

# A Cross-Platform Software System to Create and Deploy Mobile Mashups

Sandra Kaltofen, Marcelo Milrad, and Arianit Kurti

Centre of Learning and Knowledge Technologies (CeLeKT)

School of Computer Science, Physics and Mathematics

Linnaeus University, Sweden

{sandra.kaltofen,marcelo.milrad,arianit.kurti}@lnu.se

**Abstract.** Changes in usage patterns of mobile services are continuously influenced by the enhanced features of mobile devices and software applications. Current cross-platform frameworks that allow the implementation of advanced mobile applications have triggered recent developments in relation to end-user mobile services and mobile mashups creation. Inspired by these latest developments, this paper presents our current development related to a cross-platform software system that enables the creation of mobile mashups within an end-user programming environment.

**Keywords:** End-user programming, Mobile Mashup, Cross-platform development.

## 1 Introduction

Mobile devices are getting more powerful in terms of computational power and functionalities. GPS sensors, high definition cameras and Internet access are perceived as standard features of today's mobile devices. The market for mobile applications is continuously growing in direct relation to these latest developments. The more powerful these devices will get, the more powerful mobile applications will become. One of the drawbacks in the development of mobile applications is that developers have to face with the restrictions in the mobile phone hardware and the devices' specifications [1]. Furthermore, they need to have knowledge about different programming languages because the Software Development Kits (SDKs) released by the platform creators are usually tied to a specific language. All these developments generate the need of having software systems that provide end-users with a simple way for the creation of mobile applications [2]. Such systems should be able to integrate different features to create cross-platform mobile applications in such a way that they would not be bound to a particular mobile platform. Another central point of focus is *mobile mashups* and the lack of possibilities to create mashups optimized for mobile usage. In order to address these specific issues, we have designed and developed a system that enables the creation and deployment of customized cross-platform mobile mashups using an end-user programming environment.

## 2 Requirements and Proposed Solution

Requirements engineering for web applications follows the general guidelines used for software applications. According to [3], the main functional requirements of web applications need to incorporate organization, application domain, navigation and interaction requirements. Guided by these principles and the outcomes of our literature survey, we have identified a number of features that a mobile mashup system should have: 1. Mobile mashups should not be bound to device specifications and should work cross-platform [1]; 2. The mashups need to access social networks and other Web 2.0 services to exchange data [4]; 3. Previous programming skills should not be required to create mobile mashups [5]; 4. The functionalities for the mashups are offered as customizable components [2]; 5. The environment needs to allow sharing the created mashups with other users [2, 5]; 6. The environment needs to work with visual programming concepts and technologies [6].

**Proposed solution:** Based on the above-mentioned features, we have developed a system as described in figure 1. Different components are used in order to support enhanced customization of mobile mashups. *Service components* are used to provide core functionalities for a mashup; like the access to server site features and important Web APIs and/or their combination. These components are used to allow the user to receive existing content or to create new content by entering personal data. *Layout components* are used to present information and content such as text, images or videos. *Device components* are used to access the advanced features of today's mobile devices like the GPS sensor data, activation of the photo camera and access to the device content. *Additional components* are proposed to provide enhanced extensibility to the software system.

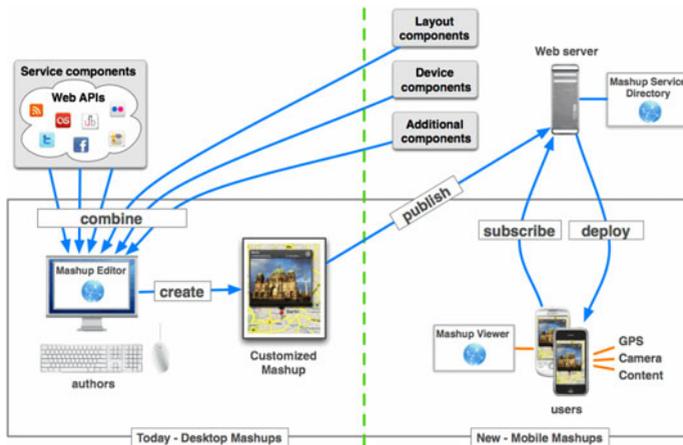


Fig. 1. Overview of the proposed solution

**Implementation:** Our software solution consists of two applications: an end-user programming editor to create the mobile mashups and a viewer to deploy them on a mobile device. The *Mobile Mashup Editor* was implemented as a Rich Internet

