MobiMash: End User Development for Mobile Mashups

Cinzia Cappiello, Maristella Matera, Matteo Picozzi,
Alessandro Caio, Mariano Tomas Guevara
Politecnico di Milano, Dipartimento di Elettronica ed Informazione
via Ponzio, 34/5 - 20133, Milan, Italy
[surname@elet.polimi.it]

ABSTRACT

The adoption of adequate tools, oriented towards the End User Development (EUD), can promote mobile mashups as "democratic" tools, able to accommodate the long tail of users' specific needs. We introduce *MobiMash*, a novel approach and a platform for the construction of mobile mashups, characterized by a lightweight composition paradigm, mainly guided by the notion of *visual templates*. The composition paradigm generates an application schema that is based on a domain specific language addressing dimensions for data integration and service orchestration, and that guides at runtime the dynamic instantiation of the final mobile app.

Categories and Subject Descriptors

D2.2 [**Software Engineering**]: Design Tools and Techniques—*user interfaces, software libraries*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*GUI, interaction styles, prototyping, standardization*

General Terms

Human Factors, Languages

Keywords

Mobile Mashups, End-User Development, Model-based Mashups, Data Integration, Data Fusion

1. INTRODUCTION

With the diffusion of sophisticated mobile devices and the proliferation of available services, *Mobile Internet* is now a reality that empowers users to access services pervasively. As already initiated by the Web mashup paradigm, this new scenario further increases the desire of users to participate in the development of their own artifacts [1, 2]. The need to self-create applications is now even stronger for the mobile users: simple applications addressing very contingent and situational requirements, as mashups are, can solve several users' information needs arising in mobile usage contexts. This paper proposes *MobiMash*, a framework for the lightweight, user-driven composition and deployment of *mobile mashups*. MobiMash is characterized by an End-User Development (EUD) Web environment, where a visual composition paradigm, based on the completion of *visual templates*, allows the users to easily configure the fusion of contents coming from different data sources, and the synchronization of such core contents with both remote APIs and local services available on the mobile device. The so-created applications are device's native applications that, in contrast with Web mashups, do not need the Web browser as execution environment - the access to mobile device services is therefore enhanced.

2. APPROACH AND ARCHITECTURE

Figure 1 illustrates the overall organization of *MobiMash*. The platform enables the mash up of data services (e.g., REST services returning result sets in XML or JSON format), UI components, namely services equipped with their own user interface (UI) [1] (e.g., Google Maps, YouTube, and Flickr), and mobile device services. Given a pool of such services, a Web **Design Environment** (DE) allows users to compose mobile mashups through the completion of *visual templates*. Based on the user composition actions, DE generates and deploys on a server an XML-based application schema. The **Execution Engine** (EE), i.e., a native app running on the mobile device, then allows the users to download on the mobile device their application schema. Based on such schema EE automatically configures and instantiates the final application.

2.1 Visual composition

The right hand side of Figure 1 exemplifies the DE visual composition paradigm based on the completion of visual templates.The composition canvas consists of two main panels: the *data panel* on the left displays the data retrieved by querying some selected data services; the *visual template panel* on the right shows a selected visual template, i.e., a representation of the UI of the final app. As soon as data items selected in the data panel are associated to UI elements, the visual template panel is updated with a preview of the association effect. In this way, the user constructs the presentation layer - and implicitly integrates data. In fact, in case of data items deriving from different sources, the user actions are targeted towards the construction of unified data views. In this integration process, the local schemas of the different sources are reshaped depending on the user selection of data items, while the global schema is derived by the structure of the visual template and the user-defined mappings between its elements and the items of each single source. Therefore, in our approach visual templates play a double role: on the one side they offer an immediate and intuitive representation of the way data will be displayed

WWW 2012 Companion, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1230-1/12/04.

