Int. J. Human-Computer Studies 87 (2016) 38–64

Contents lists available at ScienceDirect

Int. J. Human-Computer Studies

journal homepage: www.elsevier.com/locate/ijhcs

An environment for End-User Development of Web mashups[☆]

Giuseppe Ghiani[a], Fabio Paternò[a,*], Lucio Davide Spano[b], Giuliano Pintori[a]

[a] CNR-ISTI, Italy
[b] University of Cagliari, Italy

ARTICLE INFO

Article history:
Received 1 October 2014
Received in revised form
11 September 2015
Accepted 20 October 2015
Communicated by Oulasvirta Antti
Available online 1 December 2015

Keywords:
End User Development
Web mashups
User interface development tools

ABSTRACT

End-User Development aims to find novel ways that are suitable and intuitive for end users to create their own applications. We present a graphical environment in which users create new mashups by directly selecting interaction elements, content and functionalities from existing Web applications without requiring the intervention of expert developers. Then, users just need to exploit a copy–paste metaphor to indicate how to compose the selected interactive content and functionalities in the new mashup. The environment is enabled by a Web-based platform accessible from any browser, and is suitable for users without particular programming skills. We describe the architecture of our platform and how it works, including its intelligent support, show example applications, and report the results of first user studies.

1. Introduction

In recent years, the penetration of Internet applications in all work and leisure activities has made it possible for people to use computers in an increasing number of possible contexts, and for an increasingly wide range of tasks. However, end users have a wide variety of interests and requirements, and existing applications often do not support directly the wide dynamic set of tasks they want to accomplish. In order to fill this gap, it is useful to provide environments that allow them to obtain applications that better fit their needs. Indeed, even if they are not professional developers, they are becoming more and more familiar with software technologies and this sets the ground for creating tools for crafting their own solutions. End-User Development (EUD) is a recent discipline that refers to the approaches allowing people without experience in programming to develop their applications, or at least modify them, in order to better support their specific tasks (Lieberman et al., 2006). In this area various research works have started to investigate novel solutions that are suitable and pleasant for end users to create new applications. In Ko et al. (2011) several approaches to end-user software engineering have been reviewed, but the authors dedicated limited attention to emerging EUD approaches in Web environments, which have started to be addressed by recent work (Cypher et al., 2010) and are discussed in the related work section.

In this paper we focus on the Web, since it allows easy access to large amounts of data and applications (e.g. e-commerce sites, social networks, e-mail, etc.). In addition, the Web is a flexible platform that, differently from native applications, implicitly allows customization. Web pages are defined in HTML, and are represented in the browser by a *Document Object Model* (*DOM*). The DOM is inspectable through the browser, from where it can be manipulated in order to modify the content as well as the behavior of the page (i.e. make the page react in a different way when some events occur). It is worth noting that, differently from native applications, such customizations can be made directly by the user, without involving the original page developers.

We moreover focus on end user developers aiming to create new applications starting with components of existing interactive Web applications, i.e. a mashup editor. From the Human–Computer Interaction perspective, mashup refers to a composition of contents and/or features from several sources that determine new client-side interactive applications. In general, Web mashups can combine data, presentations and functionalities from different Web sites into a single, novel, Web application. For example, most of the first mashups were created to combine geographical maps or to better manage photos. Usually, mashup applications are created by developers exploiting Web APIs or programmatically gathering content from existing Web pages. Therefore, creating mashups has required some technical background (such as programming skills) that most Web end users do not have. Some initial ideas for creating Web mashups that we consider in this work were introduced in Ghiani et al. (2011). However, that solution still required manual intervention by people with good technical knowledge in order to create connections for enabling

[☆] This paper has been recommended for acceptance by Oulasvirta Antti.
[*] Corresponding author. Tel.: +39 0503153066; fax: +39 0503152810.
E-mail address: fabio.paterno@isti.cnr.it (F. Paternò).

http://dx.doi.org/10.1016/j.ijhcs.2015.10.008
1071-5819/© 2015 Elsevier Ltd. All rights reserved.

communication among the components of different Web applications, it was limited in terms of components types that could be combined, and did not provide any support for sharing mashups.

In this paper, we present the MashupEditor, a novel environment based on an intelligent environment for End-User Development of Web mashups. The MashupEditor's main goals are: to allow end users to create Web mashups by reusing existing components from different applications, regardless of their technical skills; to create novel ways to combine such components; to support sharing of the results' compositions with other people, possibly via social networks. The environment does not require knowledge of JavaScript, which is often the most problematic part in Web applications development for non-professional end users. In the composition process, end users exploit an intuitive copy–paste metaphor, which has been inspired by the *programming by example* paradigm. In particular, copy–paste examples provided by the users are used to infer how to compose the components of the existing Web applications.

