

INTELLIGENT SEARCH ENGINE-BASED UNIVERSAL DESCRIPTION, DISCOVERY AND INTEGRATION FOR WEB SERVICE DISCOVERY

¹Tamilarasi Karuppiah and ²M. Ramakrishnan

¹Faculty of Computer Science, Sathyabama University, Chennai, India.

² Faculty of Information Technology, Vellammal Engineering College, Chennai, India

Received 2013-09-02; Revised 2014-01-06; Accepted 2014-04-26

ABSTRACT

Web Services standard has been broadly acknowledged by industries and academic researches along 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. With the intension of determining the web services through effective manner with in the minimum time period in this study proposes an UDDI with intelligent search engine. In order to publishing and discovering web services initially, the web services are published in the UDDI registry subsequently the published web services are indexed. To improve the efficiency of discovery of web services, the indexed web services are saved as index database. The search query is compared with the index database for discovering of web services and the discovered web services are given 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 finding of web service is enhanced significantly by means of an efficient exploring capability provided by the proposed system and it is accomplished of providing the maximum appropriate web service. Universal Description, Discovery and Integration (UDDI).

Keywords: Web Service, Web Service Registry, Universal Description, Discovery and Integration (UDDI), Search Engine, Inverted Index

1. INTRODUCTION

The utilization of internet in terms of on-line business has increased by virtue of its popularity. Now a days Web services were the preferred skill for those services because of the inherent benefit of loose coupling (Thakar and Dagdee, 2009). 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 Tamilarasi and Ramakrishnan (2012a). The construction of web services architecture permits a service suplicant to find facilities that are available in service registry where service providers publish their services so that requesters

may find them from these registries (Thakar and Dagdee, 2009). In this environment, searching mostly relies on the availability and abilities of the sources for which these facilities are accumulated. The major specification for forming service-based repositories or registries is Universal Description, Discovery and Integration (UDDI) (Blake *et al.*, 2009). UDDI agrees for the description of universal registries where, report about services is issued. At present, UDDI is the lonely recognized typical for Web service detection across the world Aznag *et al.* (2013).

The OASIS paradigm protocol UDDI expresses a paradigm set of instructions based on networks in a Service Oriented Architecture (SOA) for issuing and finding software components Lu and Zhang (2009). UDDI provides as a methods for detecting where certain

Corresponding Author: Tamilarasi Karuppiah, Faculty of Computer Science, Sathyabama University, Chennai, India.

Web Services are recommended and who recommends them by describing data structures and API's (programming interfaces) for broadcasting service reports in the registry. Organization for the Advancement of Structured Information Standards (OASIS) categorizes information present in UDDI into white, yellow and green pages. Common report about the organization that provides the services like name, description, address, is comprised in white pages. General classification informations based on standard taxonomies for any of the company or the service offered on industrial groupings are comprised in yellow pages. Comprehensive technical information around the web service that allows one to compose an application to apply the WS are admitted in Green pages. These classes construct both searching industry-specific web services through users and producing client applications to approach them more comfortable Tamilarasi and Ramakrishnan (2012b).

The size and magnitude of UDDI Business Registries (UBRs) are estimated to increase on the growth of web services. Afterward, a main contest specifically to the consumption of primitive search schemes suggested by present UDDI AP is the ability of discovering out web services of importance over multiple UBRs. The time occupied to search presented UBRs in an boundless mode cannot be allowed by clients specifically when functioning by mobile devices. Time effective and extremely useful methods are required for discovering services of interest Tamilarasi and Ramakrishnan (2012b). The searching can be more efficient if the data mining techniques are involved (Rauch, 2012; Cuzzocrea *et al.*, 2011; Telnarova, 2012). In common procedure, the expression "search engine" normally specifies two typical entities; they are, directories and search engines. Directories and search engines essentially contrast in the manner in which their indices of Web-content are constructed. Directories are composed and organized manually. To compose their indices and arrange through information, automated programs which are normally known as "spiders", "robots", "bots" or "crawlers" are utilized by search engines Tamilarasi and Ramakrishnan (2012a).

In this study, we have consumed the search engine model in UDDI registry in order to refining the searching capability. initially, the web services are issued in the UDDI registry through the service provider. The information modernized to the registry is kept as a WSDL document, which comprises the business entity, business service, binding template and tModel. To transmission of information in among 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 business entity and business service. Here, we have sustained two distinct index databases, where we have consumed inverted index structure to develop it. In discovery stage, the search query is compared with the keywords stated in the index database and the key comparable to the coordinated keywords are delivered to the user. The retrieving performance of the current users is kept in log files, which contains two files for business entity and business service. Once, more number of users use this search engine, the reports in the log file will be enlarged and the personalized businesses and business services are offered to the user by analyzing these log files.

The basic outline of the study is described as follows. A brief review of related research is discussed in section 2. The proposed intelligent search engine-based UDDI for web service discovery is presented in section 3. Implementation of the proposed two phase search engine-based UDDI registry is given in section 4. Conclusion is summed up in section 5.

