



ELSEVIER

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico



Analyzing best practices on Web development frameworks: The lift approach



María del Pilar Salas-Zárate^a, Giner Alor-Hernández^{b,*},
Rafael Valencia-García^a, Lisbeth Rodríguez-Mazahua^b,
Alejandro Rodríguez-González^{c,e}, José Luis López Cuadrado^d

^a Departamento de Informática y Sistemas, Universidad de Murcia, Campus de Espinardo, 30100 Murcia, Spain

^b Division of Research and Postgraduate Studies, Instituto Tecnológico de Orizaba, Mexico

^c Bioinformatics at Centre for Plant Biotechnology and Genomics, Polytechnic University of Madrid, Spain

^d Computer Science Department, Universidad Carlos III de Madrid, Spain

^e Department of Engineering, School of Engineering, Universidad Internacional de La Rioja, Spain

ARTICLE INFO

Article history:

Received 1 October 2013

Received in revised form 18 December 2014

Accepted 19 December 2014

Available online 5 January 2015

Keywords:

Best practices

Lift

Scala

Web frameworks

ABSTRACT

Choosing the Web framework that best fits the requirements is not an easy task for developers. Several frameworks now exist to develop Web applications, such as Struts, JSF, Ruby on Rails, Grails, CakePHP, Django, and Catalyst. However, Lift is a relatively new framework that emerged in 2007 for the Scala programming language and which promises a great number of advantages and additional features. Companies such as Siemens© and IBM©, as well as social networks such as Twitter© and Foursquare©, have now begun to develop their applications by using Scala and Lift. Best practices are activities, technical or important issues identified by users in a specific context, and which have rendered excellent service and are expected to achieve similar results in similar situations. Each framework has its own best practices whose aim is to facilitate the development of Web applications. However, there is no current comparative analysis that identifies the best practices for Web frameworks. Thus, as its salient contribution, this paper identifies a set of best practices for Web frameworks. Afterwards, these best practices were analyzed and discussed in terms of developing Lift-based Web applications. The identification of these best practices would allow developers to construct more interactive and efficient Lift-based Web applications, integrating features of Web 2.0 technologies with less effort and exploiting the frameworks' benefits. In addition, this paper contains a comparative analysis with Web frameworks such as JSF, Struts, CakePHP, Ruby on Rails, Lift, Django, and Catalyst. Finally, as proof of concept, a set of Lift-based Web applications were developed for this paper by applying best practices such as actors, lazy loading, Comet support, SiteMap, Wiring, HyperText Markup Language, version 5 (HTML5) support, and parallel rendering.

© 2015 Elsevier B.V. All rights reserved.

* Corresponding author.

E-mail addresses: mariapilar.salas@um.es (M.P. Salas-Zárate), galor@itorizaba.edu.mx (G. Alor-Hernández), valencia@um.es (R. Valencia-García), lrodriguez@itorizaba.edu.mx (L. Rodríguez-Mazahua), alejandro.rodriguez@upm.es (A. Rodríguez-González), jllopez@inf.uc3m.es (J.L. López Cuadrado).

<http://dx.doi.org/10.1016/j.scico.2014.12.004>

0167-6423/© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Choosing the appropriate Web framework for Web development, which best fits the developers' requirements, is not an easy task, since there are many frameworks based on different languages. Moreover, selecting an inappropriate framework can lead to 1) wasting time studying the details of another language, 2) failure to meet the required time because developers are not used to the framework, and 3) spending time taking corrective actions to choose a different framework. In order to avoid these problems, it is highly important to know and identify the best practices for Web development.

A best practice is a process, technique, innovative use of technology, or a set of resources with a proven record of success in providing significant improvements in cost, schedule, quality, performance, safety, environment, or other measurable factors that impact on an organization [1]. Best practices on Web frameworks imply reducing development time and effort while saving money, increasing the quality of code, and providing the opportunity to create friendly and interactive applications.

