Mashups: A Journey from Concepts and Models to the Quality of Applications

Cinzia Cappiello[1], Florian Daniel[2], and Maristella Matera[1]

[1] Politecnico di Milano
Via Ponzio 34/5, I-20133 Milano, Italy
{cinzia.cappiello,maristella.matera}@polimi.it
[2] University of Trento,
Via Sommarive 9, I-38123, Povo (TN), Italy
daniel@disi.unit.it

Abstract. This tutorial aims to provide insight into the constantly evolving mashup ecosystem. It presents core *definitions*, overviews a representative set of *mashup components* (the resources to be integrated) and *mashup models* (how resources are integrated), illustrates *composition paradigms* supported by state-of-the-art mashup tools, and discusses the *quality* of the resulting mashups from a user perspective. The goal of the tutorial is to introduce the topic and show its applicability, benefits and limitations.

1 Context and Motivation

The term "mashup" is widely used today. There are people developing "mobile mashups," others doing research on "Web mashups," and others again selling tools for "data mashups." Yet, when it comes to a concrete discussion of the topic, it is not uncommon to discover that the parties involved in the discussion actually have very different interpretations of what mashups are and what they are not. Typical discussion points are whether a mashup must have a user interface (UI) or not to be called a "mashup", whether it must be built by using Web-accessible resources only or not, whether it must be developed with client-side technologies (e.g., JavaScript) only, and the like. That is, even after several years that the term has been around and used, there is still no common agreement on its actual meaning and implications.

Interestingly, however, in the meantime mashing up data, functionalities and user interface widgets sourced from the Web has inexorably percolated into Web Engineering as a tacitly accepted development practice. Today, it is unimaginable to develop modern Web applications without some form of reuse and integration of value-adding, third-party content or services, a task that is greatly facilitated by technologies like Web services [3], the RESTful architectural style [2], Open Data, XML, JSON, W3C widgets, and many more.

But which are the conceptual underpinnings of this practice? What does it exactly mean to "mash up" resources that can be accessed via the Web? Which are the paradigms adopted for the composition of mashups? What kinds of tools

exist that support this activity? And, eventually, what does it mean to develop "good" mashups? Working with students, discussing with colleagues, reading publications on the topic, we have seen that these questions are still open to many. We also identified a lack of suitable study material.

2 Learning Objectives

In light of these considerations, the learning objectives of this tutorial are:

- To obtain a basic understanding of the *core mashup aspects and concepts*, such as their contexts of use and target users, the most important definitions of mashups depending on the considered point of view (e.g., Web mashups, enterprise mashups, process mashups, telco mashups, mobile mashups, etc.), their intrinsic complexity and benefits.
- To get insight into the *most representative component technologies* used by mashups. Components are the basic elements of a mashup, and the comprehension of their characteristics and capability is fundamental for the understanding of what a mashup is and how it can be developed.
- To understand the *conceptual underpinning of mashups*, which developers must master and that can help them focus on the relevant issues when integrating components into mashups, as well as reference architectural patterns that can be instantiated. Both concepts and architectures can be analyzed independently of the particular technologies or sources used for an actual implementation.
- To gain insight into how mashup models and development practices can materialize into dedicated *mashup tools and composition paradigms* for assisted mashup development.
- To get insight into *quality models* for both mashup components and mashups, to understand how such models can guide the initial choice of components and the successive selection of composition patterns, and how they can also augment composition paradigms through the generation of quality-based recommendations.

The *target audience* are researchers, practitioners, advanced students who want to learn more about mashup development from a perspective that especially privileges abstractions and models, not only implementation aspects.

The tutorial is based on the authors' latest publication on mashups [1] and is complemented with an *online resource* providing additional material, slides and links for further study: `http://www.floriandaniel.it/mashupsbook`.

References

1. F. Daniel and M. Matera. *Mashups: Concepts, Models and Architectures.* Springer, 2014.
2. R. Fielding. *Architectural Styles and the Design of Network-based Software Architectures.* Ph.D. Dissertation, University of California, Irvine, 2007.
3. M. P. Papazoglou. *Web Services - Principles and Technology.* Prentice Hall, 2008.