A Linked Data Perspective for Effective Exploration of Web APIs Repositories

Devis Bianchini, Valeria De Antonellis, and Michele Melchiori

Dept. of Information Engineering University of Brescia
Via Branze, 38 - 25123 Brescia (Italy)
{bianchin,deantone,melchior}@ing.unibs.it

Abstract. In this paper, we propose a novel approach to provide a comprehensive cross-repositories view of the available Web APIs information, in order to enhance effective multi-perspective Web APIs search for fast development of web mashups. The approach is based on Linked Data principles to identify and use semantic links across repositories for search purposes. Specifically, the paper considers Web APIs search across the popular ProgrammableWeb and Mashape repositories by combining their distinctive Web API descriptions.

1 Approach Overview

The problem of searching Web APIs to be aggregated for fast web mashup development necessarily comes up against the fact that Web APIs are shared by providers across different public repositories, which focus on distinct aspects that are considered for Web API search. In this paper, we propose a novel approach to provide a comprehensive cross-repositories view of the available Web APIs information, in order to enhance effective multi-perspective Web APIs search. Specifically, we consider Web APIs search on the `ProgrammableWeb`[1] (PW) and `Mashape`[2] (MP) repositories by combining their distinctive Web API descriptions. With respect to related approaches on Linked Web services and Linked Web APIs [1,2], we rely on information stored within public available repositories without forcing the web designers to perform semantic annotation of Web APIs.

An overview of our approach is shown in Figure 1. The approach is based on Linked Data principles: (i) the contents of repositories are formally represented (based on RDF), to make them machine processable and enable access through non-proprietary tools (e.g., SPARQL endpoints); (ii) metrics and criteria to automatically identify semantic links between RDF elements (e.g., Web APIs, developers' profile) across different vocabularies are defined; (iii) identified links are also published as open data to be properly exploited for Web API search. Starting from the PW and MP repositories, RDF vocabularies which represent their contents are designed using the main concepts in this domain of

[1] `http://www.programmableweb.com/`
[2] `https://www.mashape.com/`

F. Daniel, P. Dolog, and Q. Li (Eds.): ICWE 2013, LNCS 7977, pp. 506–509, 2013.

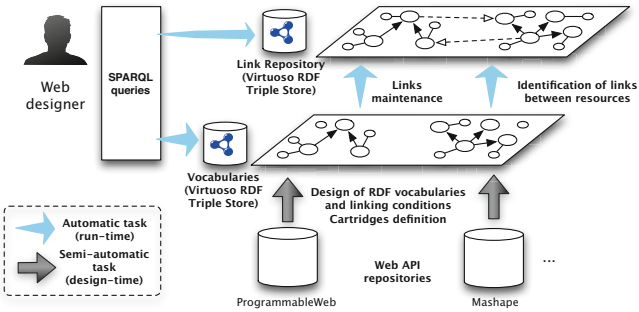

Fig. 1. Overview of the methodology for Linked Web API publication and search

interest: categories, tags, web mashups and actors involved in Web API sharing, namely providers and web mashup developers. Vocabularies are stored within the RDF Triple Store of the Virtuoso Universal Server, on which the approach is implemented. Links across different vocabularies are also stored in a *Link Repository*, built upon the Virtuoso Triple Store. Web API search strategies, based on RDF vocabularies and cross-repositories, are then applied, inspired by [3]. Such strategies are implemented through SPARQL queries issued on the Virtuoso Universal Server to query the contents of different repositories in a combined way. Due to the dynamic nature of Web API repositories, a link maintenance mechanism has been also implemented.

2 Linked Web API RDF Vocabulary Definition

The RDF representation of PW and MP vocabularies, together with cross-repositories links (dashed arrows) are shown in Figure 2. Class modeling should rely on external vocabularies or ontologies, when available [4]. For instance, we modeled actors involved in Web API sharing within the two vocabularies in Figure 2 using the FOAF (Friend of a Friend) ontology.

The relevant classes of resources which have been included in the PW and MP vocabularies reflect the distinctive features of the two repositories for Web API description. The PW repository focuses on the Web APIs, the way they are aggregated into mashups, developers who may be the owners of mashups or providers of Web APIs. On the other hand, the MP repository is a cloud API hub focusing on people involved in Web API sharing, denoted as *mashapers* in the repository, distinguishing among three roles, namely Web API providers, consumers (who used the Web API in one of their own mashups) and followers (who declare their interest on the Web API). This perspective is further enriched through relationships between mashapers.

Semantic links between resources across different repositories are identified, as well as the conditions to be verified to set the links. Formally, we represent a semantic link $\mathcal{L}$ as follows: $\mathcal{L} = \langle \texttt{type}, s\_URI, t\_URI, \texttt{conf}, [\texttt{when}] \rangle$, where

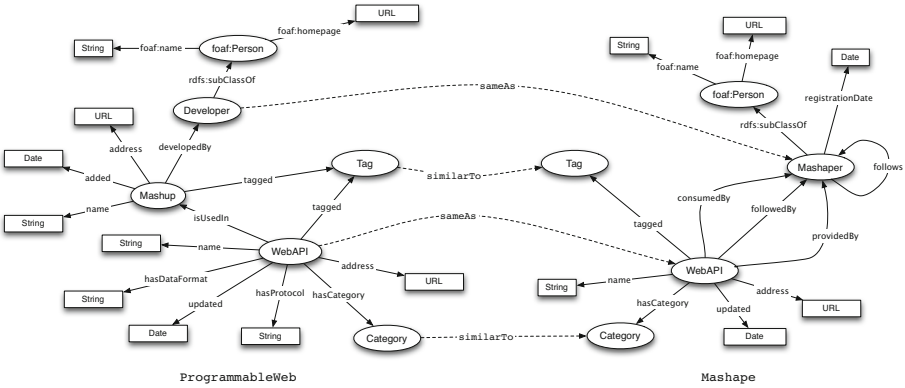

Fig. 2. Linked Web API vocabulary

s_URI and *t_URI* are the URIs of the source and target resources of the link, respectively, `conf` is the confidence to set the link (obtained through similarity metrics depending on the link type) and the optional element `when` denotes when the link has been established and threrefore stored within the Link Repository. With reference to the vocabularies shown in Figure 2, the following link types have been identified:

