Discovery and Composition of Link Open Data based RESTful Web services

Kirit J Modi[1], Sanjay Garg[2]
[1 2]CSE Department, Institute of Technology, Nirma University, Ahmedabad, India
kirit.modi@ganpatuniversity.ac.in, gargsv@gmail.com

Abstract : Web services are now widely utilized which demands to perform the discovery and composition process to meet the user request. By considering services as the base, an architectural model has been provided by Service Oriented Architecture (SOA). These web services are also known as traditional web services or SOAP web services. These web services increases overhead of XML processing at description and communication phase which could be resolved by REST (Representation State Transfer) architecture based web services which are defined as RESTful web services. The RESTful service discovery is performed to find services for a given user request while RESTful service composition defines service integration when single service is unable to serve the user request. RESTful service composition does not depend on either a central conductor service like a service orchestration or a service choreography specification, which makes it more demanding compared to traditional web services. As a result, we have proposed a REST-based approach for services discovery and composition along with experimental work to show the performance.

Keywords: Link Open Data, Representational State Transfer, RESTful Web service, Web service, Web service composition.

1. Introduction

Web services can be defined as a software component which is accessible over the web. Figure 1 depicts web service model consist of three main entities: Service provider, Service Consumer and Service Broker, each performs publish, find and service registration role respectively. According to the architectural style, web services can be classified into following two categories [26]: (i) SOAP Web services and (ii) RESTful web services

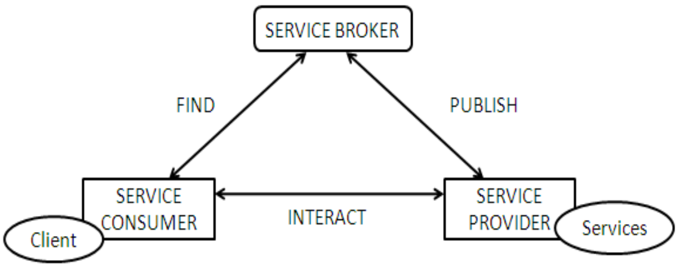

Figure 1.Web Services Model

.

1.1 SOAP Web Services

In SOAP web services, communication is based on standard Simple Object Access Protocol (SOAP) [29] which uses XML-messaging system.

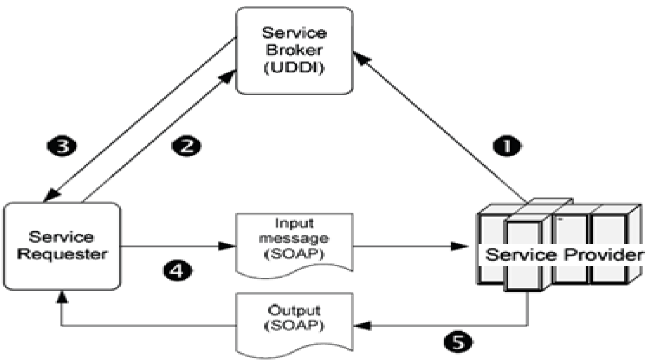

Figure 2. SOAP-based Web Service Model

As shown in figure 2, the web services offered by the service provider are described using WSDL (Web Services Description Language) [30] and registered to the UDDI (Universal Description, Discovery and Integration) [31] registry which contains all necessary information to identify the web service.

Once a requestor has queried the UDDI registry and found the required service, the WSDL file of that service could be derived from the registry and consume via SOAP message.

1.2 RESTful Web Services

In RESTful web services; communication is performed using REST approach as represented in figure 3**.** Representational State Transfer (REST [32]) defines behaviour of web application communication where application presented through collection of web pages, the user navigates through an application by selecting links, resulting in the next page (next state of the application) being presented to the user.

RESTful web services [25] are based on REST architectural model which follows the traditional mechanism of World Wide Web (WWW) and HTTP [33] protocol principles. These services follow four design principles: Utilize HTTP operations explicitly, Stateless in nature, Represent directory pattern-like URIs and Transfer data using XML, JSON or both.

