The Significance of Importance of an Architectural Pattern for AJAX Based Rich Internet Applications

Nalaka R Dissanayake[1], GKA Dias[2]

[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 popular and powerful script-based technique for developing Rich Internet Applications in the era of WEB2. Despite its strength, AJAX is suffering from difficulties because of various complexities. If we can identify the root cause of these reasons and address the complexities, we can minimize the difficulties in the AJAX based Rich Internet Applications engineering and produce quality products faster by incorporating the Rapid Application Development methodology. In our ongoing research we try to identify the complexity factors of AJAX based Rich Internet Application engineering and provide solution(s) to formalise a standard platform. We have understood the importance of having a standard architectural formalism for AJAX based Rich Internet Applications and in this paper we discuss about the significance of having a specialized architectural pattern for AJAX based Rich Internet Applications, which we think is essential in addressing the complexities in Rich Internet Application engineering and support the Rapid Application Development adequate.*

Keywords— Rich Internet Applications, AJAX, Architectural Pattern

I. INTRODUCTION

WEB2 is the era of Rich Internet Applications (RIAs), which can be considered as hybrid of both web and desktop applications (Piero, et al., 2010). RIAs have rich User Interfaces (UIs) and they are interactive and responsive than classical page sequential web applications. Users now a days experience the richness of the RIAs via popular web applications like Facebook, Google apps, Flicker, etc… and they expect to feel the same in other web application they consume too. This makes a higher demand for RIAs, hence for the RIA development technologies (Lawton, 2008).

There are three main approaches for RIA development. First approach uses the proprietary plugin based technologies like Adobe's Flash/Flex, Microsoft Silverlight and Java applets. Second approach uses the popular script-based Asynchronous Javascript And Xml (AJAX) techniques. The third and the least known approach is the browser based techniques like Mozilla's XUL (Farrell & Nezlek, 2007). AJAX is not just a trend, it's a powerful way of developing the rich features in RIAs (LIN, et al., 2008). The combination of well-known and established technologies like JavaScript, XML, HTML, CSS and http request-respond model makes the AJAX unique and strong (Mesbah & Deursen, 2007).

Since AJAX has become a popular and powerful approach in RIA engineering, various Computer Aided Software Engineering (CASE) tools including design tools, testing and debugging tools, and Frameworks/libraries have been introduced for AJAX based RIA engineering. However still the AJAX is suffering from complexities (Paulson, 2005) due to it's complicated nature – which makes the realization of the AJAX based RIAs is difficult (Li & Peng, 2012) – therefore the RIAs still needs specialized development methodologies (Preciado, et al., 2005).

Architecture is the foundation of any software system and the support gained from a carefully designed solid architecture is significant, throughout all the phases of the software engineering projects (Solutions, 2014). Architecture provides an abstract overall picture of the system and assists in realizing the system, hence helpful in reducing the complexities. Architectural patterns like 2-tier, 3-tier, n-tier, Model-View-Controller (MVC) and Service Oriented Architecture (SOA) offer a good framework in designing the system architecture; and the architectural patterns help in capturing the knowledge of successful solutions (Selfa, et al., 2006).

In our ongoing research, we are trying to identify the reasons for the complexities and difficulties engaged in AJAX based RIA engineering, and provide architectural solution(s) to overcome them. In this paper we focus on specifying the significance of importance of having a general architectural pattern for AJAX based RIAs to support the RAD adequate. In section II we specify the methodology we used in our ongoing research, and in the section III we analyse and discuss the findings. In the section IV, which is the last, we conclude our findings.

II. METHODOLOGY

The problem identification started with an experimental development of RIA using AJAX, using Apache server, and PHP as server-side language. We wanted to incorporate architectural patterns and maintain the Object Oriented Programming (OOP) practices highly, and we wanted to do the development rapidly too. As we continued, we experienced some difficulties when incorporating MVC pattern and OOP concepts into the RIA development. Then we initiated a literature survey to gain a better understanding and knowledge about the RIAs, AJAX, Architectural patterns and application of Rapid Application Development (RAD) methodology in RIA engineering.

