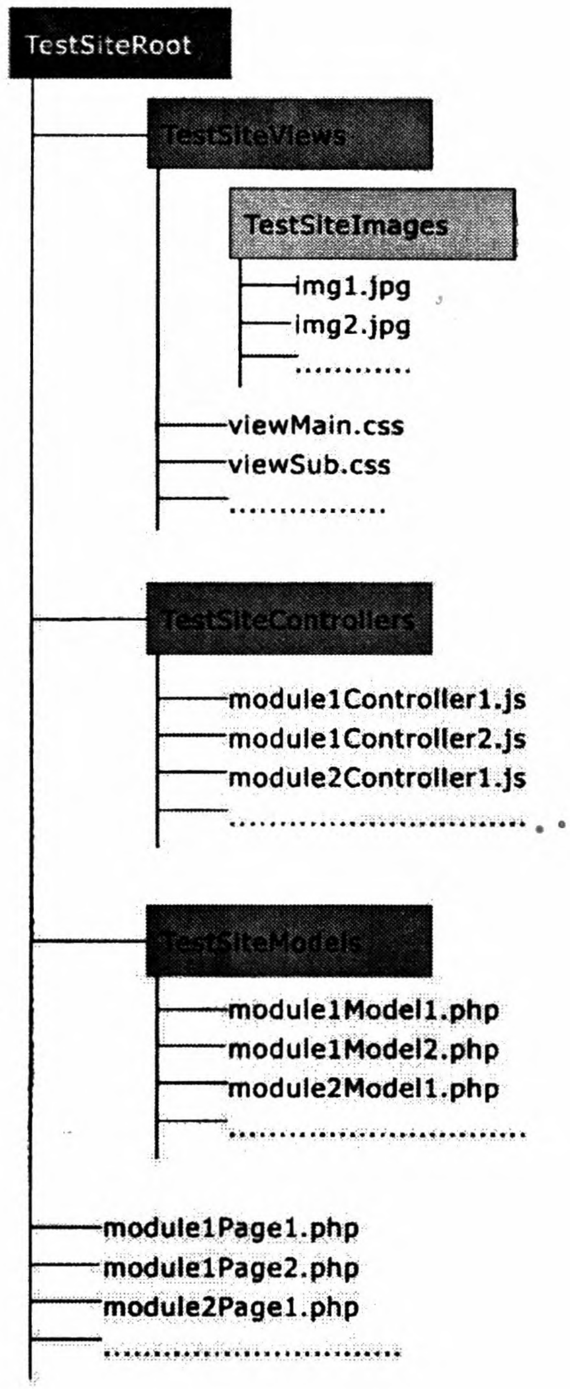

Fig. 1 Directory structure and File naming

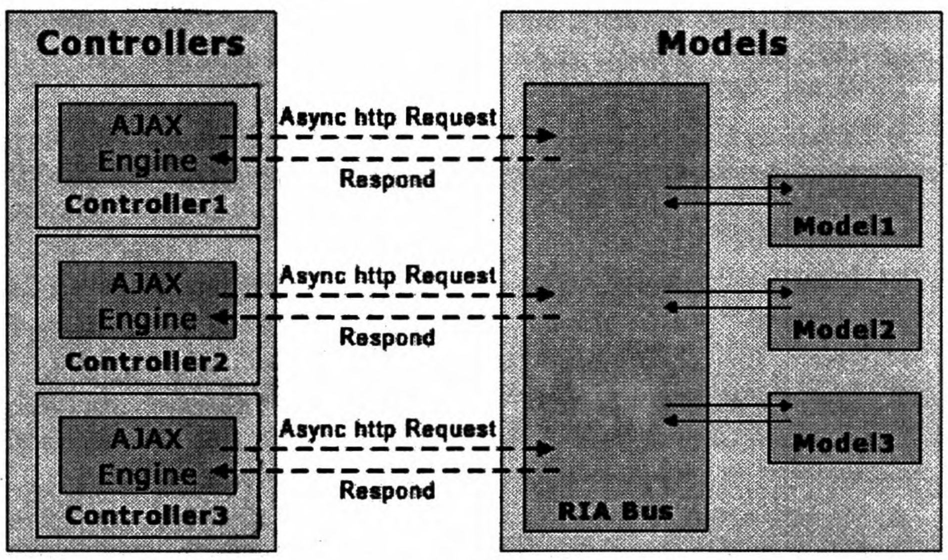

Fig. 2 RIA Bus

F. Processed deployment

To improve the security and the performance, it is always advisable to process the JavaScript/HTML/CSS files before deploying the RIA. Some techniques like consolidation, minification, and compressions are used to process JavaScript and CSS files to enhance the performance of the RIA. And to improve the security, it is wise to obfuscate JavaScript files before deploy. As a practice, we suggest not to process the original files, but a copy of the complete RIA. Always keep the original files safe for post-deploy maintenance, modifications and expansions. Prepare a processed version of the RIA and use it for the deployment.