1.1. Review of Related Research

A limited researches are presented in the literature for web service publishing and discovery that use UDDI registry. The expansion of UDDI registry has obtained considerable awareness among the researchers for efficiently identifying the web services. Here, we present some of the researches related to UDDI registry.

Juric *et al.* (2009), to maintain versioning of web service interfaces at development-time and run-time, it offered extensions to WSDL and UDDI. They concentrate on service-level and operation-level versioning, version sequencing and service endpoint mapping. They also obtained annotation extensions for developing versioned web services in Java. For versioning in two real-world environments and recognized significant improvements in service progress and preservation efficiency, better service reprocess and for simplified governance, they tested their result.

Liu *et al.* (2012) to maintain a more automatic and voracious service discovery progress in collaborative manufacturing environments, it offered a fuzzy matchmaking approach for Semantic Web Services. By the comparison degree of facility attributes the membership task is set up on the semantic data annotated on WSDL requirement of a web service and the comparison of service objects is calculated. To demonstrate their approach a combined material assortment case study in the die casting procedure for thermoelectric fan housing is used.

Tian and Huang (2012) to concentrate on these challenges, accessible efforts of combining Open

Geospatial Consortium (OGC) specifications, Universal Description, Discovery and Integration (UDDI) standards and ontologies. To improve breakthrough of web services yielding with OGC specifications and support access to geospatial information via the OGC services is the most important goal. For demonstrating service metadata controlled in the OGC capabilities document via UDDI data structures and providing a way to record semantic, descriptive and technical information of OGC services within UDDI registry to this end, they offered a method for incorporating ontology into UDDI registry and also offered an advertisement algorithm. To increase service breakthrough with the aid of semantic reasoning and to illustrate how this information could be used. Derived from their approaches and Service-Oriented Architectures (SOA) strategies, a web-based prototype system has been constructed sequentially to aid users query, access and visualize geospatial data of dissimilar types in a unified interface. To exhibit the possibility and efficiency of the prototype, numerous running examples are known.

Meditkos and Bassiliades (2011) to improve the process of developing mash ups with semantic mash up innovation abilities a method was offered to merge semantic Web services (SWS) discovery frameworks, UDDI repositories and active mash up tools in order. In the direction of this end, they offered a social-oriented extension of OWL-S advertisements, their mapping algorithm on UDDI repositories and a semantic mash up discovery algorithm. By means of the Yahoo Pipes mash up tool ultimately, they elaborated on the way our framework has been registered.

Juric *et al.* (2009) have proposed extensions to WSDL and UDDI to deal with versioning of web service interfaces at development-time and run-time. They have provided a solution to service-level and operation-level versioning, service endpoint mapping and version sequencing. Also to develop versioned web services in Java, they have proposed annotation extensions. Significant improvements in service development and maintenance efficiency, enhanced service reuse and simplified governance has been recognized when their proposed solution for versioning has been tested in two real-world environments.

Zhang and Liu (2009) have chosen Web service as the key technology for constructing a search engine based on B/S architecture. In their B/S architecture, a number of spider clients were utilized on many computers to finish crawling web pages after all the web pages are examined and refined and by storing some useful web pages into database the index has been

designed. A spider server has been used to control these spider clients. Ultimately, in order for search engine to provide the search services for any web user, Lucene has generated few indexes according to these web pages. In addition, the spider clients contact with the server using Microsoft Message Queue and lots of significant data-operated functions are structured as many web techniques in web service. Distributed spider system of the search engine has been shown to be more proficient than only one spider system by the final experimental results.

Tewari *et al.* (2009) have presented a design of a discovery cum publishing engine for Web service discovery within given optimal response time with refined searching mechanism utilizing service rating methods for successful and productive Web service discovery. They have reduced the search space in UBRs by employing data mining techniques. Moreover, their proposed engine has the capability to publish or search web service across multiple UBRs. Additionally, to store service rating data along with the service information, an extended design of service registry has been proposed. A classification scheme has been followed by the engine to publish the web services in UBR and validation test has been carried out on the discovered web services. Selection of appropriate service by user has been assisted by means of service reviews and ratings.

Chen *et al.* (2011) have proposed an approach called WTCluster, in which both WSDL documents and tags were utilized for web service clustering. Furthermore, they presented and evaluated two tag recommendation strategies to improve the performance of WTCluster. Kim *et al.* (2011) have proposed a cluster-based schema matching scheme. By clustering methods in terms of parameter similarity, the scheme can match methods more precisely than existing schema matching schemes.

The techniques presented in (Du *et al.*, 2006; Song *et al.*, 2007) have not included any user preference services with respect to the historic interest, which can put forward to obtain more. A search engine-based service discovery scheme relevant information. Juric *et al.*, (2009). Also, a frequency-based user preference of web services is not presented in (Tewari *et al.*, 2009). The clustering-based matching schemes given in (Chen *et al.*, 2011; Kim *et al.*, 2011) provided efficient results but the effectiveness of the matching will not be more promisable. So, these works can be further enhanced by adding more intelligence based on user preference by storing log information separately and with effective searching mechanism.