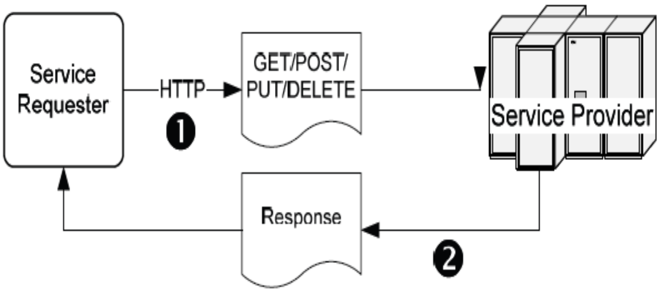

Figure 3. RESTful Web Services Model

1.3 Semantic Web

Semantic Web [34] is represented as an extension of the existing web by enriching the information with well-defined meaning, so that computers can do more work in processing to convert it in to knowledge. Semantic web services are derived from the web standards for the interchange of data, which makes it easy to combine data from heterogeneous sources and services without losing meaning.

1.4 Linked Open Data

"Linked Open Data (LOD) [35] is a new field of research to be used as a representational style for complex data. The term Linked Data is used to refer to a set of best practices for publishing and connecting structured data on the Web as a solution of the semantic web."

1.5 RESTful Web services Discovery and Composition

RESTful service discovery performs the operation to locate services for a given request while RESTful service composition emphasizes on creation of value-added services on-demand when single service is unable to serve the request.

RESTful web services are gaining popularity due to the following reasons. In REST HTTP methods i.e. GET, PUT, POST and DELETE are utilized to access resources via individual URLs rather than overloading operations (usually over HTTP POST) on single endpoints in SOAP RPC manner. Moreover, many Linked Open Data providers simply deliver SPARQL endpoints, and one standard RESTful interface. Nowadays, RESTful services are increasing over the web which can directly generate and consume RDF Data.

The remaining contents of this paper are as follows: Section 2 of the paper discusses the literature review undertaken and the work of fellow researchers. Section 3 describes the proposed work on RESTful web services discovery and composition. Section 4 discusses the experimental work and the conclusion and future work is discussed in the section 5 of the paper.

2. Related Work

In this section, we discuss a work has been carried out in the area of RESTful web service discovery and composition : In [1] an algorithm based on the directed acyclic graph is proposed which represents semantic functional dependency between input and output of mashup automatic web API which can satisfy the desired goal. The composition process can be described as that of genera-ting directed acyclic graphs (DAGs) that can produce the output satisfying the desired goal, where the DAGs are gradually generated by forward-backward chaining of APIs. This approach facilitates to locate the desire APIs with use of API composition. In this approach no technique is mentioned for performance and scalability measures where as we have measured performance of our work.

The Depth First Search algorithm for semantic web service search framework & composition is proposed in [2] for enhancing the search response.The semantic search services offered relies on the RDF data and its corresponding Ontologies built to provide search responses. As a result this method improves response time and support for ranking based on the ontology relevance score.

Due to expensiveness of service call, method [3] aims to decrease the number of calls in order to retrieve results with sufficient recall. The information that could be dynamically obtained from Web services in order to enrich RDF knowledge bases whenever the knowledge base does not suffice to answer a user query and query generator composes sequences of function calls based on the available service interfaces. Here in this approach author has provided RDF representation of Data. But author doesn’t give detail of discovery or composition algorithm.

REST-based method is utilized in [4] to solve certain limitations such as tightly coupled invocation, performance in terms of response data, non-uniform interface and no hyperlink support. Here Web Processing Service interface specification provides an approach to publish and execute geo-processes on the Web. Geo-spatial analysis plays critical role to generate alerts and recommendations.In this approach Authors have provided android based user interface with local language support but doesn’t provide performance measurement detail of the proposed algorithm.

Concept of self-descriptive RESTful service is defined in [6] as- a service that represents itself according to REST principles, to enable effective discovery by humans and machines. A self-descriptive RESTful service utilizes a unique URI which identifies a root resource as a starting point for discovering all its resources. With this approach author has given good explanation about restful web services and given framework but algorithm related detail is not covered where as we have provided service discovery and composition algorithms detail along with framework.

To minimize the manual efforts and to get the results faster, author [7] proposes the use of Semantic Web Technology like OWL, RDF and SPARQL to retrieve the information from the documents efficiently. This work proposes a model for storing the content of a DOC/HTML document in the RDF format and thereby enabling the information retrieval using SPARQL.XPATH and XQUERY that extract information from the XML files that are created by the details provided. Author has given good idea about how SPARQL, RDF and OWL are utilized. But information about standard dataset for performance measurement is missing.

