Hosted Universal Integration on the Web: the mashArt Platform

Florian Daniel[1], Fabio Casati[1], Stefano Soi[1], Jonny Fox[1], David Zancarli[1],
Ming-Chien Shan[2]

[1] University of Trento, Italy
{daniel,casati,soi,fox,zancarli}@disi.unitn.it

[2] SAP Labs- 3410 Hillview Avenue, Palo Alto, CA 94304, USA
ming-chien.shan@sap.com

Abstract. Traditional integration practices like Enterprise Application Integration and Enterprise Information Integration approaches typically focus on the application layer and the data layer in software systems, i.e., on limited and specific development aspects. Current web mashup practices, instead, show that there is also a concrete need for (i) integration at the presentation layer and (ii) integration approaches that conciliate all the three layers together. In this demonstration, we show how our *mashArt* approach addresses these challenges and provides skilled web users with *universal integration* in a hosted fashion.

Keywords: Hosted Universal Integration, Mashups, Services Composition.

1 Introduction and contributions

Mashups are online applications that are developed by composing contents and functions accessible over the Web [1]. The innovative aspect of mashups is that they also tackle integration at the user interface (UI) level, i.e., besides application logic and data, they also reuse existing UIs (e.g., many of today's applications include a Google Map). We call this practice of integrating data, application logic, and UIs for the development of a composite application *universal integration*.

Universal integration can be done (and is being done) today by joining the capabilities of multiple programming languages and techniques, but it requires significant efforts and professional programmers. There is, however, also a growing number of *mashup tools*, which aim at aiding mashup development and at simplicity more than robustness or completeness of features. For instance, Yahoo Pipes focuses on RSS/Atom feeds, Microsoft Popfly on feeds and JavaScript components, Intel Mash Maker on UIs and annotated data in web pages, while JackBe Presto also allows putting a UI on top of data pipes. None of these, however, covers the three application layers discussed above together in a convenient and homogeneous fashion.

Building on research in SOA and capturing the trends of Web 2.0 and mashups, in this demo we propose an integrated and comprehensive approach for universal integration, equipped with a proper hosted development and execution platform called *mashArt* (a significant evolution of the work described in [2]). Our aim is to do what service composition has done for integrating services, but to do so at all layers, not just at the application layer, and to do so by learning lessons and capturing the trends of Web 2.0 and mashups, removing some of the limitations that constrained a wider adoption of workflow/service composition technologies.

The mashArt approach aims at empowering non-professional programmers with easy-to-use and flexible abstractions and techniques to create and manage composite web applications. Specifically, mashArt provides the following, unique contributions:

- A *unified component model* that is able to accommodate and abstract UI components (HTML), application logic components (SOAP or RESTful services), and data components (feeds or XML/relational data) using a unified model.
- A *universal composition model* that allows mashArt users to develop composite applications on top of the unified component model and conciliates the needs of both UI synchronization and service orchestration under one hood.
- A *development and execution platform* for composite applications that facilitates rapid development, testing, and maintenance. mashArt is entirely hosted and web-based, with zero client-side code.

2 Demonstration storyboard

The live demonstration introduces the three contributions of mashArt by means of a joint use of slides (for the conceptual aspects) and hands-on platform demos (for the practical aspects). In particular, the demonstration is organized as follows:

1. *Intro:* introduction of the conceptual and theoretical background of the project, its goals and ambitions, and its contributions.
2. *UI integration*: explanation of the idea of UI integration and how UI components and the composition logic look like.
3. *UI integration demo*: demonstration of how to do UI integration with mashArt starting from a set of existing mashArt UI components. Two minutes suffice to show how to develop and run a simple application that synchronizes a search component and a map component for geo-visualization of results.
4. *Universal integration*: description of mashArt's component model and its composition model, which characterize the universal integration approach.
5. *Universal integration demo*: demonstration of how to combine service, data, and UI integration in mashArt. Again, two minutes suffice to show how to add an RSS reader component to the previous scenario and to feed it with data sourced from a RESTful service and transformed via a Yahoo! pipe.
6. *Architecture*: functional architecture of mashArt to show that mashArt is (or will be) more than what is shown in the demo.
7. *Conclusion and future works*: summary and outline of future works.

A short version of the demo can be previewed here: *http://mashart.org/mashArt.wmv*.

References

[1] J. Yu, et al., "Understanding Mashup Development and its Differences with Traditional Integration," *Internet Computing*, vol. 12, no. 5, pp. 44-52.

[2] J. Yu, et al., "A Framework for Rapid Integration of Presentation Components," in *WWW'07*, 2007, pp. 923-932.