1.2. Proposed Intelligent Search Engine-Based UDDI for Web Service Discovery

UDDI 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 (Zhang *et al.*, 2009). 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, we have developed a two phase search engine that considerably reduces the computation time taken for discovering relevant web services and also, offers customized web services to the user. The proposed web service architecture is shown in Fig. 1.

The proposed system extends the traditional UDDI registry by incorporating the concept of search engine (Henzinger *et al.*, 2002; Ntoulas *et al.*, 2004). 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. It is based on two concepts, they are (i) Search engine-based UDDI, (ii) intelligent search engine-based UDDI. The proposed intelligent search engine-based UDDI for web service discovery is shown in Fig. 2.

1.3. Search Engine-Based UDDI

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.

1.4. 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.

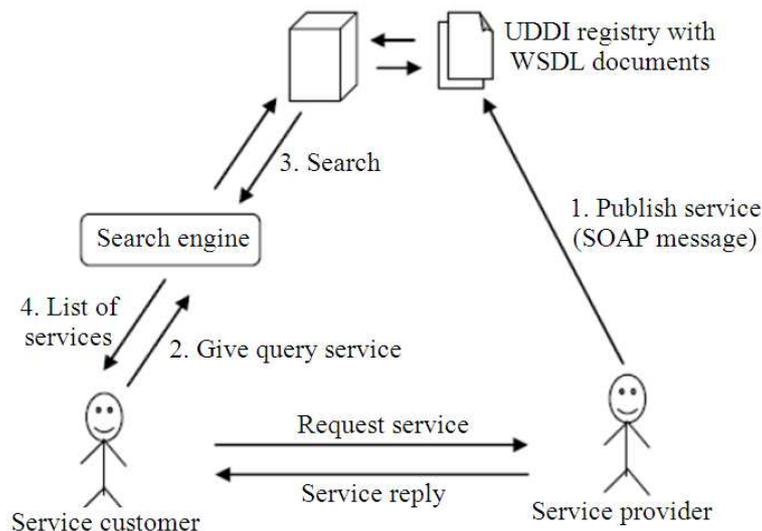


Fig. 1. Proposed web service architecture for web service registry and discovery

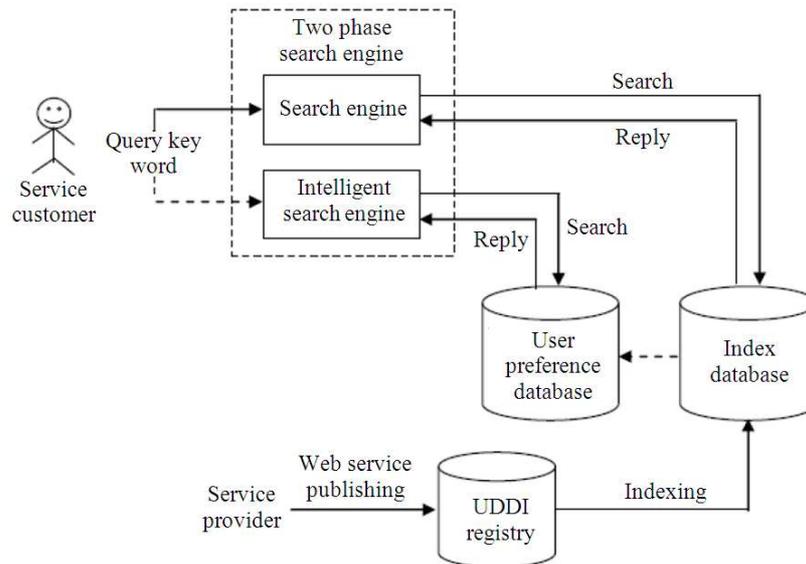


Fig. 2. Two phase search engine-based UDDI

1.4. 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. **Figure 3** illustrates the data model of the UDDI registry.

1.5. UDDI Interfaces

There are two kinds of interfaces in the UDDI registry. The publisher interfaces are consumed by the service provider to record their services to the registry. The APIs like `get_authToken`, `save_business`, `save_service` and `save_tModel` are recognized as publishing API and they are consumed by service providers to handle the contents that are present in the

UDDI registry. The inquiry interfaces are consumed by the services customer to detect their services from the registry. The inquiry API interfaces such as `find_business`, `find_service`, `get_businessDetail` and `get_serviceDetail` are employed for detecting the web services in the UDDI registry and for recovering service explanations about specific registrations.

For publishing the web services, initially, the service provider must borrow authentication token from the registry. So, for borrowing the authentication token from the registry, the service provider should yield his/her user name and password to the repository. Then, by authenticating the inputs specified by the service provider, the registry creates an authentication token for the service provider. Using this authentication token, the service provider can revise the business informations to the registry. Once the service provider modifies the business informations, the registry affords a business key afterwards, the business key is employed by the service provider to revise the web services. After publishing the business, the registry computes a service key for the user. Then, the service key is utilized for revise technical informations to the registry. For updating all these informations, 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 presented by the service provider are saved in the registry as a WSDL document. The service publishing phase is described in the following **Fig. 4**.