In [8], author provides an approach to fully exploit the growing amount of heterogeneous geospatial data to users without any background knowledge. He has modelled and published the extracted geospatial data to RDF for eliminating the data heterogeneity. Author has proposed a geospatial system as well as algorithm but doesn’t focus on comparison and optimization aspects.

In [36], in this approach author has represented about how web service gives semantic results using RDF and linked open data. In this approach author has also discussed about domain driven approach without representing framework and algorithm detail.

3. RESTful Web Services Discovery and Composition

A proposed framework for RESTful Web Services Composition is based on Composition module which contains the Discovery and Composition modules is represented in figure 4.

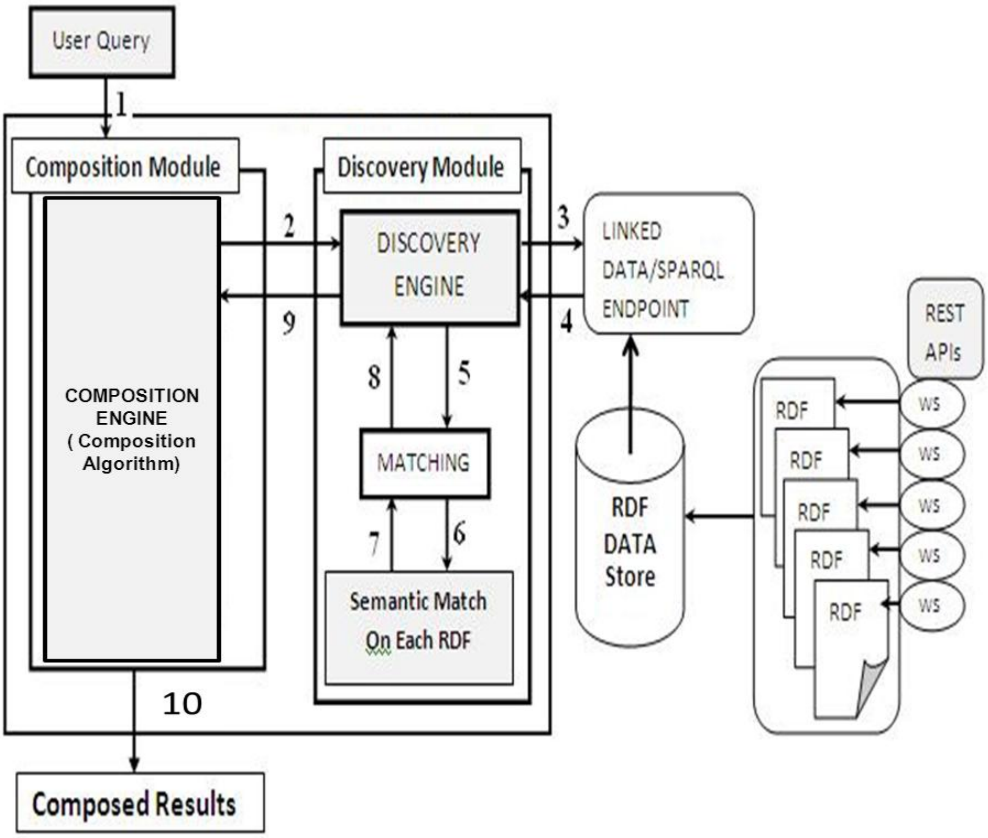

Figure 4. RESTful Web Services Discovery and Composition Framework

As shown in figure 4, there are RESTful web services are present and relevant RDF files are describes the semantic relations between the services. These RDF files are stored in RDF data store. When user's request for input and output parameters which are given to the composition phase module, in which composition module first request given to the discovery module where discovery module performs semantic match based on user request. The semantic concepts are retrieved from the RDF data store based on matching the concept by SPARQL query which gives the list of base URIs of resources. After that theses URIs and input requests given to composition module which contains the composition algorithm. Composition algorithm is matching the linked semantic relation between the resources and it will generate the composed results.

3.1 Discovery Algorithm

The web services discovery problem focuses on searching the services from the repository which contains RDF data as per the query requested by the user.