A framework is a high-level solution for the reuse of software pieces, a step forward in simple library-based reuse that allows for sharing common functions and generic logic of a domain application. It also ensures a better level of quality of the final product, since one important part of the application is already found within the framework and, therefore, has already been tested [2]. Many Web frameworks based on different programming languages are now available: JSF and Struts for Java-based Web frameworks, Ruby on Rails are Ruby-based, Grails is Groovy-based, and CakePHP is for PHP-based frameworks. However, another brand new type of Web framework emerged in 2007. Lift is a Scala-based framework, whose features are based on the experience of David Pollak, its creator, with the mistakes from other Web frameworks [3]. Lift possesses the advantages of Scala's functional programming. Functional programming (FP) is a programming style emphasizing on functions that return consistent and predictable results regardless of a program's state. As a result, a functional code is easier to test and reuse, simpler to parallelize, and less prone to bugs [4]. Scala is a hybrid functional and Object-oriented (OO) programming language, which means getting the power of the higher-level functional languages (such as Haskell, Scheme, among others) while retaining the modularity and reusability of OO components. FP concept of immutability is a specially well-represented feature in Scala, and it is one of the simplest means to provide high scalability. Scala allows for doing more in Lift with fewer lines of code [5].

Existing literature of Lift framework includes merely six books: 1) Simply Lift [6], 2) Exploring Lift [7], 3) Lift in action [8], 4) Lift Cookbook [9], 5) Lift Web Applications How-to [10], and 6) Entwicklung von Web-Applikationen mit Lift und Scala [11]. However, none of this information has yet provided a comparative analysis and a precise detection of Lift best practices in comparison to other Web frameworks.

Thus, this work presents a set of best practices to develop Lift-based Web applications. Moreover, this paper includes a comparison and discussion carried out among other well-known Web frameworks such as JSF, Ruby on Rails, Struts, Grails, CakePHP, Lift, Django, and Catalyst. Finally, the paper presents also the development of a set of Lift-based Web applications in which the best practices were applied and analyzed.

This paper is structured as follows: Section 2 presents the state of the art concerning Lift-based development and the use of best practices on Web development. Section 3 describes and discusses the use of best practices for Web development and presents a comparison table of Web frameworks in the use of best practices. Finally, the fourth section presents a set of Lift-based Web applications applying best practices, while future directions and the concluding remarks are presented in Section 5.

2. State of the art

Research has been proposed in order to identify and obtain the best practices for Lift-based applications development and Web development. Chen et al. [7] discussed the many advantages of using Lift framework and claim that Lift: 1) is a great framework for building compelling Web applications, 2) has been designed to make powerful techniques easily accessible, while keeping the overall framework simple and flexible, and 3) has cherry-picked the best ideas from a number of other frameworks, while creating some other novel ideas. These advantages are a combination of solid foundations and new techniques that make Lift so powerful.

Several research works have addressed some Lift and Scala issues. For instance, Pollak et al. [12] published the development of their multiuser and real-time chat application on Lift. This application provided a single chat server that took chat messages and redistributed them out to all listeners. In the same work, authors also presented Scala's language features such as singletons, pattern matching, traits, and immutable data types.

Wampler [13] equally addressed Scala's support for actors and messages for other Scala-based frameworks that are full-stack frameworks for building multi-tier applications. Some frameworks are "point" tools for specific parts of an application, like template libraries for generating webpages (analogous to Java Server Pages), while others focus on building particular kinds of networked servers like Representational state transfer (REST) response servers that are "headless." However, Play, Scalatra and Finagle service frameworks were only presented and discussed in Wampler's work.

Dong-Hong et al. [14] presented a new framework based on Scala and Lift which tried to solve the conflict of rapid delivery and repeated work. This framework was a new approach for agile software development. It could perform all general functions of information systems, including create, read, update, and delete (CRUD) operations and it had features

Download English Version:

<https://daneshyari.com/en/article/433669>

Download Persian Version:

<https://daneshyari.com/article/433669>

[Daneshyari.com](https://daneshyari.com)