INTRODUCTION

Guest editors' introduction to the special section from the international symposium on web systems evolution

Ying Zou · Ji Wu · Kenny Wong

Published online: 17 March 2011 © Springer-Verlag 2011

1 Introduction

With the growth of the Internet, there is an urgent need to develop Web-based applications. Initially, the Web was presented as a giant URL-based file server for publishing static information in a hypertext electronic form. With the incorporation of multi-tier architectures and internet programming techniques, such as CGI scripts, Java servlets and Ajax, the Web is evolved into an interactive medium that provides dynamic information services. Nowadays, Web services are developed as self-contained, self-describing, and modular applications that can be published, located, and invoked across the Web. The service-oriented architecturent (SOA) uses loosely coupled Web services as basic constructs to build more complex systems in a flexible and rapid way [3].

The Web is gradually adopted as the development and deployment platform for distributed computing. Organizations would like to take advantage of the Web in its various forms (i.e., Internet, Intranet, and Extranets) for their enterprise computing. This allows them to leverage functionality from their existing legacy systems without having to rebuild

Y. Zou (⊠) Department of Electrical and Computer Engineering, Queen's University, Kingston ON, Canada e-mail: ying.zou@queensu.ca

J. Wu

Department of Computer Science, Beihang University, Beijing, China e-mail: wuji@buaa.edu.cn

K. Wong

Department of Computer Science, University of Alberta, Alberta, Canada

e-mail: kenw@cs.ualberta.ca

these systems. Furthermore, heterogeneous applications and systems (e.g., legacy systems) are built on the plethora of operating systems, languages, networking protocols, and data representations. With its universal access across heterogeneous hardware and software, the Web is the best choice to address this challenge. However, Web applications tend to be frequently changed to adopt changing user requirements. To produce high quality Web applications, intensive testing and maintenance activities are needed for Web applications.

A significant amount of effort from industry and academics intend to provide infrastructure, languages and tools to design, develop and evolve various Web applications. The 11th IEEE International Symposium on Web Systems Evolution was co-located with the 25th IEEE International Conference on Software Maintenance (ICSM 2009) in Edmonton, Alberta, Canada. Previously known as the Web Site Evolution series of events, WSE began in 1999 as a lively forum for researchers and practitioners to present original work on topics related to the evolution of Web-based systems. The discussion has broadened in recent years, moving from the modernization of legacy systems to include topics, such as Web services, Web 2.0, and cloud computing. In a sense, WSE has itself evolved in step with the ever- changing technologies embodied in Web-based sites, software, services, and systems (http://www.websystemsevolution.org).

The theme of this special section focuses on recent achievements in the evolution of Web systems. In particular, the evolution of Web systems involves activities, such as improving the quality of Web systems to deliver reliable services, enhancing the maintainability of Web applications for changing requirements and migrating a stand-alone system to the Web. Reverse engineering, program understanding, software testing and software migration are well-studied techniques in the area of software maintenance. This special



section provides an overview of the most promising results achieved in these fields. In addition to the theme, other topics are also considered in WSE 2009, including:

- Model-driven re-engineering of Web systems
- Reverse engineering and analysis of Web systems
- Security and privacy in Web systems
- Quality assessment of Web systems
- · Accessibility of Web systems
- Multilingual Web systems
- Mobile Web systems
- Semantic Web
- Tools, environments, and technologies for Web systems
- Empirical studies of Web systems.

2 Selected papers

Five papers are selected to this special issue. These five selected papers include three that discuss the approaches for enhancing testing activities for Web applications; one that discusses the approach for improving maintenance of service oriented applications and one that presents techniques for porting software visualization tools into the Web which is used as the software execution platform for a larger user base.

The first of the three testing papers by Kinga Dobolyi, Elizabeth Soechting and Westley Weimer, "Automating Regression Testing Using Web-based Application Similarities", describes a regression testing technique that automatically compares the similarity of the outputs from two versions of Web applications [2]. A tool, called SMART, is developed to provide a highly precise oracle comparator. This approach reduces the efforts in the traditional methods for regression testing where manual inspection is required. This approach can predict which test cases merit human inspection. The result of the approach is 2.5–50 times as accurate as the current industrial practice without user annotations.