Dataset D = {RDF1, RDF2, RDF3,………., RDFn}
Resultset R = { T1,T2,..Tn} // Match Resultset which contains temporary result sets
UQuery Q = Query Request from the user
UQuery O = Output Request from the user

Algorithm1:Discovery Algorithm(D, Q, O,R)

1. Initialization ERDFCount = 0;
//Query Executed on RDF file from Dataset D
2. Initialization R= { } or R=NULL;
3. If D is not Exist **then**
4. Exit;
5. EndIF
6. Else
7. ForEach RDF in Dataset D **do**
8. **Initialization** T= { } or T=NULL;
// Temporary Result set from each RDF
9. **If (**match<RDF,Q,O>**)** then

```
10.             T ← Evaluation on Current RDF;
11.         Endif
12.         If   T! = { } or T!= Null then
13.             Resultset R =     R  U  T
14.         Endif
15.         ERDFCount = ERDFCount + 1;
16. EndForEach
17. RETURN    Resultset R;
18. EndElse
```

Algorithm 1. RESTful Web Services Discovery Algorithm

In the above proposed algorithm we have taken input dataset which contain RDF file(semantic description) of REST API. As per the user request we have to check each RDF file and filtering the semantic description. Step1&2: ERDFcount is variable which calculate how many RDF semantic files are executed from repository and it's initial value set to zero. And another variable R is the Resultset which contains matching result as per the query requested by the user. Step 3 to 5: It describes termination condition when repository is null at that time we are exit from the discovery engine and resultset gives empty set. Step 7 to 16: It extracts all semantic (RDF) knowledge data files from the repository. For each RDF data file we have to match the user requirements and matched results generated from each and every RDF data files which will store into temporary result set T . Here, T which contains base URI of the semantic RDF data files.

In the Discovery phase we are matched the user requirements and if semantic match found the we have to extract the base URI of the RDF data file and this URI contains the Resources information's in the RDF data store. Discovery algorithm in which we have described the extract all the result from each RDF file from the dataset, in discovery algorithm step 7 to 16 which will execute for each RDF so the complexity of this algorithm is **O(n)**.in which **n** describes the number of RDF file in the dataset. After successful evaluation on single RDF data match result set R is update the value of temporary result set T and ERDFCount variable increment with one. While reading next RDF file temporary result set T store the current RDF evaluation.

3.2 Composition Algorithm

As per the query requested by the user and collection of services or, in case a matching service is not found, searching a series of services that can be composed together is defined as the composition problem of web services.

```
Resultset R = { T1,T2,..Tn}         //Match Resultset which contains temporary result sets T.
UQuery Q = Query Request from the user.
UQuery O = Output Request from the user
SparqlResultset S = Contains the SPARQL query executed results.
```

Algorithm2:Composition Algorithm(R,Q,O)

```
1. Initialization ERCount = 0;
        //Query Executed on Result set file from Resultset R
2. Initialization R={ } ,S={ },E={ };
3. If  R == { } then
4.      Exit;
5. EndIF
6. Else
7. IF (R != Null)  then
8.      Foreach  T[i] in Resultset R   do
9.              Foreach NODE in R do
10.             E=RecusiveFunction
                (ExecuteSPARQLQuery (T[i]← (SPAQLQuery));
11.             If(E != Null) then
12.                     S = S U E
13.             Endif
14.             ERCount = ERCount + 1;
```

15. EndForEach
16. EndForEach
17. EndIf
18. RETURN S;
19. EndElse

Algorithm 2. RESTful Web Services Composition Algorithm

In the above proposed algorithm we have taken input Resultset R which contains semantic matched information from the discovery engine. Now, as per the user request we have to retrieve semantic composed result from each RDF data in the R by executing SPARQLQuery. Step 1-2:ERcount is variable which calculate how many semantic data are executed from Resultset R and it's initial value set to zero. And another variable S and C initially set to Null where SparqlResultset S is the the SPARQL query executed results which contains matching result as per the query requested by the user on each data of R and Compositionset C contains the composed result generated. Step3 to 5: It describes termination condition when Resultset R is null at that time we are exit from the composition engine and C gives empty set. Step7 to 17: It extracts all semantic (RDF) knowledge data from the resultset R. for each data we have to execute the SPARQL Query as per the user requirements and match results generated from each and every data which will store into temporary result set E which contains semantic information extracted by Executing sparql query. After successful evaluation on single semantic data matching on result set R, the value of temporary resultset E is updated by sparqlresultset S and ERCount variable increment with one. While retrieve information from next data, temporary result set E store the current evaluation. Step18:At last, successful evaluation on semantic data matching on result set R,we have to execute query on sparqlresultset S for getting maximum semantic match result and it will stored into Compositionset C.