As we learn the conceptual background – throughout the literature survey – and identify the difficulty factors of the AJAX based RIA development, we conducted a cross-sectional survey, to verify the knowledge gained from the literature survey, and to confirm the facts are up to date. Our targeted population was the individuals – designers and developers – engaged in AJAX based RIA engineering and we used random sampling technique for the selection. A structured questionnaire with closed-end questions was used to gather data, and the data were analysed using statistical methods.

Parallel to the surveys, we continued our prototype based experimental development, in an iterative and incremental process, to experiment further and experience the findings of the surveys. For evaluation purposes we limited the development to a platform with HTML, CSS, Javascript/jQuery and PHP; using MySQL database server and Apache web server. In each and every iteration we tested various solutions and techniques on different difficulty facts, and the derived knowledge was used in the later iterations. As we progress, we noticed that the results could be used to originate an architectural standards(s). In the section III, we discuss the results of both surveys and the series of experiments, in detail, to express the importance and the significance of an architectural pattern, in reducing the complexities of AJAX based RIA engineering.

III. DISCUSSION

In the analysis of the literature survey, we came across various ideas and perspectives of the experts and researchers, about the complexities of AJAX based RIA designing and development. We have analysed and discussed them further in our paper "An analysis of Rapid Application Development of AJAX based Rich Internet Applications" (Dissanayake, et al., 2013). The main conclusion of the literature survey analysis was, that all the complexities associated in AJAX based RIAs are related to the fact, that there is a lack of architectural formalism and standards for AJAX based RIAs (Mesbah & Deursen, 2007). It makes the realization of the RIA and AJAX adoption is complex and difficult. A poor realization may directly affects the designing phase hence the development phase too.

In the cross-sectional survey we tried to verify the outcomes of the literature survey, while focusing on the CASE tools usage and the help gained from the CASE tools in RIA engineering. We worked on verifying the significance of the support from the CASE tools in addressing the difficulties in AJAX based RIA development, and some other RAD related facts, through the cross-sectional survey. We have summarized the analysis of the cross-sectional survey data in our paper "What does the AJAX Rich Internet Applications Need to Support the Rapid Application Development" (Dissanayake & Dias, 2014).

The cross-sectional survey analysis shows that the higher usage of the CASE tools or good understanding of the AJAX general architecture have not helped to reduce the difficulty of the AJAX RIAs development. We noted in the analysis that the difficulty level of implementing AJAX features in the same page, increases with the number of AJAX features in the same page. Figure 1 shows the variation of the difficult level factor – which is calculated as the weighted average, which varies within the range 5 (lowest difficulty) to 41 (highest difficulty) – over the number of AJAX features implement in the same page.

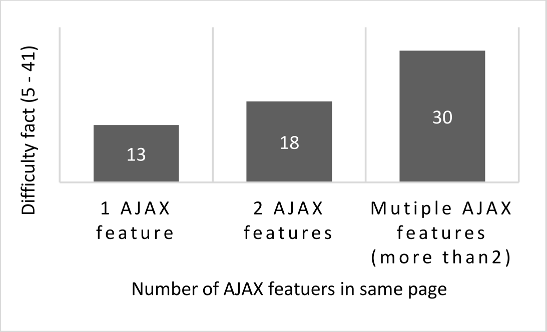

Figure 1. Difficulty factor of AJAX feature(s) development per page

The results of the cross-sectional survey, support the confirmation of the findings of the literature survey, so we resolved that the literature survey findings are up to date and valid. By looking at the analysis of the surveys, we highlighted that a good understating of AJAX general architecture or adequate usage of CASE tools are not providing enough strength to overcome the complexities in AJAX based RIA engineering. Furthermore we conclude that the complexities, which generate the difficulties in AJAX based RIA engineering are originated by the poor realization caused by the lack of architectural formalism.

As we continue on experimenting with the knowledge gained from the surveys, we focused on deriving features to be implemented as rules/standards for designing architectures for AJAX based RIAs. We expected to incorporate features – which we think essential in RIA designing and development – like MVC pattern, OOP practices, and needed to derive straight forward techniques for adopting AJAX features within the RIA. We have discussed in detail, the features we expect from a general RIA architecture, in the paper “Essential features a AJAX Rich Internet Application architecture should have in order to support Rapid Application Development” (Dissanayake & Dias, 2014). While experimenting, we were able to identify and test some good techniques, to address the difficulties in incorporating AJAX features within the RIA. And these techniques support incorporating the features we have proposed to be expected from an RIA architecture and helping in formalizing an architectural structure/framework.