More precisely, the main contributions of the proposed environment are:

- An editor to create new mashup widgets from components of existing Web applications simply using a Web browser, without specific extensions, and a proxy-server that includes scripts that allow users to select the desired components by direct manipulation;
- A method and a supporting tool for composing Web components from different applications through an intuitive and familiar copy–paste metaphor for creating novel Web applications;
- A solution for preserving users' selected preferences regarding the application output when new queries to remote services are submitted.

In the paper, after discussing related work, we describe an example application of our environment. We detail the underlying architecture and functionalities of the environment in order to explain how the intelligent support is provided to users. Then, we report on evaluation of the MashupEditor functionalities, which has been carried out through two user tests. Lastly, we draw some conclusions and provide indications for further work.

2. Related work

In this section, we discuss various contributions relevant to the approach for the creation of Web mashup applications that is presented in this paper. We briefly discuss a representative set of various types of mashup environments for Web applications, already proposed by academic and industrial research groups. We then discuss some of them focusing on criteria relevant from the perspective of end user composition and development.

2.1. Approaches to mashup applications

The term mashup has been used in both research and industrial settings for defining a broad set of environments able to create new applications by composing information from different sources. Generally speaking, different types of mashups are possible depending on the aspects that they are able to compose (data, functionalities, user interfaces). Such an approach is better supported by the Web architecture, and hence has found its main applications in Web settings.

A first distinction among mashups can be made according to their targets: enterprises and consumers. Enterprise mashups are tools for combining resources, applications and heterogeneous data from various sources in order to solve enterprise-related problems. For example, EzWeb Enterprise Mashup (Soriano et al., 2007) is a mashup platform developed by Telefónica[1] targeted to enterprises. A catalog is available on the platform containing a set of predefined gadgets, which can be composed through a piping metaphor defining the execution flow. It is possible to have several pipes running at the same time. Other examples of such kind of mashup tools are IBM Damia and SAP Research Rooftop (Hoyer et al., 2009). Consumer mashups, aimed at Internet users, exemplify the capabilities of Web 2.0 by combining diverse kinds of data from several public sources into information that is then displayed on a Web browser. One example in this area is Yahoo! Pipes[2], a Web environment to make mashups that exploits data from sources such as RSS feeds or Web services. The created mashups can be saved and made publicly available. Both enterprise and consumer mashups are devoted to providing quick solutions to narrow scope problems. In this work, we aim to provide an environment for consumer mashups, exploiting existing Web applications as the information source, rather than providing a set of predefined gadgets created by professional developers. In addition, we avoid the usage of programming languages (based e.g. on imperative or data-flow constructs). In order to define the mashup's behavior, we utilize an intelligent backend for tracking the user's actions and rely on well-known UI interactions (such as copy-and-paste).

According to their architecture, mashup systems can also be classified as client-based or server-based. The client-based ones rely on a Web browser to combine and show data, while server-based mashup systems instead perform analyses and combinations within a Web server, and subsequently forward data to the Web browser for visualization. In this respect, we have a mixed approach, since with our tool the composition is performed on the client side, and it also exploits functionalities provided by a proxy-server.

In general, interest in how to support EUD through mashups has recently increased. Some authors (Soi et al. 2014) have even considered adopting a domain specific approach, which prioritizes intuitiveness over expressive power, even if they mention that developing mashup platforms—domain-specific or not—is complex and time consuming. In the following, we discuss some work carried out in this area and indicate how we contribute to the state of art.

2.2. Page customization

Different tools allow the user to change the layout and/or the behavior of existing Web pages in order to better tailor them to different needs, with different relevant techniques.

A simple but effective way to customize a Web page is to provide an entry point (e.g. a browser extension) for injecting JavaScript code into it. For instance, Greasemonkey[3] is an extension for the Mozilla Firefox browser that allows choosing from among a set of custom scripts when a Web page is accessed. The tool not only lets the user automatically add new contents and combine them with data from other pages, but also hides texts or images (e.g. unwanted advertisements), adds shortcuts to external pages, fills in forms and compares data from several Web sites (e.g. prices from online stores). In order to exploit Greasemonkey, users typically write JavaScript code for accessing and modifying the page Document Object Model (DOM). Scripts are stored in text

[1] http:// www.telefonica.com

[2] http://pipes.yahoo.com/pipes/

[3] https://addons.mozilla.org/it/firefox/addon/greasemonkey