Reputation-Based Self-Service Environments

Donato Barbagallo, Cinzia Cappiello, Chiara Francalanci, Maristella Matera

Dipartimento di Elettronica e Informazione, Politecnico di Milano,
Via Ponzio 34/5, 20133 Milano, Italy
{barbagallo, cappiello, francala, matera}@elet.polimi.it

Abstract. The availability of a huge amount of information on the Web raises a set of issues concerning the research, the selection, and the representation of trustworthy sources and services. This paper describes a new research, which we have recently started and that aims at developing a platform to support users in the creation of mashup-based personalized self-service environments, where the users can access dependable services, selected on the basis of their reputation. This paper illustrates the motivations behind our research, and introduces some preliminary ideas about the platform design.

Keywords: Self-Service Environments, Mashups, Reputation, Quality.

1 Introduction

The Web is a huge and heterogeneous source of information. Web 2.0 technologies have enabled an active role of the users, who can create and make available their contents very easily [7]. This allows people to express their opinions, and to distribute them through several means, such as forums, blog posts and comments, and social networks. It therefore becomes possible to access other people opinions, direct witnesses and spread ideas bypassing traditional and official sources of information such as corporate websites. Of course, the availability of such an amount of information raises a set of issues concerning the research, the selection, and the representation of trustworthy sources and services. The information retrieved on the Web is indeed often characterized by inconsistent, incomplete, and erroneous data, and users are not able to distinguish the right or the most suitable data along their needs. Furthermore, all the accessible sources could be better exploited if they are combined in a mashup wise, so that to obtain a tangible added value. To respond to the previous needs, this paper discusses some issues behind the provision of personalized self-service environments where users can build their view over the Web information space, by integrating trustworthy services for information access.

The remainder of this paper is organized as follows. Section 2 illustrates a scenario that clarifies the novel requirements addressed by our research. Section 3 describes the architecture of the platform that we want to build, while Sections 4 and 5 discuss the two salient themes addressed by our project, namely reputation assessment and the construction of self-service environments based on mashup technologies. Finally Section 6 outlines future work.

2 A Reference Scenario

To understand the idea at the basis of our research, let us consider a usage scenario in the context of *patient empowerment*. We suppose that an individual with a health

problem is searching for a dependable source of information, such as a forum where to get advice on his/her specific problem. This raises a number of issues: (i) the selection of the most dependable forum, (ii) understanding who other participants are trustworthy, and eventually (iii) integrating different answers from multiple forums and dynamically monitor the most reliable sources of information on specific subjects. In addition to discovering the best sources of information, the individual might also add new value to the identified dependable sources by combining different trusted services for creating his/her personalized information access, e.g. combining the identified forum with a map service showing the location of the hospitals mentioned in the forum, as well as with services for accessing rankings, news, and images. This implies the availability of methods for easy service aggregation, which can be based on *mashup* technologies.

To the state of the art, the current technology does not exhaustively cover the previous needs. Our aim is therefore to provide a platform satisfying them.

3 Platform Architecture

The previous scenario highlights two fundamental needs: (i) to select dependable services that can fulfil specific information needs and quality requirements and (ii) to provide users with tools to compose on-demand personalized applications by means of the selected dependable services.

To respond to these needs, we aim at designing a platform providing self-service mashup functionalities, to help people create a personalized Web access environment. Based on a user profile, the platform searches for the "best" information sources, selects relevant and authoritative information, and then wraps it as data services. Data services are then mixed with other services, typically mashup components [8], to create a personalized environment based on mashup technologies.

Figure 1 illustrates the main components of the proposed platform. The *Service Registry* stores a catalogue of services that can be used for the creation of the personalized user environment. They can be *data services*, providing a binding with a dependable data source through a description of the data structure, generic Web services enabling the retrieval of some relevant information, or also UI components [8]. The registry is populated by a domain expert, who is in charge of scouting relevant data sources. A *Reputation Monitor* evaluates the services from the catalogue, and assigns measures based on objective reputation criteria (e.g., institutional reputation). Such assessment is continuously updated: periodically, the *Reputation Monitor* verifies and updates the quality level of the data sources. The output of the assessment activity is a *Reputation Descriptor*, which stores the result of the computation of the reputation measures.

The *Broker* selects the services that best match the specified user's settings (e.g., information needs expressed in the queries that the user submits to the platform) and the user's profile. The services selected by the broker are then used by the self-service environment that helps users create their personal information access environment.