In this algorithm step 7 to 17 will be executed for each node of RDF as well as it will extracts the semantic properties of the services, so the complexity of this algorithm is **O(n * m)**.in which **n** describes the number of RDF file in the dataset and m describes the number of nodes we have to check for composition on each file.

4. Experimental Work and Results

In the following section we have discussed the experimental work performed using proposed approaches i.e. discovery and composition. A possible query for the composition process is: Input Query = zipcode, Output Query = city, latitude, longitude. A prototype based on Census Information System using proposed approaches is described as below.

4.1 Census Information System using Geospatial Data

A proposed prototype implementation framework for census information system using geospatial data is shown in figure 8.

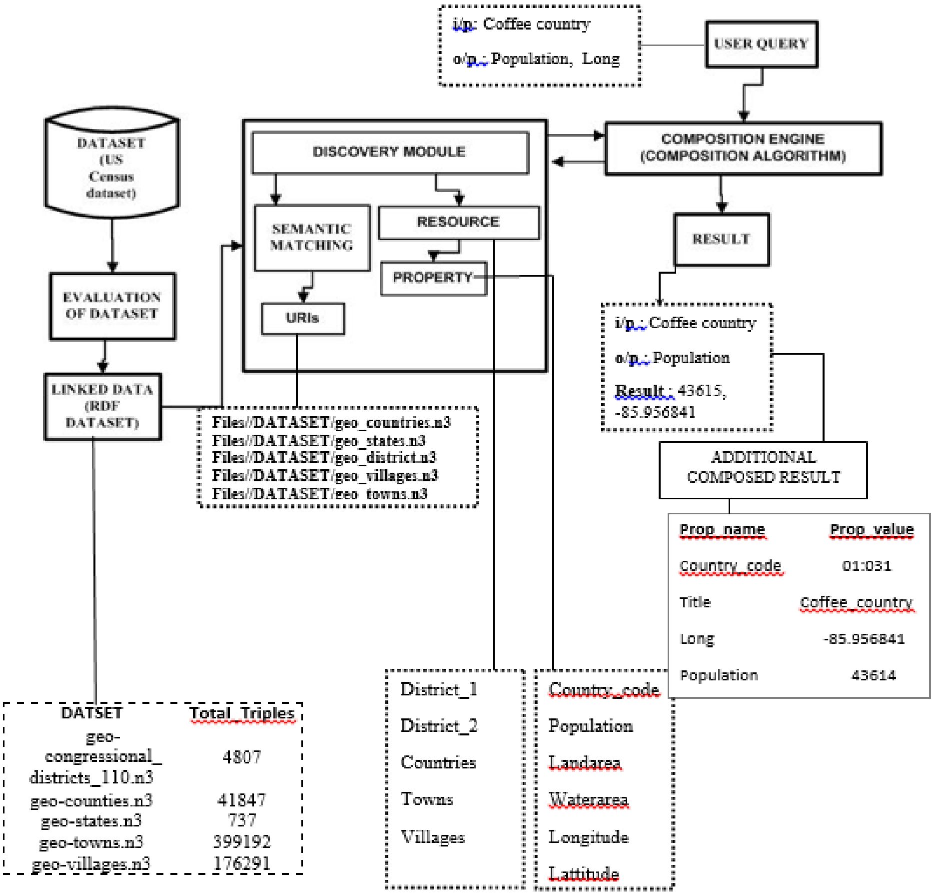

Figure 5. A Prototype for Census Information System

In the Experimental implementation, we have considered the **U.S. Census dataset** which describes rich information about the web services related to population statistics with semantic annotation. The dataset is provided as Linked Data which contains One billion triples of population statistics and basic geospatial data from the 2000 U.S. Census in RDF format.

As per the RDF format, conceptual Resource, Property and Property_values are derived from the U.S. census dataset. As shown in figure, User's query (input and output parameters) is provided to the Discovery engine for searching the requested services through Composition Engine. For example, our input query is "Coffee_country" and output result is "Population" and "Longitude".

Discovery engine performs semantic match based on the user request and the semantic concepts retrieved from data store. As a result, discovery engine provides the list of base URIs of resources for composition process.The list of base URIs of discovery result is given to the composition engine which generates the composed result demanded by the user.

