Web-Wide Application Customization:
The Case of Mashups

Stephan Hagemann, XING AG, Germany

Gottfried Vossen, University of Muenster, Germany

ABSTRACT

Mashups have become popular in recent years in the context of Web 2.0 developments. They represent a way of allowing an application to grow beyond the capabilities of its original developers through an incorporation of external functionality. This paper first introduces several approaches to integrating mashups into the Web pages or services, which commonly implement ways to determine which mashups are potentially relevant for display in a certain Web page context. It then describes in detail a novel approach called ActiveTags, which enables users to create reliable mashups based on tags and hence customized views of Web pages with tagged content. A scenario that demonstrates the potential benefits of this approach is presented. Moreover, a formalization is presented which suitably combines previous work on modeling the Web with relational meta-programming, thereby showing that ActiveTags (as well as related approaches) can conceptually be described in terms of the relational model of data.

Keywords: Application Customization, Mashups, Meta-Programming, Social Tagging, Web Customization

INTRODUCTION

Web applications ("Web apps") have been developed very rapidly over the past years. Their development is different from the development of "classic" applications in that Web apps are perceived and delivered as services, which "impacts the entire software development and delivery process" (Musser & O'Reilly, 2007). What is imminent to both kinds of applications, traditional and Web, is that there can never be a full accordance between providers, users, or other stakeholders as to which features a certain application should have. The utility of new features and functions differs between the stakeholders, and there will always be functions that will not be implemented by the provider (although they are requested by users). With their ever growing importance, the need for customizability henceforth also increases. Moreover, as Macías and Paternò (2008) note, "there is an ongoing shift to end-user centered technology, and even users with poor or no skill in Web-based languages may feel the need to customize Web applications according to their preferences." The present paper studies an approach to Web application customization that

DOI: 10.4018/jismd.2012010102

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

fits particularly well with frameworks whose purpose is to integrate mashups into Web apps.

The problem of application customization is particularly interesting for the context of Web applications. Karger et al. (2009), Lutteroth and Weber (2008), and Macías and Paternò (2008) each deal with different aspects of this. A key observation of Karger et al. (2009) is that "instead of warping their data to fit rigid applications, users should warp applications to fit their data and tasks." The present paper takes a complimentary approach by suggesting practical Web app enhancement through mashups. In the first part of this paper, we present the ActiveTags prototype originally outlined in Hagemann and Vossen (2009), which enables this based on tag-based mashups that are integrated into Web pages on the fly via a browser extension. In the second part, the generality of the approach is documented by a transformation of the application into a relational equivalent.

Attempts to cope with the challenge of application customization from a technical point of view are Application Programming Interfaces (APIs), which allow skilled developers or users to create custom functionalities based on the data and functions provided by Web applications. The number of such API has been rising continuously over the last years. This allows for what Hippel (2005) has called "democratizing innovation." Well-designed API turn Web sites into platforms that make themselves indispensable by providing the grounds for "ecosystems" of applications (cf. Musser & O'Reilly, 2007).

One form of API usage is mashup creation, i.e., Web applications which combine data or functions from one or more sources into one interface. Because of the perceived tentative character, mashups were first considered to be merely toys or gimmicks, but gradually they are receiving more recognition from both research and industry (cf. Ennals et al., 2007). The utilization of mashups in enterprises has been hampered by some important barriers which prevent widespread adoption (cf. Hinchcliffe, 2007). However, many companies, large and small, have started to turn to mashups as a useful option to quickly attain some needed functionality. In fact, the financial crisis that took off in late 2008 has strengthened the role of mashups in companies seeking cost savings and improved operations (cf. Rodier, 2009). Offering API enables third parties to develop functionality externally. However, the prerequisites are still high, as they require programming skills, knowledge of the particular API, and the appropriate infrastructure to run on.

Mashups typically do not, however, integrate with the Web page they are based upon. So, although they may be adding relevant functionality to an application, they are still perceived as separate from their subjacent services. While this may be beneficial in many use cases (as the number of mashups presented on programmableweb.com shows), it vastly prevents them from being used for application customization.

This paper therefore takes a look at mashup frameworks that can be used to integrate mashups into Web applications, thereby allowing these to be extended in any desired way. The running example will be ActiveTags, which is the prototype of a Firefox extension (Hagemann & Vossen, 2009) developed as part of Hagemann (2009). It has the additional benefit of allowing Web-wide customizations. In order to do so, *tags* are used as a layer separating Web applications and mashups. Its approach takes advantage of the fact that there are quite a number of sites that provide a platform for communication and interaction, but little or no original content. Such sites intrinsically function as platforms for User-Generated Content (UGC): users explicitly and implicitly create data through their usage of a site. One form of UGC that has found particular interest in the research community is *tagging* (cf. Voss, 2007), i.e., manual, free-for-all keyword annotation. Typically created in social tagging systems, tags come with a dual role as personal and social metadata (cf. Ames & Naaman, 2007). While tags do not constitute the most reliable form of metadata, their flexibility and ease of use has made them popular among sites employing UGC. Despite being uncontrolled, tags have evolved beyond their support of search and browsing.

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

23 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/article/web-wide-application-customization/61394

Related Content

Reduction of Defect Misclassification of Electronic Board Using Multiple SVM Classifiers

Takuya Nakagawa, Yuji Iwahoriand M. K. Bhuyan (2014). *International Journal of Software Innovation (pp. 25-36).*

www.irma-international.org/article/reduction-of-defect-misclassification-of-electronic-board-using-multiple-svm-classifiers/111448

A Three Layered Approach for General Image Retrieval

Richard Chbeir, Youssef Amgharand Andre Flory (2002). *Optimal Information Modeling Techniques (pp. 78-83).*

www.irma-international.org/chapter/three-layered-approach-general-image/27826

Security Gaps in Databases: A Comparison of Alternative Software Products for Web Applications Support

Afonso Araújo Netoand Marco Vieira (2011). *International Journal of Secure Software Engineering (pp. 42-62).*

www.irma-international.org/article/security-gaps-databases/58507

Malware Analysis and Classification

Jairaj Singhand Kishore Kumar Kumar Senapati (2023). *Malware Analysis and Intrusion Detection in Cyber-Physical Systems (pp. 42-63).*

www.irma-international.org/chapter/malware-analysis-and-classification/331299

The Influence of Personality Traits on Software Engineering and its Applications

Adrián Casado-Rivasand Manuel Muñoz Archidona (2014). *Agile Estimation Techniques and Innovative Approaches to Software Process Improvement (pp. 83-95).*

www.irma-international.org/chapter/the-influence-of-personality-traits-on-software-engineering-and-its-applications/100272