Mashing up user data in the Grapple User Modeling Framework

Fabian Abel¹, Dominikus Heckmann², Elco Herder¹, Jan Hidders³, Geert-Jan Houben¹, Daniel Krause¹, Erwin Leonard³, Kees van der Sluijs⁴

¹ L3S Research Center, Hannover, Germany, {abel,herder,krause}@L3S.de
² DFKI GmbH, Saarbrücken, Germany, heckmann@dfki.de
³ WIS, TU Delft, The Netherlands{a.j.h.hiders,g.j.p.m.houben,e.leonardi}@tudelft.nl
⁴ CS Department, Eindhoven University of Technology, The Netherlands, k.a.m.sluijs@tue.nl

Abstract

In this paper we demonstrate the Grapple User Modeling Framework (GUMF), which exploits Semantic Web technologies and Web 2.0 paradigms to model users across different applications and domains. It introduces novel features such as dataspaces, which logically bundle user data, and user pipes, which allow to mash up user data from different sources.

1 Introduction

Web systems such as Amazon or YouTube brought personalization to the general public. While those popular systems base recommendations on a large amount of data - by means of collaborative filtering and social network analysis - [Frias-Martinez et al., 2005], the majority of Web applications cannot build upon a big user population and users might not interact regularly with these systems. A promising approach to compensate for this lack of data is cross-application user modeling [Korth and Plumbaum, 2007]. In contrast to the centralized approach of generic user modeling servers [Abel et al., 2008], a conversion-based approach allows for flexible mappings. These mappings can be created from one system to another, or by making use of generalized representations, such as the General User Model Ontology (GUMO) [Heckmann et al., 2005] and UserRDF [Abel et al., 2008].

In this paper we present the architecture and implementation of the Grapple User Modeling Framework (GUMF) [Abel et al., 2009a], which re-uses, refines and enhances previous work in the area of cross-application and generic user modeling systems. GUMF organizes user profile data in dataspaces, which constitute views on a specific set of data. Dataspaces are extensible with plug-ins for mapping and integrating data from external data sources; these plug-ins can also be used to reason with existing to deduce further knowledge about the user. In addition to traditional rule-based approaches, GUMF provides so-called user pipes [Abel et al., 2009a] that mash up different (user profile) data streams, formatted in RDF or RSS, making use of Semantic Web Pipes² or Yahoo Pipes³.

2 GUMF: Grapple User Modeling Framework

Figure 1 shows the architecture of GUMF. The elements at the top provide the essential, generic functionality of the framework; elements part at the bottom right provide generic as well as domain-specific reasoning logic.

Client applications can access GUMF either via a RESTful or SOAP-based API. Further, there is a Java Client API that facilitates development of GUMF client applications. Client applications mainly approach GUMF to store user information (handled by the Store Module) or to query for information (handled by Query Engine). User profile information is modeled by Grapple statements [Abel et al., 2009b], which are basically reified RDF statements about a user, enriched with provenance metadata. GUMF currently supports SPARQL and SeRQL queries as well as a pattern-based query language that exploits the Grapple statement structure to specify what kind of statements should be returned by GUMF. Authorized client requests are answered by GUMF’s Dataspace Logic. Dataspaces are equipped with data storage repositories that either reside at the GUMF server or are distributed across the Web (possibly maintained by the client application itself), and with (reasoning) plug-ins that further enrich the data that is available in the repositories.

The Administrator of a GUMF client application can configure dataspaces and plug-ins via the GUMF Admin Interface. Activating or deactivating plug-ins directly in the repositories changes the behavior of dataspaces. Further, administrators can adjust the plug-ins and reasoning rules to their needs. For example, we developed a plug-in that gathers user profile information from Facebook and maps—with support of

---

¹Currently available at: http://semweb.kbs.uni-hannover.de:8082/grapple-umf/
²http://pipes.deri.org
³http://pipes.yahoo.com
Silk⁴—the profile to a format preferred by the client application administrator (e.g., FOAF⁵ or OpenSocial⁶).

Inspired by Web 2.0 practices, a key principle of GUMF is that dataspaces can be shared across different client applications. Therefore, clients can subscribe to other dataspaces, given that they are granted approval by the administrator of the dataspace. When subscribed to a dataspace, the client is allowed to query it. However, it might still not be allowed to access all statements that are made available via the dataspace, as fine-grained access control functionality can be embedded in the dataspaces as well.

3 Demonstration Overview

In our demonstration we primarily show how client applications can benefit from the Grapple User Modeling Framework.

Developers of client applications first have to register their application at GUMF. Upon registration, a dataspace is generated that can immediately be used to store user profile information. As an example, a client might store user interests such as “Peter is interested in Darmstadt” in GUMF, by using the RESTful API and the Java Client implementation. GUMF models such information as Grapple statement.

Figure 2 shows the administration interface of GUMF, in particular the configuration of the dataspaces. Administrators can add plug-ins to a dataspace (cf. “add plug-in”) and adjust which client applications are allowed to access the dataspace (cf. “Subscriptions”). Via GUMF’s RESTful API, client applications send advanced SPARQL queries or queries based on simple patterns. As an example, .../ds/13/predicate/interest would return all Grapple statements in the dataspace .../ds/13 on user interests. The output format of a query can be selected as well. At the moment, GUMF supports RDF/XML, RSS 2.0 and SPARQL Query Results XML format.

In our demonstration at ABIS, we will show how GUMF is applied to mash up and reason with user profile information from different tagging and social networking systems (Flickr, Facebook, TagMe⁷, and GroupMe⁸).

---

References