The major focus of our ongoing research is to introduce an architectural solution for the complexities in AJAX based RIA engineering, and to increase the support in RAD methodology. As we have noted in the introduction, the architecture provides an abstract overall knowledge about system and it acts like a foundation, where the rest of the system is built upon. We believe that obviously a general architecture will offer the realization to address the complexities hence to overcome the difficulties in AJAX based RIA engineering; if the solution provides all the features we expect from it.

But when we consider the term “Rapid Application Development”, the decision of implementing an architectural solution raised a new question. In general practice or engineering processes, the architectural designing and system designing process is considered significant, and the system is designed and documented before the implementation. But commonly in Agile based RAD processes, the design is limited to some sketches, or the design is never done; mostly the design decisions are left to be handled by the programmers (Sommerville, 2011). In such environment, will standards for architectural designing of AJAX based RIAs be helpful?

Architectures always increase the realization of the system, and the architectural patterns can be considered as abstract description of tried and tested good practice of the generic forms of the architectures; and we can expect that the architectural patterns will be success in similar systems too (Sommerville, 2011). Hence the usage of architectural patterns can be considered important, to provide significant benefits in RAD enabled projects. And the knowledge of a specific architectural pattern could help to start the development, bypassing the design phase. This general knowledge is applicable in RIA engineering either, and a good specific architectural pattern could be useful in RIAs engineering, in a RAD enabled environment. For AJAX based RIAs, the general AJAX architecture provides a good knowledge of the mechanism of an AJAX feature. But it does not provide the abstraction of the complete RIA and the AJAX adoption within the RIA. Therefore the importance of the availability of a specific architectural pattern for AJAX based RIAs can be considered significant.

With that knowledge we reached a turning point, to look at the problem in the perspective of the RAD and significance of the usage of architectural patterns in RAD environment. Since we noted that the techniques and the features we derived from the experiment results exhibit some patterns, we started to work on utilizing the features expected from RIA architecture and techniques we have learnt, to formalize not just an architectural framework but an architectural pattern.

Our intension of introducing the architectural formalism is mainly to reduce the complexities by increasing the realization of the RIAs and the AJAX adoption. An architectural pattern will do the same and not only that, it will provide a general architectural framework for the AJAX based RIAs too. We hope that this blend of the increased realization and the general architectural formalism of the architectural pattern will help the developers to jump into the development from the requirements, utilizing the abstraction of the architectural pattern as the design. As we continue with the designing and testing of the proposed architectural pattern in our ongoing research, we realized that the assistance gained from the architectural pattern can provide a significant improvement to the RAD of AJAX based RIA engineering.

However, for some systems the architectural and system designing may still be needed due to stake holders related various reasons and criteria, despite the lack of designing practices in RAD. Architectural pattern and the realization served by its architectural formalism, may provide sufficient assistance, in designing a specific system architecture and system design. With the experience gained from the experiments, we think that, the combination of an architectural pattern and some dedicated best practices for AJAX based RIAs, will create a comfortable platform with less complexities for AJAX based RIA engineering, and it will increase the support for RAD significantly.

IV. CONCLUSION

AJAX is a popular and powerful script-based technique for developing the Rich internet Applications. Despite its power, it suffers from various complexities, which make the AJAX based RIA development undergoes difficulties especially in RAD enabled environment. The root of these complexities is the lack of architectural formalism in AJAX based RIAs, which causes a poor realization of the RIAs and AJAX adoption. A good understanding of the AJAX General Architecture or higher usage of CASE tools are not capable of minimizing the difficulty level of implementing AJAX features, which increases with the number of AJAX features in the same page.

Architecture is an essential solution for increasing the realization of a system, hence identification of general architectural formalism of AJAX based RIAs is important to address the complexities and reduce the difficulties. Since designing phase is less used or avoided in RAD methodologies, having an architectural pattern – which can be utilized as the system architecture – is significantly important. A general RIA architectural pattern with best practises, may increase the realization of the architectural properties of RIAs and AJAX adoption, while providing a good platform for starting the development early, without the architectural and system design.

