

# Ajaxifying Classic Web Applications

Ali Mesbah

Software Engineering Research Group  
Delft University of Technology, The Netherlands

A.Mesbah@tudelft.nl

## Abstract

Recently, a new web development technique for creating interactive web applications, dubbed AJAX, has emerged in response to the limited degree of interactivity in large-grain stateless web interactions. In this new model, the web interface is composed of individual components which can be updated/replaced independently, and the client/server communication is based on a delta-communication style of interaction. With the rise of AJAX web applications classical multi-page web applications are becoming legacy systems. This research seeks to explore how software engineering techniques can assist in comprehending, analyzing, and restructuring classic web applications towards AJAX.

## 1 Introduction

Recently, a new web development technique, dubbed AJAX (Asynchronous JavaScript And XML) [2], has emerged for creating interactive web applications through the use of HTML, CSS, DOM, XML, JavaScript, and the XMLHttpRequest object.

In this new model, the (single-page) web interface is composed of individual components which can be updated/replaced independently, so that the entire page does not need to be reloaded on each user action. This, in turn, helps to increase the levels of interactivity, responsiveness and user satisfaction. Well known examples of AJAX web applications include Google Suggest, Google Maps, Meebo and Gmail.

AJAX web applications have attracted a strong interest in the web development community. Many organizations are beginning to consider migration, also called *ajaxification*, possibilities to this new paradigm. Enormous amounts of investments have been spent on web-based systems and most organizations are not likely to throw these valuable assets away. As a result, the well-known problem of legacy migration is becoming increasingly important for web applications. If until a year ago, the problem revolved around

migrating traditional systems to web-based settings, today we have a new challenge of re-engineering classic web applications to AJAX applications.

## 2 Problem Definition

The main focus of this work is on the migration of classic page-sequence web applications to (single-page) AJAX settings. We will address the problem of identifying and defining suitable software engineering methods and techniques to support the migration process. The migration process has generally three essential parts that we have to meet: comprehension of the target (AJAX) setting's properties, comprehension, analysis, and reconstruction of the source (classic multi-page web application) system, and transformation of the source system to the target setting.

## 3 Hypothesis

It would be credulous to think that the model reconstruction of every non-trivial web application and the migration process to various AJAX settings can be conducted *automatically*. Our hypothesis, however, is that *reverse engineering* techniques and tools to reconstruct abstract models of the source application and *model-driven and architecture-driven* software development approaches to understand the differences and perform transformations between the source and target models, can *assist* us in conducting the migration process. Our goal is to provide tool support that can be integrated and deployed for understanding, analyzing, and migrating classic web applications to the new AJAX paradigm.

## 4 Approach

In this research, we first explore whether concepts and principles as developed in the software architecture research community, and specifically those related to network-based environments [1], can be of help in formalizing the architectural properties of AJAX applications. In particular, we

propose SPIAR [5], an architectural style for AJAX applications consisting of the architectural - processing, connecting, and data - elements of AJAX, and the constraints that should hold between them in order to meet desired properties such as *user interactivity*, *user-perceived latency*, and *portability*.

Once an abstract perspective on AJAX settings has been gained, we focus on the classical web applications and the ways such applications could be restructured to AJAX architecture. Our assumption is that a web application already exists, and our intension is to explore techniques to support its comprehension, analysis, and restructuring by recovering abstract models from the current implementation. To this end, we conduct *static* as well as *dynamic* analysis of the source code, adopting *web application reverse engineering* paradigm which includes the application of traditional software engineering techniques.

*User interface reverse engineering* will also play an important role in our approach as the user interface interaction model of the source (page-sequence) and the target (single-page) systems are substantially different. Reconstructing a user interface model of the source multi-page web application is a key element in the migration process. To that end, we propose [6] a user interface migration process consisting of four major steps: Navigational Path Extraction, UI Component Model Identification, Single-page UI Model Definition, and Target UI Model Transformation

Our migration approach also involves the definition of appropriate meta-models for source and target systems involved in the process and the corresponding transformation rules. We will investigate to what extent Model-driven engineering (MDE) technologies such as the UML, the Meta-Object Facility (MOF), and the Atlas Transformation Language (ATL) can help in restructuring existing web applications towards AJAX.

## 5 Validation Strategy

Validity of our approach and solutions will be assessed by extensive empirical work. We use case studies [3] to investigate the applicability of our findings. Industrial Real-world web applications will be used to perform experiments validating or discovering the limitations of our migration techniques. We have close collaborations with software practitioners in industry, in particular with Backbase, one of the leading AJAX companies.

## 6 Related Work

To our knowledge, there is no research work on the ajaxification of classic web applications available to date; this is partly because AJAX is a new approach that is just beginning to evolve into a mature technology. There is, however,

an increasing body of knowledge on the employment of traditional software engineering methodologies, techniques, and tools to assess the analysis, maintenance, and restructuring of web applications [8]. An overview of the related work can be seen in our recent migration paper [6].

Our SPIAR architectural style for AJAX draws from many existing styles and software fields [1, 7], discussed and referenced in the paper [5]. SPIAR relates closely to the software engineering principles of the REST style [1]. While REST deals with the architecture of the Web as a whole, SPIAR focuses on the specific architectural decisions of AJAX frameworks.

## 7 Contributions

Our expected contributions include the development of concepts, models, and methods to: 1) better comprehend and analyze classic and AJAX web applications, 2) model client/server interactions in AJAX applications, 3) support reverse engineering of classic web applications to abstract models, and 4) support semi-automatic migration of classic web applications to AJAX applications. We have booked initial results [4, 5, 6] in each of these areas and are presently elaborating them through application to large case studies.

### Acknowledgments

*This work received partial support from SenterNovem, project Single Page Computer Interaction (SPCI).*

## References

- [1] R. Fielding. *Architectural styles and the design of network-based software architectures*. PhD thesis, UC, Irvine, Information and Computer Science, 2000.
- [2] J. Garrett. Ajax: A new approach to web applications. Adaptive path, 2005. <http://www.adaptivepath.com/publications/essays/archives/000385.php>.
- [3] B. Kitchenham, L. Pickard, and S. L. Pfleeger. Case studies for method and tool evaluation. *IEEE Softw.*, 12(4):52–62, 1995.
- [4] A. Mesbah and A. van Deursen. Crosscutting concerns in J2EE applications. In *WSE '05: 7th Int. Symp. on Web Site Evolution*, pages 14–21. IEEE Computer Society, 2005.
- [5] A. Mesbah and A. van Deursen. An architectural style for Ajax. In *WICSA '07: 6th Working IEEE/IFIP Conference on Software Architecture*. IEEE Computer Society, 2007.
- [6] A. Mesbah and A. van Deursen. Migrating multi-page web applications to single-page Ajax interfaces. In *CSMR '07: 11th European Conference on Software Maintenance and Reengineering*. IEEE Computer Society, 2007.
- [7] D. E. Perry and A. L. Wolf. Foundations for the study of software architecture. *SIGSOFT Softw. Eng. Notes*, 17(4):40–52, 1992.
- [8] F. Ricca. *Analysis, Testing and Re-structuring of Web Applications*. PhD thesis, University of Genova (DISI), 2003.