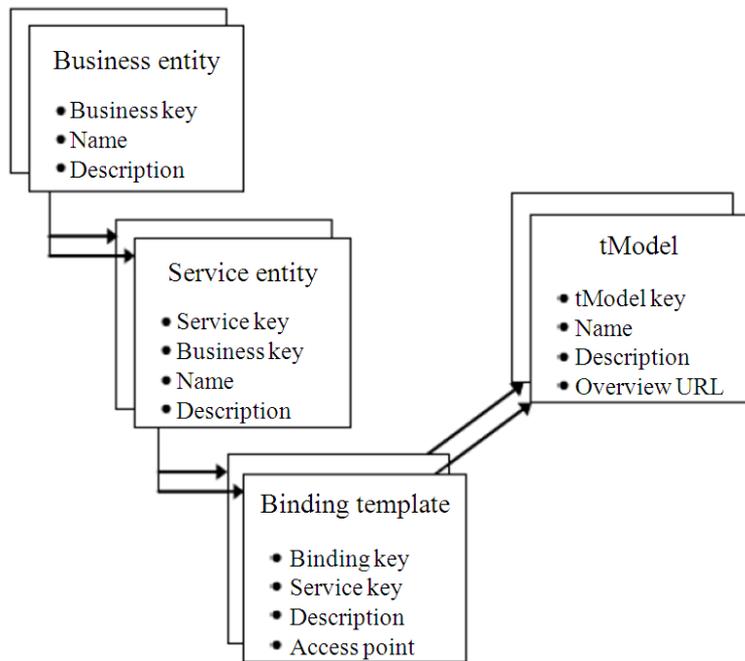


Fig. 3. UDDI data model

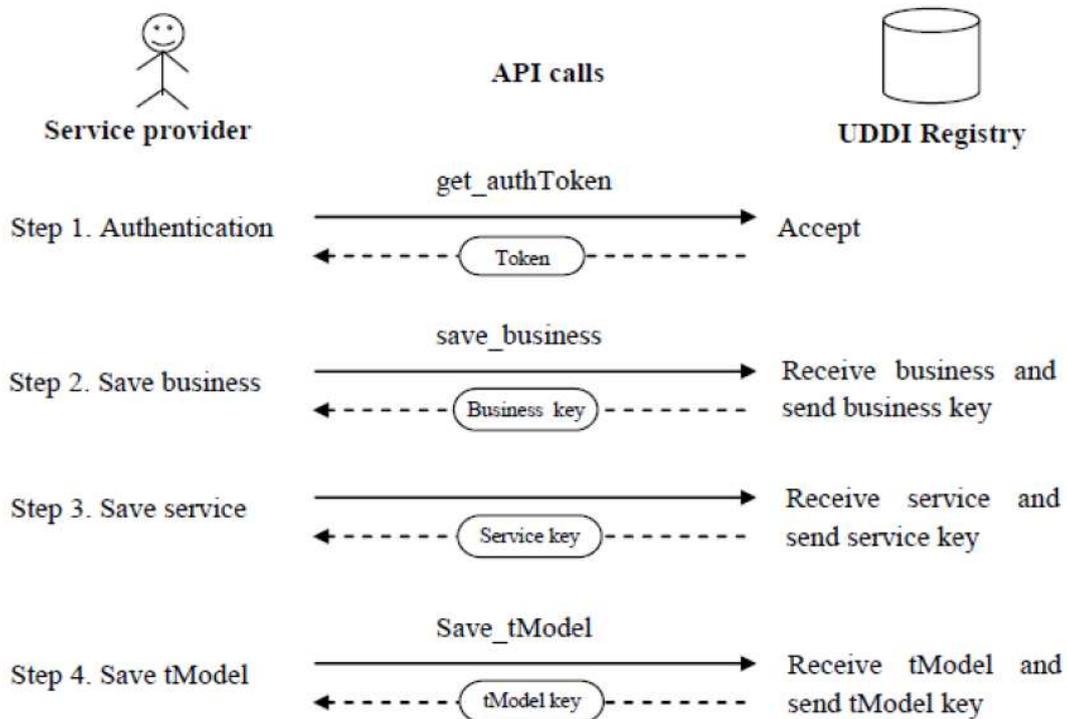


Fig. 4. Publishing of web services in the UDDI registry

1.6. Indexing of Web Services

This step is an additional work done in the UDDI registry for comfortable 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 forms instinctively. Indexing is generally achieved in many methods for example consuming Suffix tree, Inverted index, Citation index, Ngram index or Document-term matrix. In the proposed system, we utilize the Inverted index for indexing (Zhou *et al.*, 2009) the web services published in the UDDI registry. An inverted index is an index data structure which collects a mapping from content, for example words, to its position in a database file, or in a document or a set of documents (Yu *et al.*, 2010).

Here, we have intended two index databases I_B (for business entity) and I_S (for business service) consuming the inverted index data structure. The explanation given through the service provider in the business entity is employed to build the index database I_B that comprises two important areas specifically, *Keyword* and *business key*. *Keyword* mentions to the considerable keywords attained from the <description> label of the business entity. *Business key* indicates to the unique business key of the business Entity associated to the significant keyword. So, these two areas are enhanced in the index database for each business recorded in the UDDI. For a new service provider, the considerable keywords are recognized and the business key associated to the keyword is included into the index database together with the keyword. If the keyword is previously in the index database, the business key is appended to the *Business key* field that matches to the keyword. The example of the inverted index of the business entity is given in **Table 1**.

