Chapter 16
Integrating Heterogeneous Services for Semantic Mashup Construction

Khayra Bencherif
Djilali Liabes University, Algeria

Djamel Amar Bensaber
High School of Computer Science of Sidi Bel Abbes, Algeria

Mimoun Malki
High School of Computer Science of Sidi Bel Abbes, Algeria

ABSTRACT

Semantic mashup applications allow automating the process of services and data integration to create a composite application with a new user interface. Nevertheless, existing mashup applications need to improve the matching methods for discovering semantic services. Moreover, they have to create or modify workflows in mashup applications without the assistance of the original developers. Automating the combination of user interfaces is another challenge in the context of semantic mashups construction. In this chapter, the authors propose an approach that allows automating the combination of data, services, and user interfaces to provide a composite application with an enhanced user interface. The construction of the semantic mashup application is based on the use of domain ontology, a matching tool, and a collection of patterns. In order to demonstrate the effectiveness of this proposal, the authors present a use case to construct a semantic mashup application for a travel agency.

INTRODUCTION

With the coming of the Web 2.0, the user can show his existence through use of many technologies like social networking, collaborative environments, mashups, etc. (Lytras, Damiani, & Pablos, 2008). A mashup application allows integrating user interfaces, data and services from various sources to build a single Web application (Khokhar, Benjamin, Farkhund, Alhadidi, & Bentahar, 2016). Generally, the

DOI: 10.4018/978-1-5225-5384-7.ch016

mashups take the form of a Webpage or a Website that manually combines heterogeneous information and services from multiple sources (Garriga, Mateos, Flores, Cechich, & Zunino, 2016). However, the development of mashup applications allows the developers to understand the structure and the semantics of APIs and data, which make the integration process more difficult. Thus, the semantic mashups are developed to facilitate automatic integration of data and services using a semantic layer. On the other hand, patterns are software architectures that help the designers to build solution blocks which have been shown to present useful information in the past and will provide more efficient solutions in the future (Liu, Liang, Xu, Staples, & Zhu, 2011).

Several approaches have dealt with automatic integration of data and APIs in mashup applications (Kopecký, Vitvar, Bournez, & Farrell, 2007; Lathem, Gomadam, & Sheth, 2007; Maleshkova, Pedrinaci, & Domingue, 2009; Ngu, Carlson, Sheng, & Paik, 2010; Liu, Li, Pedrinaci, Kopecký, Maleshkova, & Domingue, 2011; Meditskos & Bassiliades, 2011; Malki & Benslimane, 2012; Malki & Benslimane, 2013; Lee, 2014; Tjoa, Wetz, Kiesling, Trinh, & Do, 2015; Park, Yoo, Hur, Bae, & Lung, 2015; Lee, 2015; Trinh, Do, Wetz, Aryan, Kiesling, & Tjoa, 2017; Karakostas & Kalamboukis, 2017).

Key challenges of these approaches are the need to 1) create or modify workflows in mashup applications without enlisting the talents of the original developers or vendor, 2) compute semantic and syntactic similarities between data in different services, and 3) automate the combination of user interfaces to construct an enhanced composite application.

In order to address these challenges and enhance the quality of data and services integration in mashup applications, this chapter puts forward a novel approach which consists in building a patterns-based semantic mashup for the enterprise 2.0. This approach is based on the use of domain ontology and a matching tool to identify similar information and services in different sites. In addition, the authors use a set of patterns to facilitate construction and use of the semantic mashup application. First, the pattern 'Content Integration' is used to integrate data after the matching process. Then, the pattern 'Usability Enhancer' is used to present the application in a single interface instead of wasting time to switch between different applications. Finally, the pattern 'Workflow' is used to manage the chaining of the spots.

The authors evaluate the system prototype with a study use case to develop a mashup application for a travel agency. It provides a single widget that allows the user to specify starting location, destination and favorite transportation (flight or train). After submission of data, the system retrieves the availability of the flight or train as well as hotels and car rental options using domain ontology and a matching tool.

Consequently, the main contributions of this chapter are as follows:

1. A review of existing approaches and tools that are developed to build semantic mashup applications. The authors provide also a comparative study of these approaches to outline their limitations.
2. Development of domain ontology and a matching tool to resolve the problem of data heterogeneity.
3. Use of a collection of patterns to facilitate the construction of a composite application and provide an enhanced interface to the end user.
4. Presentation of a use case of a semantic mashup application for a travel agency.

Rest of this chapter is organized as follows. Definitions and detailed descriptions of Web 2.0, mashups, process of computing data mashups, semantic mashups and mashups patterns types will be introduced in the next section. Then an overview of existing semantic mashup approaches as well as a comparison