Service-Finder: First Steps toward the realization of Web Service Discovery at Web Scale

Saartje Brockmans[4], Irene Celino[1], Dario Cerizza[1], Emanuele Della Valle[1 2], Michael Erdmann[4], Adam Funk[5], Holger Lausen[3], Wolfgang Schoch[4], Nathalie Steinmetz[3], and Andrea Turati[1]

[1] CEFRIEL – ICT Institute Politecnico di Milano, Milano, Italy
name.surname@cefriel.it
[2] Politecnico di Milano, Milano, Italy name.surname@polimi.it
[3] seekda, Innsbruck, Austria name.surname@seekda.com
[4] ontoprise, Karlsruhe, Germany surname@ontoprise.de
[5] University of Sheffield, Sheffield, United Kingdom
name-first-letter.surname@dcs.shef.ac.uk

Abstract. The Web is moving from a collection of static documents to one of Web Services. Search engines provide fast and easy access to existing Web pages, however up till now no comprehensive solution exists which provides a similar easy and scalable access to Web Services. The European research project Service-Finder is developed a first version of a portal for service discovery where service related information from heterogeneous sources is automatically integrated in a coherent semantic model to allow effective discovery and to collect user contribution in a Web 2.0 fashion.

Keywords: Web Services, Web Service Discovery, Semantic Web, Crawling, Annotation, Matching, Web 2.0

1 Motivation

The Web is moving from a collection of static documents to a collection of services. For realizing service interchange in a business to business setting, the service-oriented architecture together with the Web Services technology are widely seen as the most promising foundation. As a result, considerable attention has been given in both research and industry to Web Services and related technologies.

Within a Web of services in general, and in any service-oriented-architecture (SOA) in particular, the discovery of services is the essential building block for creating and utilizing dynamically created applications. However, current technologies only provide means to describe service interfaces on a syntactical level, providing only limited automation support. Only with descriptions on a semantic level is a precise discovery possible. Such approaches are developed in various

research projects [SWWS[6], DIP[7]]. However, they are not yet widely deployed. Moreover, existing solutions only cater for scenarios with a limited number of participants. The Service-Finder project[8] is addressing the problem of utilizing the Web Service technology for a wider audience by realizing a comprehensive framework for Discovery by making Web Services available to potential consumers similarly, to how current search engines do for content pages[9].

An essential, but mostly unaddressed problem is the creation of such semantic descriptions of Web Services. The TAO project is the only ongoing project addressing the issue of semi-automatic creation of semantic service descriptions, but it is focused on the specific area of legacy applications and pre-supposes the existence of large documentation of the underlying software. Service-Finder aims to offer automatic creation of service descriptions for a different range of Services (all the publicly available services) and to enable service consumers, not just service providers, to enrich the semantic service descriptions following a typical contribution-based approach in a Web 2.0 fashion.

The paper is structured as follow. Section 2 provides an overview of the most relevant state of the art in Web Service discovery and Semantic Web Service discovery. Section 3 details the architecture of the Service-Finder portal describing the role of each internal component. Section 4 describes the demonstration portal currently available. Section 5 provides conclusions and an overview of the planned future works.

2 State of the Art

Existing solutions for Service Discovery include UDDI, a standard that allows programmatically publishing and retrieving a set of structured information belonging to a Web Service. Several companies have operated public UDDI repositories, however due to several shortcomings of the approach such as complicated registration, missing monitoring facilities, its success was limited and only a few repositories are still publicly available. At the same time a number of Portals dedicated to providing a repository of services have appeared. However, all of them rely on a manual registration and review process, which implies limited coverage as well as inherently outdated information. Alternatively, one can use the classical search engines; however, they do not provide effective means to identify Web Services. For now, there exists no standardized file suffix, such that a query like "filetype:wsdl" does not match all service descriptions (e.g. the wsdl description Microsoft Services will have the ending ".asmx?wsdl"). Moreover, a standard search engine does not make any pre-filtering based on availability and other service-related parameters; their retrieval model is optimized for finding content and not dynamic services.

[6] `http://swws.semanticweb.org/`
[7] http://dip.semanticweb.org/
[8] `http://www.service-finder.eu`
[9] The alpha release of the Service-Finder Portal is available at `http://demo.service-finder.eu`

Current Web Service standards are limited to specify Service Requests as keywords and Service Offers as interface structures (WSDLs). Essentially Web Service technologies only allow describing Services at a syntactical level, this prevents the dynamic discovery and reconfiguration of Web Services to adapt automatically to changes, e.g. that a provider is going off-line or a cheaper provider entering the market. In this context, Semantic Web technologies can provide semantic descriptions that go beyond the syntactic descriptions of Web Services offered by current technologies, describing formally and with well-defined semantics the requester goals and the Web Service capabilities.

