Web Service Discovery – A Reality Check

Daniel Bachlechner
Digital Enterprise Research Institute
Innsbruck, Austria
daniel.bachlechner@deri.org

Katharina Siorpaes
Digital Enterprise Research Institute
Innsbruck, Austria
katharina.siorpaes@deri.org

Holger Lausen
Digital Enterprise Research Institute
Innsbruck, Austria
holger.lausen@deri.org

Dieter Fensel
Digital Enterprise Research Institute
Innsbruck, Austria and Galway, Ireland
dieter.fensel@deri.org

ABSTRACT
Web services are about the integration of applications via the Web. Hereby, the programming effort should be minimized through the reuse of standardized components and interfaces. Semantic Web services try to provide the next step through mechanizing important sub tasks within a service-oriented architecture. Otherwise, significant manual programming effort would remain as a bottleneck for this approach. One of the sub tasks in a service-oriented architecture is service discovery. While a significant number of papers have already been published in this area, most of them are more concerned in providing yet another illustration for an arbitrary logical framework rather than providing a contribution that meets current constraints in given practical settings. On the poster, we provide a comparison of existing approaches towards Web service discovery based on empirical findings. This sets the basis for analyzing the strengths and weaknesses of the existing approaches as well as the prediction of future potential improvements in this area. We also identify a useful role for semantic techniques as long as it is in a proper setting.

Categories and Subject Descriptors
D.2.11 [**Software Engineering**]: Software Architecture – *data abstraction, domain-specific architectures, information hiding, languages, patterns.*

General Term
Measurement, Experimentation

Keywords
Semantic Web Services, Discovery, Service-oriented Architecture

1. INTRODUCTION
Service-oriented architectures (SOA) emphasize that it is the service that counts for the customer, not the specific software or hardware component that is used to implement it. SOAs will likely become a leading software paradigm quickly. However, they will not scale without significant mechanization of service discovery, service adaptation, negotiation, service composition, service invocation, and service monitoring; as well as data, protocol, and process mediation [5]. Web services are service endpoints in such architecture. If the SOA paradigm succeeds there will be soon several thousand services, which can be used for composing required applications. However, for this, these services must first be discovered. Within the semantic Web community, many of the publications on service discovery tend to place more emphasis on certain aspects of reasoning rather than on focusing on current constraints and foreseeable evolvement of service discovery (cf. [1], [2]). The survey summarized on the poster takes the opposite approach. We enumerate existing approaches for public Web service discovery, compare them with respect to specific criteria and identify their strength and weaknesses. Based on the evaluation's results we conclude potential paths for semantics in Web service discovery as an extension of current approaches.

Demos and Posters of the 3rd European Semantic Web Conference (ESWC 2006), Budva, Montenegro, 11th - 14th June, 2006.

2. SURVEY
Based on previous work [4] we have identified several approaches for Web service discovery that are actually deployed and exceed the scope of a dozen test services. We have examined the standard UDDI registry approach, search via specialized portal sites and customized searches using standard Internet search engines.

2.1 Current Approaches
The first of the three described approaches for current Web service discovery is based on UDDI. UDDI (Universal Description, Discovery, and Integration) is a standard for centralized repositories. The first UDDI Business Registry (UBR) nodes were run by IBM, Microsoft, SAP and NTT Com.

Examples of the second approach are specialized portals which gather services using focused crawlers as well as manual registration. The list of Web service engines investigated within the scope of the study includes XMethods, BindingPoint, WebServiceX.NET, Web Service List, StrikeIron, Woogle, RemoteMethods, and eSynaps. This list of engines includes to our knowledge all relevant portals of the time of writing. Some repositories known from previous work like SalCentral and Grand Central could not be accessed during the time of the survey and hence have not been evaluated.

The third approach uses standard Web search engines which are able to restrict the search to WSDL files. We analyzed the search engines Google and Baidu with respect to their ability to facilitate and enable Web service discovery. Google and Baidu have different means to restrict search queries to specific types of documents, and given the huge size of the underlying document index, both likely to be big players in the long-run.

2.2 Criteria

The criteria used for the evaluation can be classified into two groups, the first of which represents basics for Web service discovery and deals with core criteria like the ways of how a search can be conducted, number of available services, status information, and supported interfaces. The second criteria group consists of service rating, test and demo features (like WSDL analyzer), and service costs which allows service discovery to be more precise and less time-consuming.