Likewise, the index database I_S of the business service is composed by capturing the important keywords from the <description> label of the business service data structure. The database I_S also comprises two fields for example, *Keyword* and *Service key*. *Keyword* denotes to the keywords that are existing in the <description> label of the business services and *Service key* signifies to the service key of the web services associated to the keyword. The example of the inverted index of the business service is given in **Table 2**.

1.7. Service Discovery from Index Database

In this section portrays the detection of web services from the index database by means of a keyword-based searching procedure. When a new user desires to attain a

web service, user should laid a query (normally by consuming key word) to the search engine. Afterward, the input query provided by the new user is compared with the keyword field of the index database. The keys compared to the coressponded keyword are captured from the index database and allotted to the user. Using this process, the business and services are exposed and the complete explanations of both of them are gained using the gotten key. The service discovery method from business entity and business service are defined in **Fig. 5a and b**.

1.8. Intelligent Search Engine-Based UDDI

The main aim of constructing the intelligent search engine-based UDDI is to find personalized businesses and their web services. For retrieving personalized services, we should investigate the user's activities of retrieving the businesses and their services from the UDDI registry. The main aim of constructing the intelligent search engine-based UDDI is to find personalized businesses and their web services. The accessing behavior of the initial users is stored in a log file, which is stored in the User Preference Database (D_{UP}). With the help of evaluating this log file, the personalized businesses and their web service are provided to the user. The distinct procedures concerned with the intelligent search engine-based UDDI are illustrated as follows.

1.9. Constructing of User Preference Database (D_{UP})

An intelligent search engine-based UDDI made up of a User Preference Database (D_{UP}), which contains a service log file and a business log file. Both files are automatically created and maintained in the D_{UP} , whenever the service customer uses the search engine-based UDDI to access business and businessServices.

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

Table 2. Example of an inverted index for business service

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

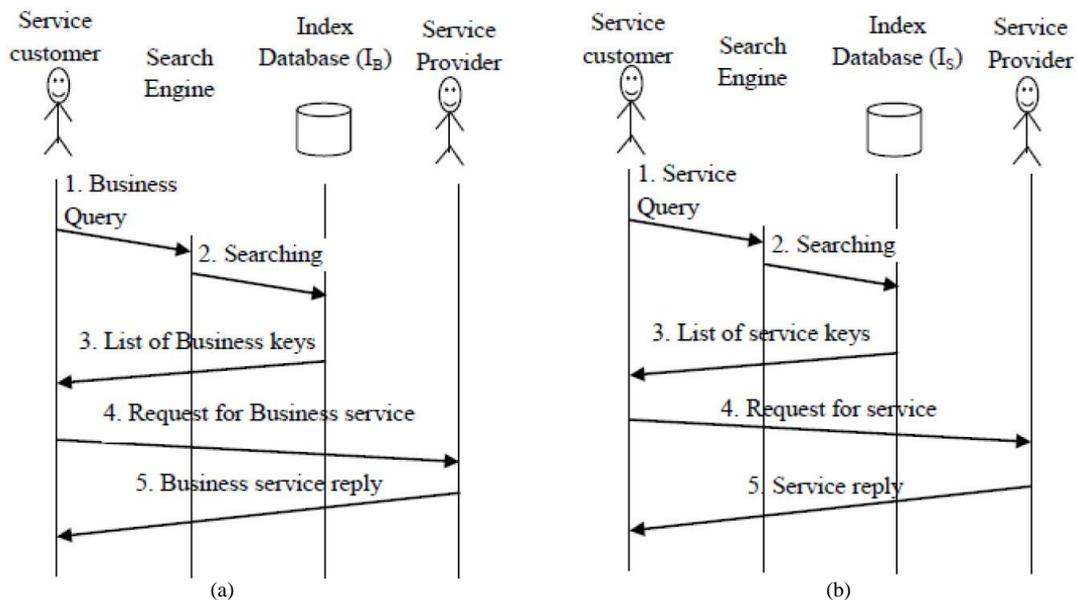


Fig. 5. Web service discovery from search engine-based UDDI, (a) Using index database I_B (b) Using index database I_S

The service log file comprises report around the service customer and his/her retrieving reports of the web services. The succeeding subjects are involved in the service log file for example, <user query, service key1, service key 2,..., service key n>, where, user query indicates the search query distributed by the user and service key denotes to the key corresponding to the interests of the user. Likewise, the business log file contains of user queries and their corresponding interested business, denoted as <user query, business key1, business key2,..., business key n >.

1.10. Grouping of Businesses and Business services

The service and business log files are employed to discover the adapted services and businesses on the basis user deeds. For instance, one record in service log file is represented as: <hotel reservation service> <“uid: EB1B645F-CF2F-491f-811A-4868705F5904”, “uid: EB1B645F-CF2F-491f-811A-4868705F5902”>. From the service log file, the field matching with the user query comprises several categories of search queries such as, hotel reservation service, food delivery, internet facility and room service. With the intention of find the customized services, we should grouping the set of records that have saimilar user query. Consuming the user query field, a set of records with the identical user query is traced and kept as a individual group. For an instance, the records that have the user query

(<hotel reservation>) are captured away and kept in a hotel reservation type. After completing the grouping process, under the business service category, we have a set of groups signified as, $S = \{G_1, G_2, \dots, G_m\}$.

