Design and Development of an Enhanced UDDI for Efficient Discovery of Web Services

K. Tamilarasi[1] and M. Ramakrishnan[2]

[1] Department of Information Technology,
SKR Engineering College, Research Scholar, Sathyabama University
thamizhanna@gmail.com
[2] Department of Information Technology, Vellammal Engineering College
ramkrishod@gmail.com

Abstract. Web Services paradigm has been widely acknowledged by industry and academic research with the progress of web technology and e-business. Increasing number of web applications have been bundled as web services that can be published, positioned and invoked across the web. The importance of the issues regarding their publication and innovation attains a maximum as web services multiply and become more advanced and mutually dependent. This paper proposes an Enhanced UDDI for discovering web services in an efficient manner within a short response time. Initially, the web services are published in the UDDI registry, which is a standard one for publishing and discovering web services. The published web services are indexed and maintained in an index database to enable the effective discovery of web services. For invoking of web services, the search query is matched with the index database and the matched web services are provided to the service customer. The way of accessing the web services is stored in a log file, which is then utilized to provide personalized web services to the user. The discovery of web service is improved considerably by means of an effective searching facility provided by the proposed system and it is capable of providing the most suitable web service.

Keywords: Web service, Web service registry, UDDI, Search engine, Efficient Discovery.

1 Introduction

The utilization of internet as a means of on-line business has increased due to its popularity. Web services have become the preferred technology for such services because of the inherent benefit of loose coupling. Web Services which are becoming a promising technology preferred for constructing understandable applications are internet-based, modular applications and they are of immense interest to governments, businesses, and individuals. The basic web services architecture permits a service requester to find services that are available in service registry where service providers publish their services so that requesters may find them from these registries. In this environment, searching mostly relies on the accessibility and capabilities of the

V.V. Das and J. Stephen (Eds.): CNC 2012, LNICST 108, pp. 109–114, 2012.

repositories for which these services are accumulated. The major specification for forming service-based repositories or registries is Universal Description, Discovery, and Integration (UDDI) [1]. UDDI allows for the description of global registries where, information about services is published. At present, UDDI is the only established standard for Web service discovery across the world [2]. Search engine concept is used in this paper to enhance UDDI registry with the intention of improving the searching facility. At first, the web services are published in the UDDI registry by the service provider. The information updated to the registry is stored as a WSDL document, which contains the businessEntity, businessService, binding template and tModel. To transfer information in between the service provider and the registry, we have used SOAP messages and API calls. Then, the index database is built using the <description> label of the businessEntity and businessService. Here, we have maintained two separate index databases, where we have used inverted index structure to build it. In discovery phase, the search query is matched with the keywords defined in the index database and the key corresponding to the matched keywords are provided to the user. The accessing behavior of the current users is stored in log files, which contain two files for businessEntity and businessService. Once, more number of users use this search engine, the records in the log file will be increased and the personalized businesses and businessServices are provided to the user by analyzing these log files. The basic outline of the paper is described as follows. A brief review of related research is discussed in section 2. The proposed Enhanced UDDI for web service discovery is presented in section 3. Implementation of the proposed Enhanced UDDI registry is given in section 4. Conclusion is summed up in section 5.

2 Review of Related Research

A few researches are available in the literature for web service publishing and discovery that use UDDI registry. The extension of UDDI registry has received considerable attention among the researchers for effectively identifying the web services. Here, we present some of the researches related to UDDI registry. Zongxia Du et.al.[4] have extended and arranged the private or semi-private UDDIs based on industry classifications to present an active and distributed UDDI architecture named Ad-UDDI. Song, H.et.al.[5] conducted an experiment to analyze better techniques of using general-purpose search engines to find Web Services. They have published web services using nine different techniques and retrieved them using Yahoo and Google search engines by employing two groups of total 18 queries.