The second of the three testing papers by Alessandro Marchetto1, Roberto Tiella1, Paolo Tonella, Nadia Alshahwan and Mark Harman, "Crawlability Metrics for Automated Web Testing", proposes an automatic testing technique to achieve high coverage of the application under investigation [4]. A crawler is designed to automatically explore the structure of the application under testing with given crawling capabilities. The proposed approach defines structural metrics to predict whether the crawler will be sufficient or not to achieve high coverage of the application. As a result of experiments, some of the proposed metrics are considered as strong indicators of crawlability. Around 80% of Web pages can be covered by crawlers. Looking ahead, the future work will focus on developing more met-

rics on the server side and on the predictive capability of the crawler.

The last of the three testing papers by Marco Torchiano, Filippo Ricca and Alessandro Marchetto, "Are Web applications more defect-prone than Desktop applications in their presentation layer?", presents an experiment to study the distribution of bugs among different layers (i.e., presentation layer, business logic and data logic) of Web and Desktop applications [6]. The goal of this work is to address the question: Are Web applications more, less or equally defectprone in their presentation, business and data layers than desktop ones? The experimental study provides a strong evidence that the presentation layer in Web applications is more defect-prone than the analogous layer in Desktop applications. An additional factor influencing the distribution of defects is represented by the application domain. Looking ahead, the future work will provide a quantitative evaluation of the causes for the higher presentation defectiveness of Web applications.

The fourth paper of the five selected papers by Adina Mosincat, and Walter Binder, "Automated Maintenance of Service Compositions with SLA Violation Detection and Dynamic Binding", presents a framework, named ADULA, that dynamically monitors, automatically detects, diagnoses and repair the performance degradation of BPEL processes [5]. The goal of the work allows the BPEL processes to automatically bind to services in order to meet the changing service level agreements. ADULA ensures that the service replacement is traceable. The solution uses light weight monitoring techniques, supports customizable diagnosis and repair strategies, and is compatible with any standards-compliant BPEL engines. The evaluation shows that ADULA maintains process performance and ensures Service Level Agreement (SLA) compliance with moderate costs in terms of overhead. The future research will extend the monitoring and detection capabilities to domain-specific and functional quality of services.

Finally, the last paper of the special section by Marco D'Ambros, Michele Lanza, Mircea Lungu, and Romain Robbes, "On Porting Software Visualization Tools to the Web", describes their experiences for porting two software visualization tools into the Web [1]. The goal of this work allows software visualization tools to be executed as Web applications to reduce the difficulty in the tool installation and make them more accessible to a larger user base. The paper also discusses the promises and perils that go with such an endeavour and presents a number of technological alternatives that are available to implement Web-based visualization tools.

This issue is a result of the tremendous effort of many people. The guest editors would like to thank Ms. Julia Rehder, Professor Bernhard Steffen for the guidance in the reviewing processes, the authors, and the reviewers for their hard work.



References

- D'Ambros, M., Lanza, M., Lungu, M., Robbes, R.: On porting software visualization tools to the web. Int. J. Softw. Tools Technol. Transfer. (2011). doi:10.1007/s10009-010-0171-9
- Dobolyi, K., Soechting, E., Weimer, W.: Automating regression testing using web-based application similarities. Int. J. Softw. Tools Technol. Transfer. (2011). doi:10.1007/s10009-010-0170-x
- Lewis, G., Smith, D., Kontogiannis, K.: 'A Research Agenda for Service-Oriented Architecture (SOA): Maintenance and evolution of service-oriented systems. Technical Note, Software Engineering Institute at the University of Carnegie Mellon. http://www.sei.cmu. edu/reports/10tn003.pdf, Last Accessed, Febuary (2011)
- Marchetto1, A., Tiella1, R., Tonella, P., Alshahwan, N., Harman, M.: Crawlability metrics for automated web testing. Int. J. Softw. Tools Technol. Transfer. (2011). doi:10.1007/s10009-010-0177-3
- Mosincat, A., Binder, W.: Automated maintenance of service compositions with SLA violation detection and dynamic binding. Int. J. Softw. Tools Technol. Transfer. (2011). doi:10.1007/ s10009-010-0181-7
- Torchiano, M., Ricca, F., Marchetto, A.: Are Web applications more defect-prone than desktop applications in their presentation layer? Int. J. Softw. Tools Technol. Transfer. (2011). doi:10.1007/ s10009-010-0182-6