3. SURVEY RESULTS

The Web service resources presented on the poster follow many different approaches of service discovery with varying success. The findings of the evaluation are represented in tabular form. The table describes the current approaches in Web service discovery in terms of the introduced evaluation criteria. The three groups of resources as well as the two groups of criteria are clearly separated. We used the developed evaluation scheme to conduct an in-depth comparison of the discovery approaches.

The majority of the approaches relies on keyword search as well as category browsing whereas XMethods only shows services in a simple list format. The UBR also allow searching for service providers and tModels. The Web service search engine Woogle additionally offers template search on operations. Obviously the state of the art of search functionality is rather limited and hampers usability. Semantic Web services could enable a more comprehensive search as well as automation of tasks. Especially in the UBR, the location of Web services is difficult as it does not provide an efficient interface for querying services. The name of a Web service, a Web service provider or a tModel must be known to get further details. The UBR keyword search only takes names into account and ignores service descriptions. Considering service descriptions could be promising in theory, but unfortunately most of the descriptions available are deficient and of low quality. Due to the limited extent of human readable descriptions in the UBR, discovery is a cumbersome and time-consuming effort. Most numbers provided by Web service search engines, concerning the number of registered Web services, are vague and imprecise. However, it is obvious that Google provides a significantly higher number of WSDL files.

The number of available services in terms of a specific discovery provider is an important indicator for the comprehensiveness of a Web service discovery engine. However, at large, service functionality and quality are of course far more important than quantity. Some Web service resources provide functionality to determine whether a service is active or not. The UBR does not provide any status information at all while StrikeIron and Woogle display the status of a listed Web service (active or inactive). BindingPoint allows for excluding inactive Web services from its listings. Another helpful piece of information provided by BindingPoint in this context is the average response times of specific services. All evaluated resources for locating Web services have Web interfaces. Selected ones also provide SOAP and UDDI Private Registry interfaces as well as RSS feeds, WS-Inspection.

4. CONCLUSIONS

Based on our findings, searching with Google has the best coverage, although the precision is limited since there is no single way to restrict a search to only retrieve active and working services supposed to test examples. Most of the public UDDI registries have been discontinued in early 2006, however, due to the limited quality of the contained data, for searching public services they have never been a good source. All existing specialized Web search engines provide less coverage than Google. However, the standard model of Google is not well suited for Web service discovery. Neither the identification of potential services through pure key word extraction nor the relevance ranking based on HTML characteristics such as hyperlinks and title tags provides much of a use in a Web service scenario. The usage of standardized vocabulary such as UNIFACT or eClass to classify Web services could significantly improve the correctness and completeness and do not provide much of a burden to Web service providers. If needed, this task can be mostly automated by approaches such as GoldenBullet [3]. Furthermore the page ranking mechanism of Google that uses the link structure and special properties of HTML documents are not applicable to WSDL files. Therefore, different post processing and filtering mechanisms of the output of Google are needed. This is a task where richer semantic annotations can play a role.

Simple application of IR technologies and later the use of ontologies to describe standard vocabulary are the most promising approaches for the near future. Rich formal frameworks are required as well, but should be considered more in the scope of semi closed environments (e.g. extra nets), where full automation is possible. For the near future the role of central portal providers will most likely become more important in the domain of public services: Take Amazon as an example. The effort in maintaining and developing this central repository is high but it is profitable too. StrikeIron for instance follows a similar business model. However in the long run the business model might be invalidated by the advancement of technology: If semantic Web service technologies advance it is likely that such intermediates (between service provider and consumer) will loose its current importance.

5. REFERENCES

[1] Akkiraju, R., Goodwin, R., Doshi, P. and Roeder, S.: A method for semantically enhancing the service discovery capabilities of UDDI. In S. Kambhampati and C.A. Knoblock, eds., *Proceedings of the IJCAI-03 Workshop on Information Integration on the Web*, pages 87-92, 2003.

[2] Benatallah, B., Hacid, M-S., Rey, C. and Toumani, F.: Request rewriting-based Web service discovery. In *The Semantic Web - ISWC 2003*, pages 242-257, October 2003.

[3] Ding, Y., Korotkiy, M., Omelayenko, B., Kartseva, V., Zykov, V., Klein, M., Schulten, E. and Fensel, D.: GoldenBullet in a Nutshell. In *Proceedings of the 15th International FLAIRS Conferenc*e, Pensacola, USA, 2002.

[4] Fan, J. and Kambhampati S. A Snapshot of Public Web Services, *SIGMOD Record*, 34(1), pages 24-32, March 2005.

[5] Fensel, D. and Bussler, C.: The Web Service Modeling Framework WSMF. In *Electronic Commerce Research and Applications*, 1(2), pages 113–137, 2002.