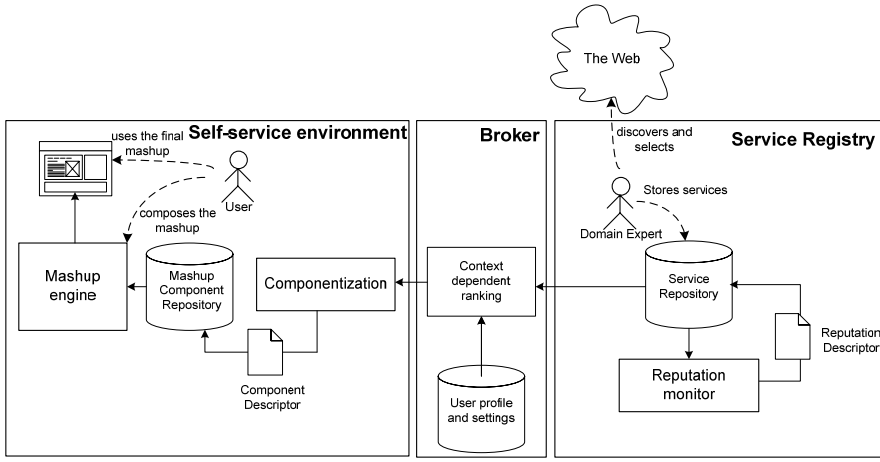

Figure 1- Platform architecture

The *Self-Service Environment* is centred on the availability of a *mashup engine*, through which the user can select some relevant components from the *Mashup Component Repository* and combine them to generate new value. The components available in the repository are those previously selected by the broker. In order to be combined into a mashup, these services need to be *componentized*, i.e. each service must be associated with a descriptor highlighting the properties useful for combination and choreography purposes.

When the services selected by the broker are not provided with a proper user interface, as it happens for pure data services, the componentization process also requires the generation of a presentation layer. This process implies the selection of some visualization widgets (from a widget repository) that best match the operations and the exchanged data as indicated by the service descriptors (WSDL descriptor, API, or also additional service profiles). The previous functionalities raise the research challenges described in the following sections.

4 Reputation Assessment

The selection of the "best" sources, the assessment of their trustworthiness, and the integration of the relevant contents are all based on the ability to assess the reputation of the information sources. The concept of reputation is the result of the assessment of several quality properties of information sources, including correctness, completeness, timeliness, dependability, and consistency [3]. The literature provides consolidated data quality techniques in the case of structured data. For example, a classical approach to data brokering in the context of syndicated data is represented in [1], where data are structured and query answers are fully integrated and returned to users as a table (or a set of tables), as with traditional databases. This work has been subsequently extended with the concept of reputation [2]. In a multi-source context, it proposes to assess the reputation of each information source by means of i) an a-priori assessment of the reputation of the information source, based on the source's authority in a given field and ii) an assessment of the source's ability to offer relevant

answers to user queries based on historical data on the source collected by the broker as part of its service. This approach is original since it defines reputation as a context- and time-dependent characteristic of information sources, and leverages the ability of the broker to keep a track record of each source's reputation over time, but it should be extended with dimensions related to the quality of mashup components to be integrated into the final self-service environment. Quality criteria for the different aspects of traditional software applications have been proposed and thoroughly analyzed in the literature, but the adaptability, dynamicity, and heterogeneity that characterize the mashup ecosystem require a separate and focused analysis.

Our research goal is therefore to define an assessment of the reputation of an information source, e.g., a Web page, based on the assessment of a number of properties of the sources along the traditional data and software quality dimensions, and also on the indirect assessment of the information sources and contributors that the original source includes. This involves the selection of relevant quality dimensions, their operating definition, their metrics, and their composition into an overall assessment of reputation that leverages the historical and contextual knowledge base of the broker.To understand the methodological approach that we will implement in our platform, let us assume that the Web page belongs to a forum. First of all, the Reputation Monitor in Figure 1 must provide an evaluation of the forum's reputation as information source. A source's *completeness* represents a fundamental quality dimension. The completeness of the information provided by a forum could be defined as the breadth of the forum in terms of number of users and issues raised, which, in turn, would lead to the operating definition of completeness as the total number of subscribers/contributors and their participation rate in terms of number of posts per day. Then, we could evaluate the forum's reputation along the *dependability* dimension in terms of the probability with which a post receives an answer, which leads to define dependability as the number of answers per post. This metric could be corrected with a semantic analysis of the relevance of responses. Subjective measures should also be considered, such as comments on the quality of a contribution that the site records explicitly, as different types of ratings, which can be internal to the forum, or external, i.e., stored by sites that aggregate and rate other sites' news. Numerous time-dependent measures of reputation could also be defined, for example along the *timeliness* dimension, by considering metrics such as the frequency of update of a site, its age, etc.

Behind assessing the reputation of the information sources, the broker must be able to evaluate the quality of the mashup components that must provide access to the information source in the self-service environment, also trying to assess the overall quality of possible integrations of multiple components. For this purpose, quality criteria for the different aspects of traditional software applications already proposed in the literature can be exploited, but the adaptability and dynamicity that characterize the mashup ecosystem require a separate and focused analysis. In fact, mashups integrate heterogeneous components available on the Web, such as RSS/Atom feeds, Web services, wrapped content or programmable APIs (e.g., Google Maps). It is self-evident that the quality of the final combination is strongly influenced by the quality of each single component. If we look at components as standalone modules, then we can say that their quality is determined by the attributes that traditionally characterize software quality. However, it is necessary to consider that the publication of mashup

components through APIs hides their internal complexity and, therefore, also their internal details. Given that from the outside component-internal properties cannot be assessed, a novel quality model is required in order to characterize an "external" quality model. Selected quality attributes should be able capture the specific requirements deriving from the components' intended use, i.e., their combination within mashups. In fact, after a component has been deployed, external quality factors are the only criteria able to drive the evaluation of its suitability into mashup compositions. We have already defined a preliminary model of external quality to fill the literature gaps discussed above [4]. Further work is however needed to extend the broker for the coverage of such new quality requirements.