Application (RIA) using the *Google Web Toolkit* (<http://code.google.com/webtoolkit>) framework. The editor provides a GUI that enables to combine and to configure mashup components using visual programming concepts and techniques. Using this editor, mashups can be published to a service directory on a web server to make them accessible to other users. The *Mobile Mashup Viewer* runs on the mobile device as a native application. The viewer was implemented using the cross-platform mobile framework *Titanium Mobile* ([www.appcelerator.com](http://www.appcelerator.com)) and it provides access to the remote mashup service directory. In contrast to desktop mashups, mobile mashups have to run within the viewer application and not within a web browser. This is necessary to be able to access the sensor data and the content of mobile devices within a mashup. For that reason, the editor creates and publishes the mashup not as a mobile web application but rather as description of the created mashup. This description contains the mashup structure and settings, its components and their configuration. The viewer can interpret this description and deploys the mobile mashup application out of it. The proposed software solution is cross-platform; both on the editor and the viewer site.

### 3 Cross-Platform Mobile Mashup Example

A mobile mashup can contain different pages and components that are added to each specific application. When a component is added to a page it can be configured, resized and positioned on it. A prototype of the *Mobile Mashup Editor* is shown in figure 2 and contains an example with three pages. The example below has three different components: 1) A *service component* for a Google Map that shows either an address the author can configure or a location information received through a Web API, 2) A *layout component* with a label and an entered mailing address text and 3) A *device component* for calling a configured telephone number. Within the *Mobile Mashup Viewer* application, published mashups can be deployed. The deployed example mashup has been tested on the iPhone and Android platforms as shown in figures 3 & 4. After the deployment, the user can navigate through the different mashup pages. The layout of the pages and the functionalities are equal to the designed pages and the configuration the author made in the editor.



Fig. 2. Mobile Mashup Editor



Fig. 3. iPhone



Fig. 4. Android

## 4 Conclusion and Future Work

The focus of the prototype implementation was to create the basis for an extensible software system capable of creating and deploying mobile mashups. Initial tests of our current prototype on the iPhone and Android platforms have proven the validity of this approach. The system offers new ways and possibilities for the creation of mobile applications that will be accessible from several mobile platforms. Moreover, the end-user programming environment, the *Mobile Mashup Editor*, provides an easy to use development platform, as an author can make use of visual programming concepts and techniques. Aspects related to privacy and trusts are beyond the scope of this paper.

**Innovative features and benefits:** The proposed solution provides the following unique features compared to existing approaches or systems: 1. The installed viewer application on the mobile device allows an easy access to the published mashups. Developers can therefore provide mobile mashup applications without the restrictions of an application store or licensing; 2. The use of cross-platform technologies on the editor and viewer site does not bind the solution to particular platforms and devices. As a result, the potential benefits of the proposed software system can be described as follows: 1. The deployed mobile applications have the layout and functionalities that were designed for and can use the features of the mobile device like GPS, camera and the device content; 2. The software system is component-based and therefore extensible through the integration of *additional components*. In our future development, we plan to integrate database components to our system. This will allow the *Mobile Mashup Viewer* to provide offline working capabilities and synchronisation with a remote database. Another line of exploration is the specification and development of *additional components* that can be used in different application domains.

## References

1. Chaisatien, P., Tokuda, T.: A Web-Based Mashup Tool for Information Integration and Delivery to Mobile Devices. In: Proceedings of the 9th International Conference on Web Engineering. LNCS, vol. 5648, pp. 489–492. Springer, Berlin (2009)
2. Bosch, J.: From software product lines to software ecosystems. In: Proceedings of the 13th International Software Product Line Conference, San Francisco, California, August 24–28, pp. 111–119. ACM Press, New York (2009)
3. Casteleyn, S., Florian, D., Dolog, P., Matera, M.: Engineering Web Applications: Data Centric Systems and Applications. Springer, Berlin (2009)
4. Sheth, A.: Citizen Sensing, Social Signals, and Enriching Human Experience. IEEE Internet Computing, 87–92 (2009)
5. Jensen, C.S., Vicente, C.R., Wind, R.: User-Generated Content: The Case for Mobile Services. IEEE Computer 41, 116–118 (2008)
6. Trevor, J.: Doing the mobile mashup, pp. 104–106. IEEE Computer Society, Los Alamitos (2008)