Several Description Frameworks (WSMO, OWL-S, SAWSDL) provide the semantic descriptions needed for dynamic location of Web Services that fulfill a given request. They allow, given a goal, the dynamic location of a Web Service. The focus so far has been to capture the functionality of the service in a sufficient level of detail, to ground the accurate matching of requester goals and Web Services. Little research has been conducted on the aspect of how to obtain those descriptions and on other aspects of a service description such as its non-functional properties. Moreover, the approaches developed so far are based on two assumptions: the discovery engine knows all the Web Service descriptions and, the Web Service descriptions are completely correct because an expert draws them up. However, in a Web scenario these assumptions are unreasonable. First, the services are located in different locations that could be unknown a priori, and sometimes they are not available. Secondly, most of the services are described only syntactically, so that their semantics have to be deduced from other available sources, such as service's documentation, Web pages and so on. For this reason, we cannot assume the correctness of the description of a service. Due to these reasons, current approaches for discovering services are not suitable in a Web scenario.

3 The Architecture

Service-Finder overcomes current shortcomings of current Web Service portals and search engines by creating a public Web portal capable of:

- Employing automated methods to gather WSDLs and all related resources such as wikis, blogs or any webpage in which useful information are given;
- Leveraging semi-automatic means to create semantic service descriptions of a Web Service out of the information gathered on the Web;
- Describing and indexing the aggregated information in semantic models to allow matchmaking-based reasoning and to enable fast searches;
- Providing a Web 2.0 portal to support users in searching and browsing for Web Services, and facilitating community feedbacks to improve semantic annotations;
- Giving recommendations to users by tracking their behavior.

Figure 1 shows the general architecture of the Service-Finder Portal accessible at `http://demo.service-finder.eu`. The current components and flows

of data in Service-Finder can be summarized using the continuous arrows. The available services and related information are obtained by crawling the Web (Service Crawler - SC); then, the data gathered is enriched in an analysis step (Automatic Service Annotator - AA) accordingly to the ontologies (Service-Finder and categories ontologies) that models a coherent semantic model. The unified representation of each crawled and annotated service is stored and indexed (Conceptual Indexer and Matcher - CIM) to allow effective retrieval for Web Service search and user contributions on the Web 2.0 style (Service-Finder Portal Interface - SFP). In addition we gather data by analyzing the users' behavioral patterns when they search for services, and use these for Clustering (Cluster Engine - CE).

The seekda search engine[10] is used as an external component delivering four functionalities improving the service-finder portal such as: provide relevant seed URLs to start crawling, supply a rank score for each service (based on various factors such as availability, number of views, online usage, number of inlinks, etc.), provide availability information (over time as graph), and other additional features such as: registration of new services and invocation of services.