Alternatively, the business log file is denoted as: <Restaurant business> <“uid: EB1B645F-CF2F-491f-811A-4868705F5904”, “uid: EB1B645F-CF2F-491f-811A-4868705G5904”>. Likewise, the records which comprise the similar user query are traced and kept as a group of records for business log file. Thus, from the business log file also, we attain a set of groups signified as, $B = \{g_1, g_2, \dots, g_p\}$.

1.11. Service Discovery using User Preference Database

The customized businessServices and businesses are exposed for the different user with the help of the user preference database D_{UP} . When a new user desires to take a web service, user should place the query (normally by handling key word) to the intelligent search engine-based UDDI. On the basis of input query given by the user, the corresponding group is identified from the user preference database D_{UP} under the business service or business category. Lastly, the key that has the maximum frequency in the recognized group is specified to the user and the described explanation of the customized web services are attained using the reached gained key. The service discovery process from the business entity and business service are described in Fig. 6.

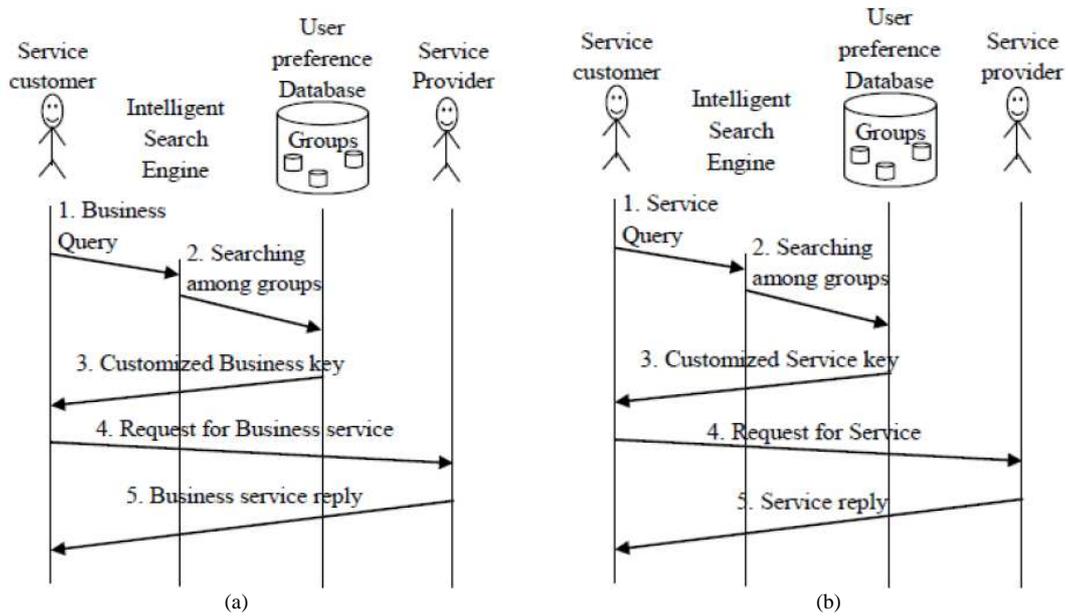


Fig. 6. Web service discovery from Intelligent search engine-based UDDI, (a) Using D_{UP} for business entity, (b) Using D_{UP} for business service

```

<businessentity businesskey="uuid:DC7B41FC-84B3-ACBE-B361-
B7695C3F0A75" operator="http://localhost:8080"
  authorizedname="ragavan">
  <name>alpha</name>
  <description> restaurant
  </description>
  <contacts>
    <contact usetype="general info">
      <description>general information</description>
      <personname>ragavan doe</personname>
      <phone>(044) 234-34534</phone>
      <email>raga@alpha.com</email>
    </contact>
  </contacts>
  <identifierbag>
    <keyedreference
      tmodelkey="uuid:8609c81e-ee1f-4d5a-b202-3eb13ad01823"
      name="d-u-n-s"
      value="5678534" />
  </identifierbag>
  <categorybag>
    <keyedreference
      tmodelkey="uuid:c0b9fe13-179f-413d-8a5b-5004db8e5bb2"
      name="villairs"
      value="112436" />
  </categorybag>
</businessentity>
    
```

Fig. 7. Example of business entity data structure

1.12. Implementation

This section defines the implementation of the search engine-based UDDI for web service registry and discovery. The proposed system was moderately implemented using: Sun Java Development Kit Version 1.6, Apache Tomcat Version 6.0.18 (for application server support) and web service technologies (SOAP, WSDL and JSP).