REFERENCES

Dissanayake, N. R. & Dias, K., 2014. Essential features a general AJAX Rich Internet Application architecture should have in order to support Rapid Application Development. Thailand, s.n.

Dissanayake, N. R. & Dias, K., 2014. What does the AJAX Rich Internet Applications need to support the Rapid Application Development. Sydney, Australia, s.n., pp. 1-4.

Dissanayake, N. R., Dias, K. A. & Jayawardena, M., 2013. An analysis of rapid application development of AJAX based Rich Internet Applications. Colombo, Sri Lanka, s.n., p. 284.

Farrell, J. & Nezlek, G. S., 2007. Rich Internet Applications The Next Stage of Application Development. Cavtat, Croatia, s.n., pp. 413 - 418.

Lawton, G., 2008. New Ways to Build Rich Internet Applications. Computer, August, 41(8), pp. 10 - 12.

Li, J. & Peng, C., 2012. jQuery-based Ajax General Interactive Architecture. Beijing, s.n., pp. 304-306.

LIN, Z., WU, j., ZHANG, Q. & ZHOU, H., 2008. Research on Web Applications Using Ajax New Technologies. s.l., s.n., pp. 139 - 142.

Mesbah, A. & Deursen, A. v., 2007. An Architectural Style for AJAX. Mumbai, s.n.

Paulson, L. D., 2005. Building rich web applications with Ajax. Computer, Oct, 38(10), pp. 14 - 17.

Piero, F., Gustavo, R. & Fernando, S.-F., 2010. Rich Internet Applications. Internet Computing, IEEE, 14(3), pp. 9-12.

Preciado, J., Linaje, M., Sanchez, F. & Comai, S., 2005. Necessity of methodologie to model Rich Internet Applications. s.l., s.n.

Selfa, D. M., Carrillo, M. & Boone, M. d. R., 2006. A Database and Web Application Based on MVC Architecture. s.l., IEEE Computer Society, p. 48.

Solutions, A., 2014. The Importance of Software Architecture. s.l.:Architech Solutions.

Sommerville, I., 2011. Software Engineering. India: Dorling Kindersley.

BIOGRAPHY OF AUTHORS

[1]Nalaka Ruwan Dissanayake born in Anuradhapura Sri Lanka on 1982, is working as an assistant lecturer in Software Engineering at Informatics Institute of Technology (IIT), Colombo 6, Sri Lanka; who is also reading for an M.Phil program as a student of University of Colombo School of Computing, Colombo 7, Sri Lanka. He graduated as a Bachelor of Science with a Special honors degree in Software Engineering from Sri Lanka Institute of Information Technology (SLIIT), Sri Lanka on 2007. Nalaka started his academic career as a Student Instructor in SLIIT when he was studying the final year as an undergraduate. After completing the degree he continued as an Instructor in SLIIT. Since then up to date Nalaka has worked in both state and private universities and institutes SLIIT, Rajarata University of Sri Lanka (RUSL), IIT, SLIATE, ESoft as an academic personal over six years. With the experience he has gained over a year in industry as a Software Designer, he works on Research and Development parallel to his studies and academic career. With a keen interest in researching areas such as Software Engineering, Architectures, Rapid Application Development and Rich Internet Applications, Nalaka spends his free time experimenting for new techniques and strategies in Rich Internet Application design and development. nalakadmnr@gmail.com

[2]G.K.A. Dias received his BSc in Physical Science (1982) from the University of Colombo, Sri Lanka, Postgraduate Diploma in Computer Studies from University of Essex, UK (1986), and MPhil in Computer Science from University of Wales, UK (1995). He is currently a senior Lecturer, Head/Department of Communication and Media Technologies of the University of Colombo School of Computing (UCSC). He is also the Coordinator of the M.Phil Program at the UCSC. He is a member of the Computer Society of Sri Lanka, Sri Lanka association for Advancement of Science (SLASS) and a member of the ACM the world's largest educational and scientific computing society. His Research Interests are Computer Aided Software Engineering, Multimedia for Education, Modelling and Simulation. gkad@ucsc.lk