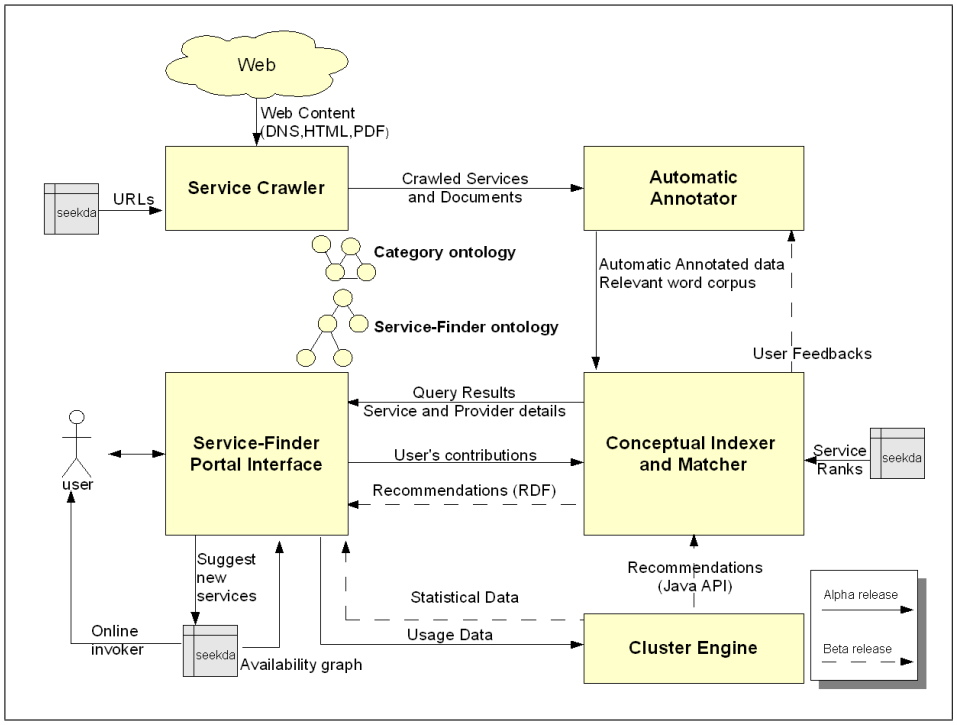

Fig. 1. Overview of the Service-Finder components and dataflow

[10] `http://www.seekda.com`

The dashed arrows in Figure 1 refer to the changes that we intend to implement for the beta release of the Service-Finder portal. While the components in the architecture remain the same, the dataflows are supplemented by some new ones: The CIM will forward the user feedback data to the AA, such that the AA can use the users' re-categorizations of services as training data to improve the subsequent automatic category annotation. The CE will provide cluster data to the CIM, which will be used by the latter to provide recommendations to the SCP.

The following subsections describe briefly the contribute of each component.

3.1 Ontologies

The purposes of the ontologies is to provide a coherent model to collect and manage all the information related to the crawled Web Services.

The Service-Finder ontology focuses in modeling the information related to the core concepts of Service-Finder. The central entity is a service that is associated with several other entities. Several "pieces" of additional information can be gathered around a service. This additional information can stem from the user community as well as from the automatic annotator, respectively the Web. Information extracted by the annotator is obtained by analyzing several documents that can be related to a service. In addition to the service and the information and documents connected to it, we model the provider as the entity that operates a specific service, as well as the user that searches and eventually uses services.

The purpose of the Service Category Ontology is to provide a coarse grained classification to allow users to get an overview of available services. We provide a set of categories based on the experiences gathered during the initial services crawling and operation of the service search-engine at seekda.com. Thus we believe that the chosen categories cover the major domains in which currently publicly accessible services are available.

3.2 Service Crawler (SC)

The Service Crawler is responsible for gathering data from the Web. It pursues a focused crawl of the Web and only forwards relevant data to subsequent components for analyze, index and display purposes. It identifies both services and relevant service-related information on the Web. Around the located services it gathers as much related information as possible, i.e. information from external sources as from the provider's service description, documents or Web pages pointing to that description or to the service, etc.

3.3 Automatic Annotator (AA)

The Automatic Annotator processes the data from the Service Crawler and produces semantic analyses for the Conceptual Indexer and Matcher. We adapted

natural language processing and information extraction techniques (using the GATE platform and bespoke software for Service-Finder) to obtain relevant information from the textual documents (HTML and PDF) relating to the web services and to compare them with the relevant pre-processed WSDL files. The Automatic Annotator processes each provider's set of documents independently, translates the extracted information into RDF-XML or F-logic, and forwards it to the Conceptual Indexer and Matcher, along with compressed archives of the plain text content of interesting documents for keyword indexing. We aim to evaluate the quality of the results by comparing them with a manually annotated sample of the output data and by examining the results of queries made to the Conceptual Indexer and Matcher.

3.4 Conceptual Indexer and Matcher (CIM)

One key principle to realize the goal of effective Web Service discovery is the actual matchmaking process, i.e. retrieving good services for a given user request, related to crawled services and the automatic annotations.

The Conceptual Indexing and Matchmaker is built over the OntoBroker reasoner that internally is based over F-Logic. Since the other components exchange RDF data, to bridge the gap between F-logic and RDF, a particular query language has been developed for the matchmaking component. Moreover, as the matchmaker combines syntactic and semantic retrieval techniques, it takes advantages of both the speed of syntactic search and the expressiveness of semantic relations.

3.5 Service-Finder Portal (SFP)

The main point of access for the final users to the results of the project is the Service-Finder Interface, a Web portal through which it is possible to search and browse for the services crawled, annotated and indexed by the other components. The current version of the Service-Finder Interface[11], provides the basic functionalities to find services, navigate through them, add rating and simple annotations and test the services. The design and implementation of the Service-Finder Interface took into account the requirements of the users, the trends in Web 2.0 and interaction design fields and the research results in applying Semantic Web technologies to the development of portals and Web interfaces via the employment of the STAR:chart framework [1][12]. The logs of users interaction with the Service-Finder Interface are the basis for the work of the Cluster Engine.

