A Personalized Mashup Using Rule-based Reasoning and Linked Data

Aikaterini K. Kalou, Georgia D. Solomou, Dimitrios A. Koutsomitropoulos and Theodore S. Papatheodorou

HPCLab, Computer Engineering and Informatics Dpt., University of Patras, Building B, 26500, Patras-Rio, Greece
{kaloukat, solomou, kotsomit, tsp}@hpclab.ceid.upatras.gr

Abstract. In this paper we propose an intelligent personalization service, built upon the idea of combining Linked Data with Semantic Web rules. This service is mashing up information from different bookstores, and suggests users with personalized data according to their preferences. This information as well as the personalization rules are then processed and managed by a scalable knowledge repository. Finally, they are made available as Linked Data, thus enabling third-party recipients to consume knowledge-enhanced information.

Keywords: Linked Data, ontologies, Semantic Web rules, triple stores, mashups.

1 Introduction

The great proliferation of Linked Data paves the way for the deployment of robust applications, able to consume large datasets in a more effective way [2]. However, their limited semantics often leads to applications with poor knowledge discovery capabilities [5, 6]. On the other hand, the use of full-fledged ontologies comes with an additional overhead, especially when inferencing is required.

In an attempt to achieve a trade-off between powerful reasoning and scalability, we propose a web application that combines the basic principles of Semantic Web rules and Web 2.0 mashups [4], aiming to provide an intelligent and scalable service for personalized querying over popular on-line bookstores. Our application ('Books@HPCLab') provides users with the ability to search and find sell-offers for books that fit their preferences. These data are mashed up from different and heterogeneous sources on the Web, like Amazon and Half Ebay. The descriptive metadata of retrieved books are finally triplified, thus producing a linking connection to their offers. All collected information is stored in a scalable rule-based triple store (OWLIM [1]).

Gathered information is mapped to an OWL ontology that has been developed in order to describe users, books and offers (the 'BookShop' ontology). The BookShop ontology is kept in the triple store together with a number of rules, which reflect user preferences. This ontology- and rule- oriented approach renders the gathered

information reusable and sharable. At the same time, we expose and make it available to the Linked and Open Data (LOD) world, through an appropriate interface.

Books@HPCLab application is a significantly enhanced version of a previous one presented at [7]. This new application actually serves as a proof of concept about the successful and efficient combination of Semantic Web, Linked Data and rules in designing intelligent mashups.

2 Design and Architecture

The full functionality of Books@HPCLab application is carried out by several distinct components that interact with a central management mechanism, the '*Core Unit*' (see Fig. 1). The Core Unit implements the application logic of the service and is responsible for communicating with the '*Mashup Component*', the '*Triple Store*', the '*User Interface (UI)*' and finally with the '*Linked Data Component'*, which is responsible for the proper exposure of retrieved information as Linked Data. The implemented architecture is actually based on a REST-ful communication strategy, where data transfer is implemented via HTTP requests.

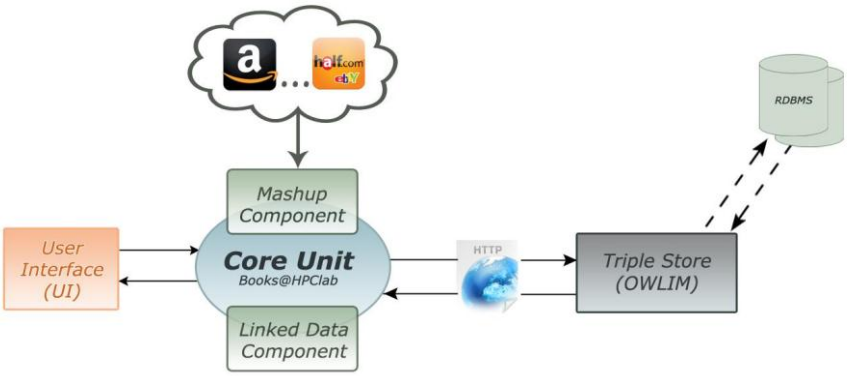

Fig. 1. Architecture overview.