3 Enhanced UDDI for Web Service Discovery

UDDI [7] is a standard proposed to afford a structure for symbolizing business, business relationships, web services, specification metadata, and web services access points. UDDI offers a searchable directory of businesses and their web services. Keyword search on the names and the features of businesses and service descriptions is the only discovery mechanism provided by UDDI; unfortunately, keyword search cannot distinguish the similarities and the dissimilarities between the capabilities provided by web services. Moreover, in the UDDI registry, the search query is mapped to every WSDL document to find the suitable web services. So, discovering of web services takes too much time to match the query keyword with each WSDL document that contains the business name and service name of the web services, if more web services are published in the registry. To address these problems, this paper proposes a search engine based enhanced UDDI that considerably reduces the computation time taken for discovering relevant web services and also, offers customized web services to the user. The proposed system extends the traditional UDDI registry by incorporating the concept of search engine. The proposed system is capable of providing the following advantages over the traditional UDDI registry, (1) reliable search for web services, (2) potential solution to the scalability problem of web services searching, (3) a wide variety and quantity of web services, (4) most relevant web services to the user using index database, (5) consensus recommendation of web services to the user by mean of user preference database. This section details the proposed search engine-based UDDI for publishing and discovering the web services with less computation time. Initially, web services are published in UDDI registry by the service provider using the technologies like WSDL, SOAP and XML. Then, the web services are delivered to the service customer by matching their requirement with the WSDL document. Here, for matching of web services in an easy way, we make use of the search engine concept, where the published web services are indexed in the index database. The process of constructing the search engine-based UDDI is discussed in the following three steps.

3.1 Service Publishing in UDDI Registry

Here, we discuss about how to publish web services in the UDDI registry. Before discussing the registry process, first we outline the basic concepts of the UDDI registry. The UDDI registry contains two important parts, (i) data model and (ii) UDDI interfaces.

(i) Data Model: The four data structures present in the data model of the UDDI registry are Business Entity, Business Service, binding Template and the tModel. *Business Entity* has information regarding business including its name, short description and fundamental contact details. Each business can also be connected with a distinct business identifier. An individual web service provided by the business entity is represented by the *Business Service* structure. Information on how to bind to the web service, what type of web service it is, and what taxonomical categories it belongs to are included in its description. The information for specifying the technical entry point for a particular web service and depicting service-specific technical features including parameter-specific settings are present in *Business Templates*. Several binding Template objects that share common interface but exist in diverse domain are referenced by the *tmodel*. It should be noticed that *tmodel* encloses a link to the authoritative binding information for that web service and not the actual binding information.

(ii) UDDI interfaces: There are two types of interfaces in the UDDI registry. The *publisher interfaces* are used by the service provider to register their services to the registry. The APIs like *get_authToken, save_business, save_service and save_tModel* are known as publishing API and they are used by service providers to manage the entries that are present in the UDDI registry. The *inquiry interfaces* are used by the services customer to discover their services from the registry. The inquiry API interfaces such as *find_business, find_service, get_businessDetail* and *get_serviceDetail* are used for discovering the web services in the UDDI registry and for retrieving service descriptions about particular registrations. For publishing the web services, at first, the service provider should get authentication token from the registry. So, for getting the authentication token from the registry, the service provider should submit his/her user name and password to the repository. Then, by validating the inputs given by the service provider, the registry generates an authentication token for the service provider. Using this authentication token, the service provider can update the business details to the registry. Once the service provider updates the business details, the registry provides a business key and then, the business key is utilized by the service provider to update the web services. After publishing the business, the registry generates one service key for the user. Then, the service key is used for updating technical details to the registry. For updating all these details, the service provider uses get_authen API, save _business API, save_service and save_tModel interfaces. So, description about business, web services and technical information business provided by the service provider are stored in the registry as a WSDL document.

3.2 Indexing of Web Services

This step is an additional work done in the UDDI registry for easy retrieval of web services and optimization of speed and performance of relevant web services searching. Once the service providers register their services in the registry, the index database, where each service is indexed builds automatically. Indexing is commonly performed in several ways for example using Suffix tree, Inverted index, Citation index, Ngram index or Document-term matrix. In the proposed system, we make use of Inverted index for indexing the web services registered in the UDDI registry. Here, we have designed two index databases I_B (for businessEntity) and I_S (for businessService) using the inverted index data structure. The description given by the service provider in the businessEntity is used to construct the index database I_B that contains two important fields namely, *Keyword* and *business key. Keyword* refers to the significant keywords obtained from the <description> label of the business entity. *Business key* refers to the unique business key of the business entity related to the significant keyword. So, these two fields are added in the index database for each business registered in the UDDI. For a new service provider, the significant keywords are identified and the business key related to the keyword is inserted into the index database along with the keyword. If the keyword is already in the index database, the business key is added to the *Business key* field that corresponds to the keyword. The example of the inverted index of the businessEntity is given in table 1.