4.2 Experimental Setup

For the implementation we have utilized resources as specified here: Operating System :Windows XP, Framework: Microsoft .Net Framework 4.0, RAM :2.5 GB, Software Tool: Microsoft Visual Studio 2010, Processor: Intel Core 2 Duo, Space Requirement:5 GB.

We have implemented our proposed work for discovery and composition and it's working is based on user requested query. This application is implemented using .Net framework 4.0 and Open Source .Net Library [27] on .Net platform which provides an easy and powerful for working with RDF, SPARQL and the Semantic Web.

dotNetRDF[27] provides toolkit for the user which in turn provides the way to work with single RDF file which contains RDF Editor, SparqlGUI and so on. It is also provides an open source SemWeb.NET library for programming purpose.

4.3 Dataset

In the Experimental implementation, we have considered the U.S. Census dataset [28] which describes rich information about the web services with semantic knowledge concept. This dataset describes setting up the dataset as Linked Data which contains One billion triples of population statistics and basic geospatial data from the 2000 U.S. Census as RDF format.

4.3.1 RDF

"The Resource Description Framework (RDF) [29] is a W3C standard for describing Web resources, such as the title, author, modification date, content, and copyright information of a Web page [12]". RDF data which contains triple stores where we have to extract or retrieve and manipulate the resources using SPARQL query. A RDF data model for US Census dataset in n3 format is shown in figure 6 as follows.

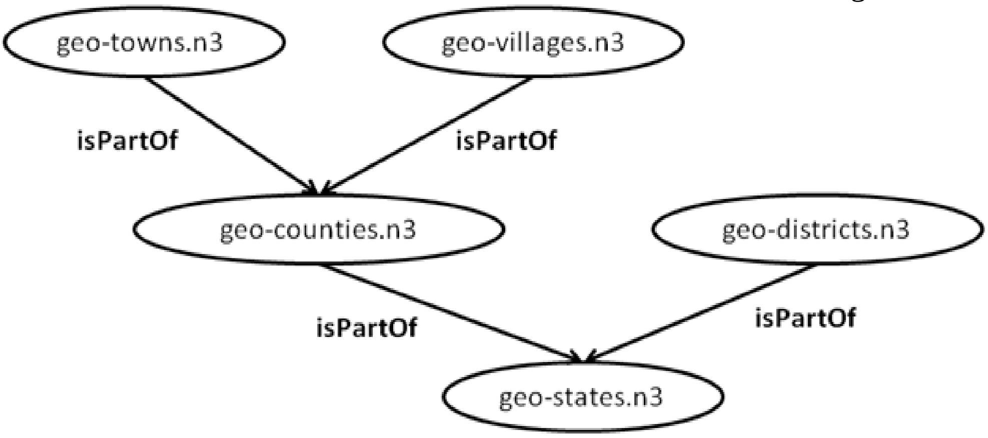

Figure 6. RDF data model

4.4 Experimental Results

This next section focuses on detail about dataset and various steps to be performed to show the applicability of the proposed work.

4.4.1 RDF data extraction from Dataset

The results shown in table 1 are extracted from the U.S. Census dataset, total number of nodes are extracted in the triple format. Based on that conceptual Resource, Property and Propertyvalue, results are derived for each RDF file in U.S. Census dataset. Base URIs of the resources are retrieved by performing semantic matchmaking in the discovery operation.

Dataset File Name	Total Triples	Subject (Resource) Nodes	Predicate (Property) Nodes	Objects (Property Value) Nodes	All Node Extraction Time(ms)
geo-congressional_districts_110.n3	4807	489	11	3335	102
geo-counties.n3	41847	3271	13	30894	281
geo-states.n3	737	53	14	610	30
geo-towns.n3	399192	39438	11	247665	640
geo-villages.n3	176291	26256	11	116138	542
TOTAL	**622874**	**69507**	**60**	**398642**	**1595**

Table 1. Extracted data from the data set

3. User Interface of Discovery and Composition Process

Following section shows the user interfaces developed for the service discovery and service composition operations.

3.1 Discovery Interface

Figure 7 presents user interface to perform discovery operation and to show the generated results as an output. It provides extraction process of RDF dataset based on user's request.

