Using semantic techniques to improve service composition by end users

Giuseppe Desolda

Dipartimento di Informatica, Università degli Studi di Bari Aldo Moro
via Orabona, 4, 70125 - Bari, Italy
giuseppe.desolda@uniba.it

Abstract. My PhD research focuses on supporting non-technical end users to flexibly integrate, into personal interactive workspaces, heterogeneous services available in the Web, in order to satisfy their own personal needs in various contexts of daily life. End users should be enabled to shape their application at use time, and also supported on identifying data and services of interests. The latter requirement addresses an issue about improving the quality of the retrieved services with respect to the end user's goal. As described in this paper, a direction that I am exploring refers to the use of linked open data as a new source of data to be exploited, in order to better fulfill the end user's desires.

Keywords: Composition Paradigms, End-User Development, Linked Data

1 Introduction

Nowadays we are facing the increasing amount of data and services available on Internet. This situation, together with the opportunities offered by Web 2.0, stimulates researchers to investigate new ways for effectively allowing laypeople, i.e., end users without expertise in programming (often called end users in the rest of this paper), to access, manipulate, and combine different kinds of resources in order to generate personalized contents and applications. Mashups have been very much investigated in the last years; they are applications assembled by end users through the integration of heterogeneous resources (APIs, databases, spreadsheets, HTML pages, etc.), in order to solve situational needs. Unfortunately, the proposed mashup platforms are not very end-user oriented [1]. Some mashup tools provide graphical user interfaces for combining services, but the adopted visual composition languages are not suitable for end users, who have difficulties in understanding the integration logic (e.g. data flow, parameter coupling) [2]. Moreover, platforms are usually general purpose and not adequate to the needs of specific application domains and specific end users.

The research I'm interested for my PhD aims at empowering end users to create personal interactive workspaces by combining End User Development (EUD) principles with mashup methodologies and techniques. In this way, services available on the Web can be composed to create new contents and applications by using intuitive and easy-to-use composition mechanisms and platforms. The first couple of years of my research, I investigated recent proposals of mashup compositions and identified models and techniques derived from the lessons learned on End-User Development; they were used in developing a novel methodology and the prototype of a platform for service and data composition to satisfy end users' needs [3]. A key point is that the platform has to offer mechanisms to be customized to a specific domain in which it has to be used, so that, capitalizing on the knowledge of the people working in that domain, it can offer a composition process that is adequate to such people. Thus, the developed platform is based on a meta-design approach and a novel stratification into different design layers. Meta-design permits the involvement of different stakeholders in the design: the first phase (the meta-design phase) is performed by professional developers and consists of designing software environments that allow some stakeholders (including domain experts) to create templates, basic elements, and software environments appropriate for end users in the specific application domain, making possible the domain customization; in the second phase, using such environments, end users are able to compose their Personal Information Space (PIS), in which they integrate and manipulate services of interest. Thus, as explained in details in [3], there are different design layers: at the top layer, professional developers create software environments, services descriptors and visual templates for other stakeholders by using an integrated development environment; at the middle layer, domain experts and professional developers perform participatory design to customize the platform by selecting visual templates and registering and composing services; at the bottom layer, end users can create, use and update their PIS.

2 Using linked open data

User studies recently performed indicated new requirements, such as adding new services, changing their visualization, composing different services. In [4], prototypes developed to satisfy such requirements are presented. Moreover, a need for mashup data quality emerged. From the end users' point of view, data quality also refers to the amount of information provided by the retrieved services and its appropriateness to their needs [5]. Actually, some people using the platform prototype observed that few information were retrieved and with few details, compared to what they expected. This because end users are very diverse and have a great variety of needs. Often, not only a single service does not provide the required information, but even the mashup of more services is not capable to completely satisfy end users, who are increasingly more demanding.

I present here an approach that exploits linked data as a new source to be combined with services: it overcomes the above problem of data quality by adding semantic annotations to the services. Previous proposals tried to annotate services on the basis of manual or semi-automatic approaches [6]. The novelty of this approach is that it exploits automatic annotation of services, as described after the following scenario.

Scenario. Tony, the main persona of this scenario, is an organizer of entertainment events. He is looking for a musician for a local event he is organizing. He has added to his workspace *Grooveshark*, a service that, receiving in input the name of a musical artist, retrieves her albums and songs. Tony wants to know details like age, band, birthplace, etc. To find these details, Tony looks in the platform for some services to be combined with Grooveshark, but he is very disappointed because he does not find anything. How can linked data be useful to solve Tony's problem? The use of linked data enable Tony to query a knowledge base and have support in finding relevant data to be composed with other services.