3.6 Cluster Engine (CE)

Cluster Engine is responsible for providing recommendations to users meanwhile they are interacting with the Service-Finder portal. The literature presents two

[11] Accessible on the Web at `http://demo.service-finder.eu`
[12] See also `http://swa.cefriel.it/STAR`

different approaches for making recommendations: content-based filtering and collaborative filtering. The first one aims at recommending items that are most similar to the ones that the user already appreciated, while collaborative filtering tries to derive similarities between users in order to recommend items that similar users tend to prefer. We chose to base the Cluster Engine over the collaborative filtering approach since it better fits the collaborative typical of Web 2.0 applications and it also foster the serendipity principle, which can help users in finding services. The Cluster Engine monitors users' behaviour in interacting with the portal to derive users' profiles and, then, compares all users' profiles in order to estimate similarities between users. Exploiting such similarities, it provides users with personalized recommendations.

4 Service-Finder at Work

The Alpha release of the Service-Finder portal is freely accessible at `http://demo.service-finder.eu` where users can found a brief explanation of the portal.

By clicking on the search button, users access the search page. The portal provides three types of search:

Keyword Search enables users to search services by keywords that are syntactically matched against the textual parts of the service model (service name, description, name of its operations) and the related resources (web pages, PDFs).

Category Search supports users in browsing the taxonomy tree and searching for services. The category classification is performed by the Automatic Annotator and is refined thanks to the users contributions.

Tag Search enables users to search for services using the tags that has been associated by other users.

At the time of writing this paper, Service-Finder crawled and indexed more than 25.000 services and about 200.000 related web pages. The results of searches are displayed to the users using standard pagination techniques for long lists.

When users select one specific service, all the details related to that service are shown organized in groups: generic service information; details about the operations with the possibility to invoke the remote service using a tool provided by seekda.com; non-functional aspects of services such as their availability; ratings and comments and, finally, the list of related documents.

Further to the service pages, the portal shows information related to the providers that aggregate services hosted by the same entity. Users can also contribute in suggesting new Web Services by providing the URL of their WSDL.

We envision that the achievements of the project are exploitable in different scenarios:

- Web 2.0 developers may use Service-Finder for building mash-ups of lightweight services.

- ICT Companies engaged in Application Integration scenarios in large enterprises may actively use Service-Finder to look for "address validation", "fraud detection", "credit worthiness", etc.
- The Service-Finder technologies may be sold as an Appliance capable of providing services search capabilities at very low installation and maintenance costs, due to the automated crawling and annotation features.

5 Conclusion and Future Works

Service-Finder provided a first version of a portal able to bring Web Service discovery to a Web scale. The alpha version of the Service-Finder Portal is live at `http://demo.service-finder.eu` and it's gathering feedbacks and contributions from Web users. Such feedbacks will work as inputs for the internal technical activities in order to provide an enhanced version of the portal.

Future works will include added features such as:

- for the Service Crawler, we aim to further enhance the focus of the crawl as well as the overall performance to detect more services and related information. Moreover additional pre-processing will be performed to ease the work for the automatic annotator;
- for the Automatic annotator, we aim to improve incrementally the quality of information extraction, using machine learning techniques to treat feedbacks coming from users' annotations through the Service-Finder Portal;
- for the Conceptual Indexer and Matchmaker, we aim to provide other features like filtering and re-sorting;
- for the Service-Finder Interface, we aim at enhancing the portal by adding more functionalities for the final users, both enriching the current portal (with features like bookmarking and service comparison) and providing APIs to use Service-Finder services via code;
- for the Cluster Engine, we aim to investigate ways to exploit content-based information in order to overcome some of the limitations of collaborative filtering.

6 Acknowledgement

The work described in this paper is partially supported by the European project Service-Finder (FP7- 215876). For further information, please visit `http://www.service-finder.eu` or contact the coordinator of the Service-Finder Project: `emanuele.dellavalle@cefriel.it`

References

1. Irene Celino, Dario Cerizza, Francesco Corcoglioniti, Alberto Guarino, Andrea Turati, and Emanuele Della Valle. STAR:chart Preserving Data Semantics in Web-Based Applications. In Witold Abramowicz, editor, *Proceedings of the 12th International Conference on Business Information Systems (BIS 2009)*, volume 21 of *LNBIP*, page 97108, Poznan, Poland, 2009. Springer-Verlag Berlin Heidelberg.