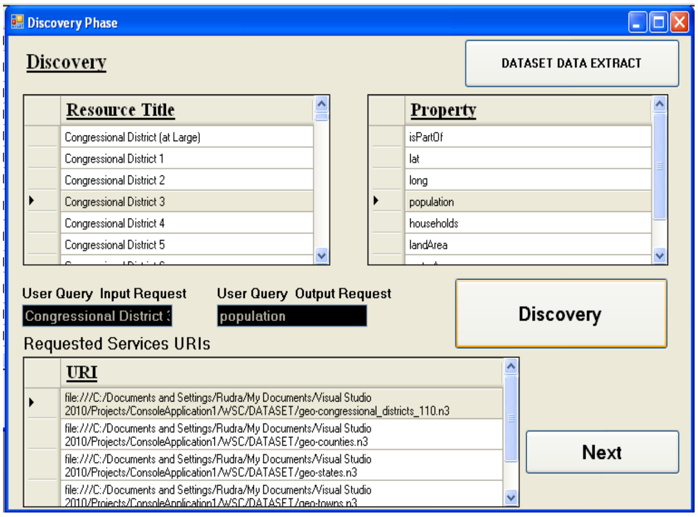

Figure 7. Result of Discovery Process

3.2 Composition Interface

Figure 8 presents user interface to show the generated results of composition process while figure 13 shows the composite result on the map.

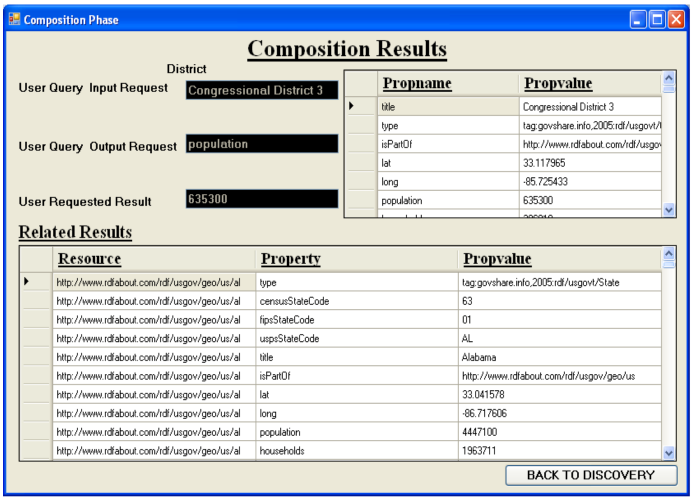

Figure 8. Result of Composition Process

4. Conclusion and Future Work

RESTful web services are gaining popularity over the SOAP web services because of lightweight nature and resource oriented design. Nowadays, RESTful web services is continuously increasing on the Web. Based on these considerations, we have developed framework for RESTful services discovery and composition using Linked Open Data based approach. We have demonstrated the experimental work with results for the feasibility of the work.

As a future work, we can integrate the service selection approach to refine the result of composition process. A prototype model for the healthcare domain could be developed to demonstrate the potential of the approach.

References

[1]. Yong-Ju Lee , “Algorithm for Automatic Web API Composition”, The First International Conference on Building and Exploring Web Based Environments,2013

[2] P. Nandhakumar, M. Hemalatha, Kashyap Dhruve “Adorning Web Search based on Ontology Semantics Spaces Provisioned through Web Services” .In International Journal of Computer Applications (0975 – 8887) Volume 55– No.7, 2012.

[3] M.Mahmoud, ElHafeez ,AhmedBadawy.“Active Knowledge: Dynamically Enriching RDF Knowledge Bases by Web Services”.In "SIGMOD, Indianapolis United States ,2010.