The term linked data refers to best practices for publishing and linking structured data on the Web. Because they have been adopted by an increasing number of data providers, a global data space containing billions of assertions - the Web of Data – has been created. Linked data will be implemented in our platform by building a meta-layer over the registered services. In this layer, all relevant properties (input, output, etc.) of registered services are annotated by using classes' name of the ontologies adopted in the knowledge base of reference. This approach faces some critical issues: 1) identification of significant service properties to annotate; 2) how to label these properties, manually or automatically; 3) how to use these annotations to 'connect' services with linked data. For 1), being the considered mashups a composition of services oriented to information retrieval, the important properties are input query, output attributes and topic of service. For 2), a solution to automatically annotate services is the following. For each service offered by the platform, the platform administrator produces some significant query examples, which are used by an annotation engine to query these services and to collect many instances. Afterwards, for each service the automatic annotation phase starts: for each instance of each service attribute, the annotation engine queries a knowledge base, e.g. DBpedia, and obtains a set of classes related to the specific attribute instance. An algorithm establishes, for each attribute, the most important classes. The annotation engine annotates each attribute using the relevant classes. The same procedure is applied to the input attribute. This flow is summarized in this pseudo-code.

Annotation algorithm

```
Input: Set $T$ of Triples $t=(s,A,Q)$, $s$ is a service, $A$ is a set of attributes $a$ for s, and $Q$ is a set of query on $s$
for each $t \in T$ do
    create $I$ as empty set of instances results for queries
    for each $q \in Q$ do
        query $s$ by using $q$ and collect instances results into $I$
    end for
    for each attribute $a$ of $s$ do
        create $L$ as empty set of label for $a$
        for each $i \in I$ do
            query DBpedia by using value of $i$ respect to $a$ and obtain a set $C$ of classes
            put values of $C$ into $L$
        end for
        annotate $a$ by choosing most important class of $L$
    end for
end for
```

Starting from the annotation of service attributes, an algorithm establishes the service topic. The proposed approach will be empowered by investigating recent algorithms and techniques proposed in ontology matching research area [7] and methods of semantic annotation of Web service based on DBpedia [8]. The result is a meta-level to each service that describes attributes in term of classes of an ontology. For example, in the case of our previous scenario, the artist_name attribute of Grooveshark is annotated with the Musical_artist class contained in DBpedia ontology. The most critical point is the third problem: how the system can exploit these annotations? To cover the lack of information of previous scenario, the platform offers Tony the opportunity to use DBpedia as new data source. When Tony, starting from Grooveshark, opens DBpedia source in the mashup platform to add details, the system retrieves the classes used to annotate the artist_name attribute (Musical_artist in the previous example) and accesses to the DBpedia ontology to retrieve all possible links of the Musical_artist class with other classes or properties like age, instrument, style, etc. Tony can choose some of these links so that he can view in his future searches new artist's details contained in DBpedia.

This approach allows end users to deal linked data as new rich knowledge base connected with each service through an automatically built meta-layer transparent to the end user. However, this solution presents some open questions: how many classes have to be used to annotate services? How the system should choose the correct abstraction level of the class respect to the ontology? If the links of thighs to show to the users are too many, how the system should visualize all these information?

3 Acknowledgments

This work is partially supported by Italian Ministry of University and Research, grant VINCENTE, and by Ministry of Economic Development (MISE), grant LOGIN.

References

1. Casati, F.: How End-User Development Will Save Composition Technologies from Their Continuing Failures. In: Costabile, M., Dittrich, Y., Fischer, G., Piccinno, A. (eds.), End-User Development - Is-EUD 2011. Vol. LNCS 6654, pp. 4-6. Springer, (2011)
2. Namoun, A., Nestler, T., De Angeli, A.: Conceptual and Usability Issues in the Composable Web of Software Services. In: Daniel, F., Facca, F.M. (eds.), Current Trends in Web Engineering - ICWE 2010. Vol. LNCS 6385, pp. 396-407. Springer, (2010)
3. Ardito, C., Costabile, M.F., Desolda, G., Lanzilotti, R., Matera, M., Piccinno, A., Picozzi, M.: User-driven visual composition of service-based interactive spaces. Journal of Visual Languages & Computing (In print)
4. Ardito, C., Costabile, M.F., Desolda, G., Lanzillotti, R., Matera, M., Picozzi, M.: Visual Composition of Data Sources by End-Users. In: Proc. of AVI '14, in print
5. Picozzi, M., Rodolfi, M., Cappiello, C., Matera, M.: Quality-Based Recommendations for Mashup Composition. In: Daniel, F., Facca, F.M. (eds.), Current Trends in Web Engineering - ICWE. Vol. LNCS 6385, pp. 360-371. Springer, (2010)
6. Zeshan, F.: Semantic Web Service Composition Approaches: Overview and Limitations. Int. Journ. of New Computer Architectures and their Applications 3(1) (2011)
7. Shvaiko, P., Euzenat, J.: Ontology Matching: State of the Art and Future Challenges. IEEE Transactions on Knowledge and Data Engineering 25(1), pp. 158-176 (2013)
8. Zhen, Z., Shizhan, C., Zhiyong, F.: Semantic Annotation for Web Services Based on DBpedia. In: IEEE Int. Symp. on Service Oriented System Eng. pp. 280-285, (2013)