5 Self-Service Environments

Mashups are innovative applications that create new value out of the services they integrate, in that they combine them in a novel, value-adding manner and thereby provide a functionality that was not there before. A variety of mashup tools (e.g., Yahoo Pipes, Google Mashup Editor, Intel Mash Maker, Microsoft Popfly, and IBM QEDWiki - now part of IBM Mashup Center) have recently emerged. Their principal goal is to facilitate the combination of components via simple, graphical user interfaces, sets of predefined components, and abstractions from technicalities. However, they all assume the existence of ready-to-use components published on the Web, while they neglect the ensemble of issues related to the creation and selection of trusted components.

Our research project will exploit the availability of a consolidate mashup environment, *Mixup* [8], which supports the fast development of Web applications based on the mashup of UI components. The distinguishing characteristic of Mixup is that it focuses on the integration of components at the presentation layer, leaving application and data management logic inside components. It also makes use of component and composition models that are inspired by the research on Web services and the service-oriented architecture (SOA). The component model specifies the *events* that a component can generate and that communicate to the outside world changes in the internal state. It also specifies the *operations* which enable the outside world to modify the internal state of a component. Given such abstract component description, the composition logic is then described through an event-driven composition model, where events from one component may be mapped to operations of one or more other components; mappings are expressed by means of so-called listeners. In addition to the direct mapping of events to operations, listeners also support data transformations in form of XSLT transformations, and the specification of more complex mapping logics via inline JavaScript. The definition of listeners represents the composition logic, while the layout of a composite application is specified by means of a suitable HTML template that contains placeholders, which can be used at runtime to embed and execute components, thereby re-using their UIs.

Within this project, we will exploit Mixup as the basic engine for allowing users to mashup their self-service environment. The new issues to be investigated are related to the componentization of trusted services. We want to define some techniques to turn services providing dependable information (Web services, data services, but also Web applications publishing relevant and trusted information) into components for

mashup combination. In [5], we already investigated a technique for turning Web applications into mashup components. The idea is that the HTML of generic (dynamic) Web pages can be augmented with annotations for event and operation tagging and combined with a descriptor specifying such events and operations. A wrapper then provides programmatic access to the application, i.e., an API. Under particular conditions, both the descriptor and the wrapper can be automatically generated from the HTML tags.

We expect to further improve the previous results by introducing techniques for the componentization of generic Web and data services. Such may require mechanisms for the automatic selection of visualization widgets suitable for rendering the service data. Based on the semantic description of services, suitable UI widgets are used to display the data (e.g., a regional map is used to display postal addresses, 2D graphs are used to display historical financial data, graphs with nodes and arcs are used for depicting social networks). For this purpose, a registry of UI widgets, together with their semantic descriptions, will be designed and implemented. This feature, which will go beyond the provision of a mere Web programming environment exploiting ready-to-use components, is to our knowledge still unexplored.

6 Conclusion and Future Work

This paper has presented some ideas about a new research project that deals with the construction of self-service environments for the access to dependable information services. We are now working at the platform development, acting on two different fronts that reflect the two fundamental themes also discussed in this paper. On the one hand we are extending the quality broker with techniques for reputation assessment and for the assessment of the quality of mashup components [4]. On the other hand, we are developing an environment for the easy creation of mashup components, based on an automated technique for user interface construction. Our research is still in its infancy. However, given the recent trends in Web information access, we believe it is very promising and we hope to get soon sound results.

References

1. Ardagna, D., Cappiello, C., Comuzzi, M., Francalanci, C., Pernici, B.: A broker for selecting and provisioning high quality syndicated data. In: 10th International Conference on Information Quality (ICIQ 2005), pp. 262-279, MIT Press, Boston, 2005.
2. Ardagna, D., Cappiello, C., Francalanci, C.: Reputation-based brokering of multisource data. Technical Report n. 12/2009, DEI - Politecnico di Milano, Jan. 2009.
3. Batini, C., Cappiello, C., Francalanci, C., Maurino, A.: Methodologies for data quality assessment and improvement. ACM Comp. Surveys, 41(3), Sept. 2009.
4. Cappiello, C., Daniel, F., Matera, M.: A quality model for mashup components. Proc. of ICWE 2009.
5. Daniel, F., Matera, M.: Turning Web applications into mashup components: issues, models and solutions. Proc. of ICWE'2009.
6. Ennals, R. Garofalakis, M.N.: MashMaker: Mashups for the Masses. Proc. of SIGMOD 2007, pp. 1116-1118.
7. Murugesan, S.: Understanding Web 2.0. IT Professional, 9(4), pp. 34-41, 2007.
8. Yu, J., Benatallah, B., Saint-Paul, R., Casati, F., Daniel, F., Matera, M.: A framework for rapid integration of presentation components. Proc. of WWW 2007, pp. 923-932, 2007.