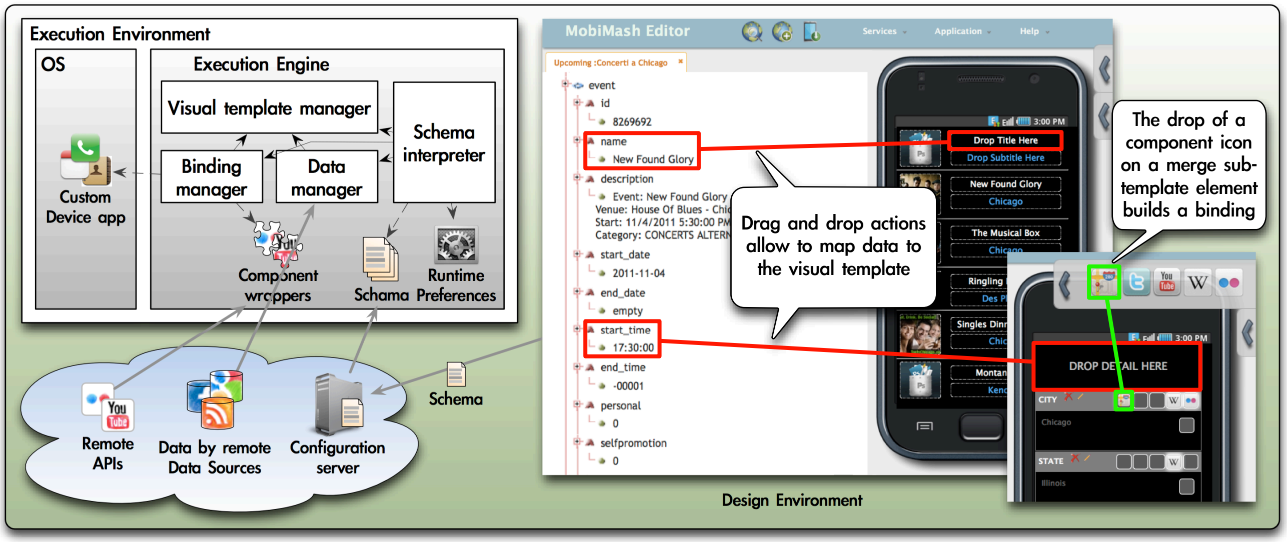

Figure 1: The overall organization of the MobiMash platform.

in the final application; on the other side, their visual completion with data fields extracted from multiple data sources enables an "integration-by-example" paradigm. The DE also allows the user to synchronize the integrated data view with the invocation of UI components and local services. The aim is to enrich the core data with contents of different nature. For example, data on concerts could be integrated with multimedia contents (videos and images) retrieved through APIs such as YouTube and Flickr. As shown in Figure 1, the available services are shown to the user through an icon menu in the upper part of the screen, which adapts its listed items by showing the only APIs and local services that are compatible with the data associated to the selected visual element.

2.2 Generation of the application schema

Based on the user visual actions for data integration and service binding definition, the DE automatically generates and uploads on a server a configuration file that contains rules for the automatic instantiation of the final mobile app. The schema reflects our model for mobile mashups, i.e., a domain specific language strongly characterized by the notion of visual template. The schema specifies:

- The *presentation layer*: the set of *visual element types* that will appear in the final UI, as deriving by the visual template completed during the mashup design.
- The *service data layer*: for each visual element, the data services providing the associate data, the corresponding queries and other properties (such as the definition of data filters) defined during the data service registration.
- The *service binding layer*: for each visual element, the list of local and remote services subscribed to the selection of the associated data items.

2.3 Mobile mashup execution

The left-hand side of Figure 1 illustrates the main elements composing the Execution Engine running on the mobile device. A *schema interpreter* parses the application schema and masters the instantiation of the application by invoking the other modules. The *visual template manager* translates the user-defined visual template into device native code for the UI layout generation, and also populates the visual elements any time new data are retrieved. Based on the data mapping rules defined for the visual elements, the *data manager* queries the involved services, and executes algorithms for data fusion in case of data coming from multiple sources. A *binding manager* is instead in charge of "listening" to the selection of *publisher* data items, capturing parameters embedded in such data items, and invoking, by means of the API wrappers, the synchronized service operations.

3. CONCLUSIONS

The MobiMash platform demonstrates how adequate abstractions, mainly based on visual template construction, can enable a lightweight development of mobile mashups. Some performance experiments have shown that our data integration approach is feasible even under the limited capabilities of the mobile device. A user study has also validated the effectiveness and efficiency of our mashup approach with respect to varying end-users abilities. A MobiMash demo is available at http://youtu.be/0spXKby_17E.

4. REFERENCES

[1] C. Cappiello, M. Matera, M. Picozzi, G. Sprega, D. Barbagallo, and C. Francalanci. Dashmash: A mashup environment for end user development. In *Proc. of Web Engineering - 11th International Conference, ICWE 2011, Paphos, Cyprus, June 20-24, 2011*, volume 6757 of *LNCS*, pages 152–166. Springer, 2011.

[2] F. Daniel, M. Matera, and M. Weiss. Next in mashup development: User-created apps on the web. *IT Professional*, 13(5):22–29, 2011.