G. Standard best practices

Other than our specified best practices, we recommend to use general software engineering practices and RAD best practices in RIA engineering to support RAD adequate. Some good set of conventions have being introduced for JavaScript by Douglas Crockford [22]; and Austin Govella also has presented a set of best practices for JavaScript in RIAs [17]. We highlight that it is important to select the suitable generic best practices for the RIA, in the beginning of the project.

IV. Conclusion

AJAX, the powerful and popular script based RIA development approach is suffering from complexities, hence the RAD is not supported adequate. We propose a set of best practices to favor the RIA engineering project management, to facilitate the RAD not only in Initial development, but also in post-deployment maintenance, modifications and expansions needs.

Our ongoing research is in progress on designing a general architecture for AJAX based RIAs, to increase the realization of AJAX adoption. Combining the general architecture with the best practices we discussed in this paper, we expect to formulate a platform to increase the support for RAD of AJAX based RIAs engineering.

References

[1] G. Lawton, "New Ways to Build Rich Internet Applications," *Computer*, vol. 41, no. 8, pp. 10 - 12, August 2008.

[2] O. Cameron, "A Look at Rich Internet Applications," *ORACLE MAGAZINE*, July/August 2004.

[3] E. J. Correia, "What's Next for HTML5?," Intel Software Adrenaline, 2013.

[4] J. Preciado, M. Linaje, F. Sanchez and S. Comai, "Necessity of methodologie to model Rich Internet Applications," in *Proceedings of the 2005 Seventh IEEE International Symposium on Web Evolution*, 2005.

[5] J. Farrell and G. S. Nezlek, "Rich Internet Applications The Next Stage of Application Development," in *Proceedings of the ITI 2007 29th Int. Conf. on Information Technology Interfaces*, Cavtat, Croatia, 2007.

[6] J. Li and C. Peng, "jQuery-based Ajax General Interactive Architecture," in *Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference*, Beijing, 2012.

[7] Z. LIN, j. WU, Q. ZHANG and H. ZHOU, "Research on Web Applications Using Ajax New Technologies," in *International Conference on MultiMedia and information Technology*, 2008.

[8] A. Mesbah and A. v. Deursen, "An Architectural Style for AJAX," in *Software Architecture, 2007. WICSA '07. The Working IEEE/IFIP Conference*, Mumbai, 2007.

[9] S. Salva and P. Laurencot, "Automatic Ajax application testing," in *Fourth International Conference on Internet and Web Application and Services*, 2009.

[10] B. Kao-ming and C. Geng-guo, "AJAX-based applicable framework research and design," in *Information Science and Engineering (ICISE), 2010 2nd International Conference*, Hangzhou, China, 2010.

[11] L. D. Paulson, "Building rich web applications with Ajax," *Computer*, vol. 38, no. 10, pp. 14 - 17, Oct 2005.

[12] M. Held and W. Blochinger, "Collaborative BPEL Design with a Rich Internet Application," in *Eighth IEEE International Symposium on Cluster Computing and the Grid*, 2008.

[13] K. Kronauer, "Best Practices for Building Rich Internet Applications," [Online]. Available: https://techcommunity.softwareag.com/pwiki/-/wiki/Main/Best+Practices+for+Building+Rich+Internet+Applications/pop_up;jsessionid=C3D842B94E0523BCDA7BBFC4C272AC41?_36_viewMode=print. [Accessed 28 April 2014].

[14] C. Inc, "What is Rapid Application Development?," 2002.

[15] D. Jennings and S. Wyman, *Rich Internet Application UI Development Best Practices*, TendemSeven, 2008.

[16] A. Solutions, *The Importaance of Software Architecture*, Architech Solutions.

[17] A. Govella, "Best Practices: Implementing Javascript For Rich Internet Applications," 27 July 2005. [Online]. Available: http://thinkingandmaking.com/thoughts/73/best-practices-implementing-javascript-for-rich-internet-applications. [Accessed 5 May 2014].

[18] N. R. Dissanayake, K. A. Dias and M. Jayawardena, "An Analysis of Rapid Application Development of AJAX based Rich Internet Applications," in *International Conference on Advances in ICT for Emergin Regions (ICTer)*, Colombo, Sri Lanka, 2013.

[19] N. R. Dissanayake and K. Dias, "What does the AJAX Rich Internet Applications need to support the Rapid Application Development," in *3rd International Conference on Human Computing, Education and Information Management System (ICHCEIMS 2014)*, Sydney, Australia, 2014.

[20] N. R. Dissanayake and G. K. A. Dias, "Essential Features a General AJAX Rich Internet Application Architecture Should Have in Order to Support Rapid Application Development," *International Journal of Future Computer and Communication*, vol. 3, no. 5, pp. 350-353, 2014.

[21] D. Hough, "Rapid Delivery: An eveolutionary approach for application development," *IBM SYSTEM JOURNAL*, vol. 32, no. 3, pp. 397-419, 1993.

[22] D. Crockford, "Code Conventions for the JavaScript Programming Language," [Online]. Available: http://javascript.crockford.com/code.html. [Accessed 04 05 2014].