Hands-on actionable mashups

Carmelo Ardito[1], Maria Francesca Costabile[1], Giuseppe Desolda[1], Markus Latzina[2], Maristella Matera[3]

[1]Dipartimento di Informatica, Università degli Studi di Bari Aldo Moro
Via Orabona, 4 – 70125 – Bari, Italy
{carmelo.ardito, maria.costabile, giuseppe.desolda}@uniba.it
[2]Strategic Projects, Products & Innovation – Technology, SAP SE
Dietmar-Hopp-Allee 16 – 69190 – Walldorf, Germany
markus.latzina@sap.com
[3]Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano
Piazza Leonardo da Vinci, 32 – 20134 – Milano, Italy
maristella.matera@polimi.it

Abstract. This paper describes how to involve end users without expertise in programming in a session where they will be asked to accomplish some tasks according to a new paradigm for actionable mashups. The goal will be to understand what the advantages of this new paradigm are with respect to traditional methods for mashup composition and information exploration.

1 Introduction

In several application domains there is an increasing demand by end users (simply called "users" in the rest of this paper) to effectively access, integrate, and visualize multiple resources available online. In this respect, platforms for service integration, and especially End-User Development (EUD) paradigms for mashup composition, play an important role as they let users integrate heterogeneous information that otherwise would be totally unrelated [1]. Web mashups are "composite" applications constructed by integrating ready-to-use heterogeneous resources exposed by public or private Web services and APIs [2]. They offer in particular the possibility to integrate such resources at the presentation layer, an aspect that enables the creation of full-fledged applications whose user interface (UI) is easily achieved by synchronizing the UIs of the different components. Several works in the last year have been proposing platforms for mashup composition. Some of them offer novel paradigms to allow users to construct interactive and pervasive information spaces [2]. One such tool is proposed in [1]. Let us briefly describe how a user creates mashups with this tool.

Maria, a teenager keen on rock music, interacts with a web application on a PC to retrieve and explore various information about musical events by means of mashups.

Maria can browse information provided by different online services and classified by various categories (e.g., video, photo, music, social). Maria decides to access the information through SongKick, a service registered into the platform that provides information about music events. From a number of UI templates provided by the system, Maria chooses the map (in particular, using the Google Maps service) for displaying the retrieved music events in the form of pins on the map. She also decides to visualize some further details on each single event, this time using a table view. By selecting a pin on the map, the details of the corresponding event get visualized accordingly. **Fig. 1** depicts an example of the created mashup, rendered within a web browser. By typing "Afterhours" in the search box, Maria retrieves the forthcoming events of this rock band, which are visualized as red pins on the map. Maria clicks on one pin and a table appears showing the details of that event.
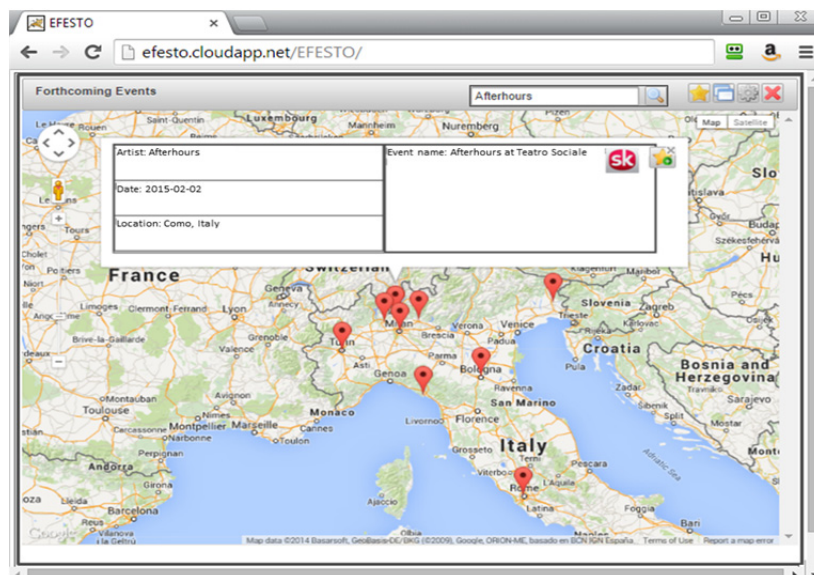

Fig. 1. Example of mashup created with the platform in [1] for monitoring rock concerts

The example illustrated above shows that, with the platform described in [1], the user performs the mashup without writing any code, but only using visual interaction mechanisms. However, Maria could not accomplish much more than visualizing a list of data, modifying the way data are visualized, or inspecting data details. Additional features would be needed in order to make the mashup information *actionable*, which means that Maria can act on the information resulting from the mashup, manipulate and even transform it according to her specific task. The proposal to achieve actionable mashups is to adopt TUX (Transformative User Experience) principles [3]. As discussed in [4], TUX indeed provides a framework for designing such task-oriented features. Therefore, besides adopting exclusively UI templates as containers supporting the access and the visualization of the information provided by services (as described in [1]), through TUX the mashup platform can exploit the notion of *task con-*

tainers, i.e., extensions to UI templates in charge of providing task semantics. In other words, in a task container information items gain the status of task objects to which task functions, specific to this container, can be applied [5]. In the next section, we describe how participants to the hands-on activity, impersonating Maria, will be required to interact with a mashup platform integrated with TUX principles; the goal of this activity will be understanding whether by means of the TUX extensions users perceive a greater utility of the mashup data with respect to the possibility to fulfil some goals which go beyond the mere retrieval of information.