The UI component is the application front-end that provides users with all necessary facilities for mashing information that is in accordance to their preferences. By interacting with the UI, a user can create his profile, submit a search request and finally view and access appropriate results ranked by the number of matching preferences (Fig. 2). The user's request, consisting of search keywords, is passed on to the Core Unit which in turn assigns it to the corresponding components.

The Mashup component carries out the communication with Amazon and Half Ebay services, by interacting with their respective Web APIs. Whenever the user sends a *searching call*, the searching process starts to query data from Amazon Web Services (AWS).Then, for each book returned by Amazon, additional offers are found, that may be available at Half Ebay. The application uses the RDF API for PHP [3] in order to process aggregated data, to assign resolvable identifiers and ultimately to convert them to Linked Data in RDF format. This conversion happens in both

directions, i.e. when the data are collected as well as when a user requests RDF data in the book view page (Fig. 3). In the latter case however, information is already available in the triple store and fetched directly, with only a minimal intervention to tide them up.

Fig. 2. Presentation of preferred books for User_16.

Fig. 3. Linking data about Books and Offers.

Harvested data are matched against the BookShop ontology and get stored at the triple store. The triple store keeps the ontology schema and a set of custom rules. These "personalization rules" actually act as a filter to the search results, being able to distinguish among those books that satisfy user's preferences and those that are irrelevant to the user's profile. The population of data as well as the reasoning process is performed on loading. The stored rules are fired immediately and produce a number

of inferred statements. A couple of such rules are shown in Table 1, which check for example whether a book matches the user's preferred book condition (e.g. "new" or "used") and maximum book price respectively. Results (explicit and inferred) become available as RDF triples and can be accessed via HTTP, by making SPARQL queries.

Table 1. Example rules for matching users' preferences.

No	Rule Body	Rule Head
1	IF ***u*** *prefersCondition* ***z*** AND ***b*** *hasOffer* ***o*** AND ***o*** *hasBookCondition* ***z***	***u*** *prefersBookbyCondition* ***b***
2	IF ***u*** *prefersMaxPrice* ***z*** AND ***b*** *hasOffer* ***o*** AND ***o*** *offerPrice* ***z***	***u*** *prefersBookbyPrice* ***b***

References

1. Bishop, B., Kiryakov A., Ognyanoff, D., Peikov, I., Tashev, Z., Velkov, R.: OWLIM: A Family of Scalable Semantic Repositories. Semantic Web Journal. 2(1), pp. 33--42 (2011)
2. Bizer, C., Heath T, Berners-Lee, T.: Linked Data - The Story So Far. International Journal on Semantic Web and Information Systems. 5(3), pp 1--22. (2009)
3. Cowles, P.: Exposing Web Applications Semantically Using RAP (RDF API for PHP). PHP Architect. 3(10), pp 34--39. (2004)
4. Crupi, J., Warner, C.: Enterprise Mashups: The New Face of Your SOA. SOA World Magazine, http://soa.sys-con.com/node/719917 (2009)
5. Hogan, A.: Integrating Linked Data though RDFS and OWL: Some Lessons Learnt. In: 5th International Conference on Web Reasoning and Rule Systems, LNCS. (to appear)
6. Hogan, A., Pan, J., Polleres, A., Yuan, R.: Scalable OWL 2 Reasoning for Linked Data. In: Reasoning Web. Semantic Technologies for the Web of Data. LNCS vol. 6848, Springer (2011)
7. Kalou, A., Pomonis, T., Koutsomitropoulos, D., Papatheodorou, T.: Intelligent Book Mashup: Using Semantic Web Ontologies and Rules for User Personalisation. In: 4th IEEE Int. Conference on Semantic Computing, pp. 536--541. (2010)