1.13. Service Registry Phase

At first, the service providers distribute their business informations in the business entity using API calls. Formerly, the service provider's whole services are issued in the businessService data model and the procedural informations are uploaded in the tModel. The informations updated by the service provider are kept in

a WSDL document, which is placed in the UDDI registry. An instance of WSDL files placed in the UDDI registry for business entity, business service and tModel are given in the following code **Fig. 7-9**.

1.14. Service Discovery using Search Engine-Based UDDI

In service detection stage, the query keyword provided to the search engine through the new user is represented with the index database and the keys appropriate to the keyword are offered to the user to access the web services. Accessing of web services can be achieved by matching the business query keyword and also, the business service keyword. The result given by the search engine-based UDDI after entering the business query keyword is given in **Fig. 10**.

```

<businessservice servicekey="uuid:014E27B0-E641-DDEF-8B01-
E87E1C6EB176"
  businesskey="uuid:dc7b41fc-84b3-acbe-b361-b7695c3f0a75">
  <name>regency</name>
  <description> food delievery
</description>
  <bindingtemplates>
<bindingtemplate servicekey="uuid:014E27B0-E641-DDEF-8B01-
E87E1C6EB176"
  bindingkey="uuid:FBA30EC1-B4EC-73AF-5D59-878E0BB852DD">
  <description> fine dining services</description>
  <accesspoint urltype="http">
  http://localhost:8080
  </accesspoint>
  <tmodelinstancedetails>
  <tmodelinstanceinfo
  tmodelkey="uuid:14FDD188-7919-8835-A1D5-3DAD09E0BE40">
  <instancedetails>
  <overviewdoc>
  <description>
  refer the defintion of wsdl service
  </description>
  <overviewurl>
  http://localhost/helloworld.wsdl
  </overviewurl>
  </overviewdoc>
  </instancedetails>
  </tmodelinstanceinfo>
  </tmodelinstancedetails>
</bindingtemplate>
</bindingtemplates>
</businessservice>

```

Fig. 8. Example of business service data structure

```

<tmodel authorizedname="ragavan" operator="http://localhost:8080"
tmodelkey="1286A5F6-2F7A-23A6-B3F8-F3BA2080A220">
  <name> beta </name>
  <description xml:lang="en">
    interface for web services
  </description>
  <overviewdoc>
    <description xml:lang="en">
      wsdl source document
    </description>
    <overviewurl>
      http://mach3.ebphost.net/wsdl/hertz_reserve.wsdl
    </overviewurl>
  </overviewdoc>
</tmodel>
    
```

Fig. 9. Example of tModel data structure



Fig. 10. Screenshot of the search engine-based UDDI after inputting the business query

4.3 Service Discovery using Intelligent Search Engine-Based UDDI

For attaining the web services by intelligent search engine, the services customer can consume a business

name or a business service name. Based on the afforded query keyword, the proposed system supplies customized web services to the user. The result presented by the intelligent search engine-based UDDI after entering the business query keyword is presented in Fig. 11.



Fig. 11. Screenshot of the intelligent search engine-based UDDI after inputting the business query

2. CONCLUSION

We have established an intellectual search engine-based UDDI, which is an advanced version of the standard UDDI for service issuing and detection. The proposed work, search engine comprises dual system, (i) Search engine-based UDDI, (ii) intelligent search engine-based UDDI. Initially, web services are available in the public registry by gaining entire service data from the service provider and the issued web services are indexed and sustained in the index database. Then, by consuming the search engine conception, the appropriate web services are delivered towards the user and meanwhile, the retrieving reports are kept in the log file. With the help of log files, we have invented an intelligent search engine-based UDDI that affords custom-made web services. The proposed intelligent system is an efficient performance of web service registry and meanwhile, it is qualified for offering reliable searching feature aided by the index database. In future, the proposed intelligent system can be extended by incorporating collaborative filtering-based personalization.

3. REFERENCES

Aznag, M., M. Quafafou, N. Durand and Z. Jarir, 2013. Web services discovery and recommendation based on information extraction and symbolic reputation. *Int. J. Web Service Comput.*, 4: 1-18.

- Blake, M.B., M.F. Nowlan, A. Bansal and S. Kona, 2009. Annotating UDDI registries to support the management of composite services. *Proceedings of the ACM Symposium on Applied Computing*, Mar. 08-12, Honolulu, HI., pp: 2146-2153. DOI: 10.1145/1529282.1529756
- Chen, L., L. Hu, Z. Zheng, J. Wu and J. Yin *et al.*, 2011. WTCluster: Utilizing tags for web services clustering. *Service-Oriented Comput.*, 7084: 204-218. DOI: 10.1007/978-3-642-25535-9_14
- Cuzzocrea, A., J.N. Mazon, J. Trujillo and J. Zubcoff, 2011. Model-driven data mining engineering: From solution-driven implementations to 'composable' conceptual data mining models. *Int. J. Data Min. Model. Manage.*, 3: 217-251.
- Du, Z., J. Huai and Y. Liu, 2006. Ad-UDDI: An active and distributed service registry. *Proceedings of the 6th International Workshop on Technologies for E-Services*, Sept. 2-3, Trondheim, Norway, pp: 58-71. DOI: 10.1007/11607380_6
- Henzinger, M.R., R. Motwani and C. Silverstein, 2002. *Challenges in web search engines*. Pennsylvania State University.
- Juric, M.B., A. Sasa, B. Brumen and I. Rozman, 2009. WSDL and UDDI extensions for version support in web services. *J. Syst. Software*, 82: 1326-1343. DOI: 10.1016/j.jss.2009.03.001