2   Hands-on activity

People participating to the hands-on activity will be end users without specific expertise in programming. The interaction with the platform is individual. Each participant is briefly introduced to the scenario: he or she acts as Maria, who wants to attend a musical event with her friends. She uses the platform to search for forthcoming music events. She also gathers information that can inform the discussion with her friends about which event to attend. In the following paragraph, we describe in more details the tasks executed by Maria, i.e., by each participant.

By means of a PC, Maria logs in the web platform that offers a workspace where she can retrieve information by mashing up services and act on the information through specific functions provided by task containers. The workspace has been previously customized, to provide the tools required to carry out the hands-on activity. The platform is equipped with services providing data on music events, plus some other services of generic utility, e.g., map services. To enlarge the set of available data to be integrated in the mashup, a polymorphic data source, exploiting the information structured in the Linked Open Data cloud, is provided [6]. The workspace also offers a collection of task containers. Each container is represented as a box widget with a labelled icon that indicates its intended purpose by highlighting a primary task function, e.g., a World globe for browsing, two side-by-side paper sheets for comparing, a call-out for communicating. When needed, a container representation can be moved by the user from this collection into the main area of the workspace, in order to activate its full functional scope.

(Step 1): Maria selects the task container "Events" and chooses "music" as event type. A map is displayed: every music event is represented as a pin at specific coordinates. The details of each event can be inspected through the corresponding pin. *(Step 2)*: Maria includes the "Selecting" container where she makes a pre-selection by dragging from the "Events" container those events she is more interested in. She further refines her selection by means of a "Comparing" container, which offers features supporting the comparative inspection of items. After this analysis, Maria chooses the three most promising events and removes the others from the "Selecting" container. *(Step 3)*: Maria drags the "Housing" container in the main area of the workspace, in particular touching the "Selecting" container. In this way, she synchronizes the two containers. Three lists of hotels, one for each different event place, are visualized. For each hotel it is displayed a thumbnail photo, name, price, guests' rating. Maria per-

forms those actions usually allowed by hotel booking web sites, i.e., changing dates, ordering, filtering, inspecting details. She selects a couple of hotels for each location. On the basis of the housing information, she reduces the candidate events to only two and eliminates the third from the "Selecting" container. *(Step 4)*: Maria wants to send an email with a summary of the information related to the two chosen events. Thanks to the "Communicating" container, she is not forced to use an email client external to the workspace. She drops the items from the "Selecting" to the "Communicating" container, where she selects the recipients and the communication channel, e.g. a post on a social network, an email, etc. She decides to send an email. The email addresses of her friends are displayed and the email body is prefilled automatically with the information about the events and the hotels. The message can be edited by Maria before sending. It is noteworthy to remark that Maria is not constrained to a predefined flow: for example, she could directly move events from "Selecting" to "Communicating", thus deliberately skipping the "Comparing" or the "Housing" container.

3   Objectives and learning goals for the activity

The proposed activity allows users to interact with a system that enables them to express and respond to their task needs rather directly and dynamically. By observing the end users while using this system we aim to assess the validity of our ideas on the integration of mashups and TUX principles, and to verify whether making mashups actionable actually provides an added value with respect to the users' needs and expectations. We also aim to propose new EUD paradigms that can empower users to shape up software environments that can really support their situational needs.

References

1. Ardito, C., Costabile, M.F., Desolda, G., Lanzilotti, R., Matera, M., Piccinno, A., Picozzi, M.: User-Driven Visual Composition of Service-Based Interactive Spaces. Journal of Visual Languages & Computing 25(4), pp. 278-296 (2014)
2. Daniel, F., Matera, M.: Mashups - Concepts, Models and Architectures. Springer (2014)
3. Latzina, M., Beringer, J.: Transformative user experience: beyond packaged design. Interactions 19(2), pp. 30-33 (2012)
4. Ardito, C., Costabile, M.F., Desolda, G., Latzina, M., Matera, M.: Making mashups actionable through elastic design principles. In this volume
5. Beringer, J., Latzina, M.: Elastic Workplace Design. In: Wulf, V., et al. (eds.), Designing Socially Embedded Technologies in the Real-World. Springer, London (2015)
6. Desolda, G.: Enhancing Workspace Composition by Exploiting Linked Open Data as a Polymorphic Data Source. In: IIMSS '15, June 17-19, Sorrento, Italy. (2015)