Table 1. Illustration of an inverted index for businessEntity

Keywords	Business key
Hotel	key1, key2, key3, key4
Air travels	key1, key2, key3
Hospital	key1, key2
Reservation	key1, key2, key3, key4

Similarly, the index database I_S of the businessService is constructed by taking the significant keywords from the <description> label of the businessService data structure. The database I_S also contains two fields for example, *Keyword* and *Service key*. Keyword refers to the keywords that are present in the <description> label of the businessServices and *Service key* refers to the service key of the web services related to the keyword. The example of the inverted index of the businessService is given in Table 2.

Table 2. Illustration of an inverted index for businessService

Keywords	Service key
Food delivery	key1, key2, key3, key4
Hospitality	key1, key2
Lodging	key1, key2
Wi-Fi connection	key1, key2, key3, key4

3.3 Service Discovery from Index Database

This section describes the discovery of web services from the index database using a keyword-based searching mechanism. When a new user wants to obtain a web service, he/she should put a query (typically by using key word) to the search engine. Then, the input query given by the new user is matched with the keyword field of the index database. The keys corresponding to the matched keyword are taken from the index database and given to the user. Using this procedure, the business and services are discovered and the detailed descriptions of both of them are obtained using the attained key.

4 Implementation

The partial implementation of Enhanced UDDI for web service registry and discovery is done and consists of two phases

4.1 Service Registry Phase

Initially, the service providers publish their business details in the businessEntity using API calls. Then, the service provider's entire services are published in the businessService data model and the technical details are uploaded in the tModel. The details updated by the service provider are stored in a WSDL document, which is located in the UDDI registry.

4.2 Service Discovery Using Search Engine-Based UDDI

In service discovery phase, the query keyword given to the search engine by the new user is mapped with the index database and the keys relevant to the keyword are given to the user to access the web services. Accessing of web services can be accomplished by matching the business query keyword and also, the businessService keyword.

5 Conclusion

We have developed an Enhanced UDDI, which is an extension of the standard UDDI for service publishing and discovery using search engine concept. At first, web services are published in the public registry by obtaining all the service information from the service provider and the published web services are indexed and maintained in the index database. Then, by using the search engine concept, the relevant web services are provided to the user and at the same time, the accessing records are stored in the log file. The proposed intelligent system is an effective implementation of web service registry and at the same time, it is capable of discovering web services efficiently.

References

1. Brian Blake, M., Nowlan, M.F., Bansal, A., Kona, S.: Annotating UDDI registries to support the management of composite services. In: Proceedings of the 2009 ACM symposium on Applied Computing, Honolulu, Hawaii, pp. 2146–2153 (2009)
2. Brogi, A., Corfini, S.: Behaviour-Aware Discovery of Web Service Compositions. International Journal of Web Services Research (IJWSR) 4(3) (2007)
3. Spies, M., Schoning, H., Swenson, K.: Publishing of Interoperable Services and Processes in UDDI. In: Proceedings of the 11th IEEE International Enterprise Distributed Object Computing Conference, Annapolis, MD, p. 503 (October 2007)
4. Du, Z., Huai, J., Liu, Y.: Ad-UDDI: An Active and Distributed Service Registry. In: Bussler, C.J., Shan, M.-C. (eds.) TES 2005. LNCS, vol. 3811, pp. 58–71. Springer, Heidelberg (2006)
5. Song, H., Cheng, D., Messer, A., Kalasapur, S.: Web Service Discovery Using General-Purpose Search Engines. In: Kalasapur, S. (ed.) Proceedings of IEEE International Conference on Web Services, Salt Lake City, UT, pp. 265–271 (July 2007)