- Kim, B., H. Namkoong, D. Lee and S.J. Hyun, 2011. A clustering based schema matching scheme for improving matching correctness of web service interfaces. Proceedings of the IEEE International Conference on Services Computing, Jul. 4-9, IEEE Xplore Press, Washington, DC., pp: 488-495. DOI: 10.1109/SCC.2011.33
- Liu, M., W. Shen, Q. Hao, J. Yan and L. Bai, 2012. A fuzzy matchmaking approach for semantic Web services with application to collaborative material selection. *J. Comput. Indus.*, 63: 193-209.
- Lu, W. and D. Zhang, 2009. Research of enterprise application integration base on service oriented architecture. Proceedings of the International Conference on Computational Intelligence and Software Engineering, Dec. 11-13, IEEE Xplore Press, Wuhan, pp: 1-9. DOI: 10.1109/CISE.2009.5362693
- Meditskos, G. and N. Bassiliades, 2011. A combinatory framework of Web 2.0 mashup tools, OWL-S and UDDI. *J. Expert Syst. Applic.*, 38: 6657-6668. DOI: 10.1016/j.eswa.2010.11.072
- Ntoulas, A., J. Cho and C. Olston, 2004. What's new on the Web?: The evolution of the web from a search engine perspective. Proceedings of the 13th International Conference on World Wide Web, May 17-22, ACM., New York, NY, USA., pp: 1-12. DOI: 10.1145/988672.988674
- Rauch, J., 2012. EverMiner: Consideration on knowledge driven permanent data mining process. *Int. J. Data Min. Modell. Manage.*, 4: 224-243. DOI: 10.1504/IJDM.2012.048105
- Song, H., A.D.C. Messer and S. Kalasapur, 2007. Web service discovery using general-purpose search engines. Proceedings of the IEEE International Conference on Web Services, Jul. 9-13, IEEE Xplore Press, Salt Lake City, UT, pp: 265-271. DOI: 10.1109/ICWS.2007.190
- Tamilarasi, K. and M. Ramakrishnan, 2012a. Design and development of an enhanced UDDI for efficient discovery of Web services. Proceedings of the 3rd International Conference, Advances in Communication, Network and Computing, Feb. 24-25, Springer Berlin Heidelberg, Chennai, India, pp: 108, pp: 109-114. DOI: 10.1007/978-3-642-35615-5_16
- Tamilarasi, K. and M. Ramakrishnan, 2012b. Design of an intelligent search engine-based UDDI for web service discovery. Proceedings of the International Conference on Recent Trends in Information Technology, Apr. 19-21, IEEE Xplore Press, Chennai, Tamil Nadu, pp: 520-525. DOI: 10.1109/ICRTIT.2012.6206753
- Telnarova, Z., 2012. Data modelling and ontological semantics. *Int. J. Data Anal. Techniques Strateg.*, 4: 237-255.
- Tewari, V., N. Dagdee, I. Singh, N. Garg and P. Soni, 2009. An improved discovery engine for efficient and intelligent discovery of web service with publication facility. Proceedings of the World Conference on Services-II, Sept. 21-25, IEEE Xplore Press, Bangalore, pp: 63-70. DOI: 10.1109/SERVICES-2.2009.21
- Thakar, U. and N. Dagdee, 2009. An Approach to discover web service providers based on security support. *Int. J. Web Applic.*, 1: 47-56.
- Tian, Y., M. Huang, 2012, Enhance discovery and retrieval of geospatial data using SOA and Semantic Web technologies. *Expert Syst. Applic.*, 39: 12522-12535. DOI: 10.1016/j.eswa.2012.04.061
- Yu, G., X. Xie and Z. Liu, 2010. The design and realization of open-source search engine based on Nutch. Proceedings of the International Conference on Anti-Counterfeiting Security and Identification in Communication, Jul. 18-20, IEEE Xplore Press, Chengdu, pp: 176-180. DOI: 10.1109/ICASID.2010.5551507
- Zhang, H. and J. Liu, 2009. Search engine design based on web service and lucene. Proceedings of the WASE International Conference on Information Engineering, Jul. 10-11, IEEE Xplore Press, Taiyuan, Shanxi, pp: 458-461. DOI: 10.1109/ICIE.2009.100
- Zhang, J., X. Yu, P. Liu and Z. Wang, 2009. Research on improving performance of semantic search in UDDI. Proceedings of the Global Congress on Intelligent Systems, May 19-21, IEEE Xplore Press, Xiamen, pp: 572-576. DOI: 10.1109/GCIS.2009.74
- Zhou, B., T. Huang, J. Liu and M. Shen, 2009. Using inverted indexing to semantic WEB service discovery search model. Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing, Sept. 24-26, IEEE Xplore Press, Beijing, pp: 4872-4875. DOI: 10.1109/WICOM.2009.5301035