Journal of Web Engineering, Vol. 13, No. 3&4 (2014) 181–182

EDITORIAL

For several years, research in Web engineering was dominated by Web services, the Service-Oriented Architecture (SOA) and Service-Oriented Computing (SOC) [5, 1]. Front-end development was considered an additional implementation effort needed to allow users to access the wealth of services and composites that were enabled by Web services and similar back-end APIs (e.g., most notably, RESTful services [4]). The advent of mashups [2], i.e., of generic Web applications that are developed starting from resources available on the Web (typically RSS or Atom data feeds, Web services and pieces of user interfaces), has changed this situation: mashups established integration at the user interface (UI) level [3] and fostered the emergence of suitable component technologies, such as W3C or OpenSocial widgets. This, in turn, fuelled a new wave of research and component-based development practices that specifically focus on the UI of applications, rather than on their back-end. Given the recent advancements of client-side, in-browser technologies (e.g., HTML 5 and AJAX), this focus typically comes with special attention for the client side and end-user-oriented concerns.

This special issue does not want to limit its focus to mashups and similar composition practices. It rather aims to provide an overview of the *issues, advancements and opportunities* that *component-based and client-oriented development practices* imply in general for modern Web engineering and to provide the reader with insight into the current state of the art. This special issue is the result of a selection of the best papers presented at the 13th International Conference on Web Engineering (ICWE 2013, `http://icwe2013.webengineering.org/`), which was held in July 8–12, 2013, in Aalborg, Denmark. Although this special issue therefore represents a bottom-up effort, interestingly all selected papers fit into the above outlined focus, which confirms the importance of the topic and the timeliness of this special issue.

The selection of articles included in this special issue comprises four contributions that, together, provide a good overview of which problems and research questions motivate and challenge the Web engineering community of today. Specifically:

- Leone et al. ("Component-based Web Engineering using Shared Components and Connectors") look at the problem of components in a holistic fashion, without explicitly focusing on either the client or the server side. Motivated by the lack of suitable component technologies in the contexts of generic Web development frameworks and content management systems (CMSs), they propose a component model that supports composition at the level of the schema and data, the application logic and the user interface. They specifically show its applicability in Symfony and WordPress.

- Kovachev et al ("DireWolf: A Framework for Widget-Based Distributed User Interfaces") focus on client-side UI widget technologies and describe their framework for the development of Web applications that are naturally distributed over multiple devices (smart phones, tablets and computers) and, hence, require the live synchronization

of UI widgets across different devices and browsers. The proposed approach features inter-widget communication and seamless user session mobility.

- Ast et al. ("Efficiently Developing Progressively Enhanced Web Applicationsby Sharing Presentation and Business Logic Between Server and Client") open the curtains and take a look inside how client-side components can be developed. They specifically concentrate on the problem of distributing application logic between the client (JavaScript in the browser) and the server (JavaScript in node.js) and describe their SWAC framework, which particularly suits the practice of progressive enhancement.

- Choudhary et al. ("Model-Based Rich Internet Applications Crawling: Menu and Probability Models") finally provide state-of-the-art insight into how to crawl and interpret Web applications that make use of client-side technologies, such as AJAX, a common problem of Web crawlers. The authors propose two model-based crawling techniques able to reconstruct the possible in-browser navigation states, along with the respective transitions, and test the performance of the two approaches.

These four contributions nicely integrate with each other: Leone et al. and Kovachev et al. study the problem of development with components and widgets, while Ast et al. study how to best develop components, and Choudhary et al. study how all these new practices are manifest inside the client browser and how they affect what crawlers can automatically extract from modern Web applications.

We strongly believe the reader will appreciate the quality of these works and would like to thank the authors for their high-quality and meticulous work. We also would like to thank the reviewers of this special issue (all papers underwent an additional round of reviews, independently of the paper selection process of ICWE 2013) for their commitment and competent feedback.

Florian Daniel, University of Trento, Italy (daniel@disi.unitn.it)
Peter Dolog, Aalborg University, Denmark (dolog@cs.aau.dk)
Qing Li, City University of Hong Kong, China (itqli@cityu.edu.hk)

Guest Editors

References

1. Gustavo Alonso, Fabio Casati, Harumi Kuno, and Vijay Machiraju. *Web Services: Concepts, Architectures, and Applications*. Springer, 2003.
2. Florian Daniel and Maristella Matera. *Mashups: Concepts, Models and Architectures*. Springer, 2014.
3. Florian Daniel, Jin Yu, Boualem Benatallah, Fabio Casati, Maristella Matera, and Regis Saint-Paul. Understanding UI Integration: A survey of problems, technologies. *IEEE Internet Computing*, 11(3):59–66, may/jun 2007.
4. Roy Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. Ph.d. dissertation, University of California, Irvine, 2007.
5. Michael P. Papazoglou. *Web Services - Principles and Technology*. Prentice Hall, 2008.