- `sameAs` link between Web APIs, to denote that two APIs registered in the repositories refer to the same component;
- `similarTo` link between categories and tags used to classify Web APIs;
- `sameAs` link between developers' profiles, to denote that a developer within the `ProgrammableWeb` repository corresponds to a mashaper registered within `Mashape`.

3 Link Exploitation and Maintenance

Contents of the Link Repository can be exposed as open data and used to browse Web API information across PW and MP repositories. A query is formally defined as $\mathcal{Q} = \langle \mathcal{C}_\mathcal{Q}, \mathcal{T}_\mathcal{Q}, \mathcal{F}_\mathcal{Q}, \mathcal{M}_\mathcal{Q} \rangle$, where $\mathcal{C}_\mathcal{Q}$ is the set of categories, $\mathcal{T}_\mathcal{Q}$ is the set of tags, $\mathcal{F}_\mathcal{Q}$ (optional) is a set of pairs $\langle$`tech_feature=value`$\rangle$ and $\mathcal{M}_\mathcal{Q}$ (optional) is a mashup (that is, a set of Web APIs) which the Web API to search for will be aggregated in. A search interface supports designers that are not confident with SPARQL in query formulation.

When the query Q is formulated, the Link Repository is inspected to expand the set of categories and tags specified in the query. The expanded set of categories and tags are used to retrieve Web APIs from the PW and MP repositories. Pairs of retrieved Web APIs (one from PW and one from MP) are compared: if a `sameAs` link between Web APIs is already stored within the Link Repository, the Web APIs are reconcilied, otherwise the link is checked to be established. At the same time, the Link Repository is updated with the new links. Links oldest than

a predefined set of days in the Link Repository are never considered to check semantic links across repositories and are periodically deleted by the system. The last search steps concern filtering and ranking of search results. Retrieved Web APIs are filtered out according to the set of required features $\mathcal{F}_\mathcal{Q}$ if specified in $\mathcal{Q}$. Finally, search results are ranked according to their appropriateness with respect to the target mashup $\mathcal{M}_\mathcal{Q}$ if specified in $\mathcal{Q}$ and according to their popularity. Specifically, we define the similarity between two mashups $\mathcal{M}_1$ and $\mathcal{M}_2$ (as sets of Web APIs) using the following formula:

$$MashupSim(\mathcal{M}_1, \mathcal{M}_2) = \frac{2 \cdot |\mathcal{M}_1 \cap \mathcal{M}_2|}{|\mathcal{M}_1| + |\mathcal{M}_2|} \quad (1)$$

where $|\mathcal{M}_1 \cap \mathcal{M}_2|$ denotes the number of common Web APIs in the two mashups and $|\mathcal{M}_i|$ the number of Web APIs in the mashup $\mathcal{M}_i$. Given the set $M_\mathcal{W}$ of mashups of a Web API $\mathcal{W}$ among search results, the appropriateness of $\mathcal{W}$ with respect to the mashup $\mathcal{M}_\mathcal{Q}$ is given by $max_j\{MashupSim(\mathcal{M}_\mathcal{Q}, \mathcal{M}_j)\}$, where $\mathcal{M}_j \in M_\mathcal{W}$. Popularity of a result is measured as the number of developers who used that Web API in their own mashups.

4 Concluding Remarks

In [3,5] we demonstrated the usefulness of considering multi-perspective Web API description for searching purposes on the `ProgrammableWeb` repository, leaving to the Web API consumers the task of adding information about their profile and the way they used Web APIs in their mashups. On the basis of those results, we moved to the development of a novel approach based on Linked Data principles to provide a comprehensive cross-repositories view of the available Web APIs information. In this paper, we presented the approach by considering Web APIs search on the ProgrammableWeb and Mashape repositories and combining their distinctive Web API descriptions.

References

1. Taheriyan, M., Knoblock, C., Szekely, P., Ambite, J.: Semi-Automatically Modeling Web APIs to Create Linked APIs. In: Proc. of the ESWC 2012 Workshop on Linked APIs (2012)
2. Pedrinaci, C., Liu, D., Maleshkova, M., Lambert, D., Kopecky, J., Domingue, J.: iServe: a Linked Services Publishing Platform. In: Proc. of ESWC Ontology Repositories and Editors for the Semantic Web (2010)
3. Bianchini, D., De Antonellis, V., Melchiori, M.: Semantic Collaborative Tagging for Web APIs Sharing and Reuse. In: Brambilla, M., Tokuda, T., Tolksdorf, R. (eds.) ICWE 2012. LNCS, vol. 7387, pp. 76–90. Springer, Heidelberg (2012)
4. Villazón-Terrazas, B., Vilches, L., Corcho, O., Gómez-Pérez, A.: Methodological Guidelines for Publishing Government Linked Data. Springer (2011)
5. Bianchini, D., De Antonellis, V., Melchiori, M.: A Multi-perspective Framework for Web API Search in Enterprise Mashup Design. In: Salinesi, C., Norrie, M.C., Pastor, Ó. (eds.) CAiSE 2013. LNCS, vol. 7908, pp. 353–368. Springer, Heidelberg (2013)