[4] M. H. Bohara, M. Mishra and S. Chaudhary, 'RESTful Web Service Integration Using Android Platform', Fourth International Conference on Computing Communication and Networking Technologies, Tiruchengode, Tamilnadu, India, 2013.
[5] Tarek M. Mahmoud,Tarek Abd-El-Hafeez,Ahmed Badawy" A Framework For An E-Learning System Based on Semantic Web," International Journal on Computer Science and Engineering (IJCSE).
[6] LucaPanziera,FlavioDePaoli."Framework for Self-descriptive RESTful Services",Fourth International Workshop on RESTful Design (WS-REST 2013),May 2013.
[7] A.M.Abirami, P.Sheba Alice, Dr. A.Askarunisa "An Enhanced Method for the Efficient Information Retrieval from Resume Documents using SPARQL", Semantic Web Interoperability, Usability, Applicability an IOS Press Journal, Vol 4, No 1 (2012).
[8] Ying Zhang, Yao-Yi Chang, Pedro Szekely, Craig A. Knoblock "A Semantic Approach to Retrieving, Linking, and Integrating Heterogeneous Geospatial Data", International Joint Conference on Artificial Intelligence (*IJCAI), 2013*.
[9] Haibo Zhao, Prashant Doshi, "Towards Automated RESTful Web Service Composition," IEEE International Conference on Web Services, 2009.
[10] Christian Bizer, Tom Heath, Tim Berners-Lee, "Linked Data - The Story So Far", in International Journal on Semantic Web and Information Systems IJSWIS, 2009.
[11] Frank Manola, Eric Miller, Brian McBride, "RDF Primer", W3C Recommendation, 10 February 2004.
[12] Deborah L. McGuinness, Frank van Harmelen, "OWL Web Ontology Language: Overview", W3C Recommendation, 10 February 2004.
[13] W3C OWL Working Group, "OWL 2 Web Ontology Language: Document Overview", W3C Recommendation, 27 October 2009.
[14] Dan Brickley, R.V. Guha, Brian McBride, "RDF Vocabulary Description Language 1.0: RDF Schema", W3C Recommendation, 10 February 2004.
[15] Krzysztof Janowicz, Martin Raubal and Werner Kuhn."The semantics of similarity in geographic information Retrieval", Journal of Spatial Information Science, 2011.
[16] Nancy Wiegand, Cassandra García. 2007. "A task-based ontology approach to automate geospatial data retrieval". Transactions in GIS. 2007.
[17] Peng Yue, Liping Di, Wenli Yang, Genong Yu, Peisheng Zhao.2007."Semantics-based automatic composition of geospatial web service chains". Computers and Geosciences, 2007.
[18] Y. J. Lee and J. H. Kim, "Semantically Enabled Data Mashups using Ontology Learning Method for Web APIs," Proceedings of the 2012 Computing, Communications and Applications Conference, 2012.
[19] R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases," Proceedings of the 1993 ACM-SIGMOD International Conference Management of Data, 1993.
[20] M. Paolucci, T. Kawamura, T. R. Payne, and K. Sycara, "Semantic Matching of Web Services Capabilites," Proceedings of the International Semantic Web Conference (ISWC), 2002.
[21] K. Kona, A. Bansal, M. Blake, and G. Gupta, "Generalized Semantics-based Service Composition," Proceedings of the IEEE International Conference on Web Services (ICWS), 2008.
[22] E. Sirin, B. Parsia, D. Wu, J. Hendler, and D. Nau, "HTN Planning for Web Service Composition using SHOP2", Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 1, No. 4, pp. 377-396, 2004.
[23] P. Rodriguez-Mier, M. Mucientes, and M. Lama, "Automatic Web Service Composition with a Heuristic-based Search Algorithm", Proceedings of the International Semantic Web Conference (ISWC), 2011.
[24] M. Shiaa, J. Fladmark, and B. Thiell, "An Incremental Graph-based Approach to Automatic Service Composition", Proceedings of the International Semantic Web Conference (ISWC), 2008.
[25] Zheng Li, Liam O'Brien. "Towards Effort Estimation for Web Service Compositions using Classification Matrix," the Fifth International Conference on Internet and Web Applications and Services (ICIW 2010), IEEE Computer Society, June 2010.
[26] Gurdas Singh, Dr. Jaiteg Singh. "Querying RDF Data: Methods and Issues." Indian Journal Of Applied Research, Volume:3, Issue : 2 ,February 2013.
[27] dotNetRDF an open source C# . NET library for RDF, Available at http://www.dotnetrdf.org. Retrieved on 01-09-2015.
[28] 2000 U.S. Census in RDF (rdfabout.com) http://datahub.io/dataset/2000-us-census-rdf. Retrieved on 01 09-2015..
[29] Box, Don, David Ehnebuske, Gopal Kakivaya, Andrew Layman, Noah Mendelsohn, Henrik Frystyk Nielsen, Satish Thatte, and Dave Winer. "Simple object access protocol (SOAP) 1.1." (2000).
[30] Christensen, Erik, Francisco Curbera, Greg Meredith, and Sanjiva Weerawarana. "Web services description language (WSDL) 1.1." (2001).
[31]Bellwood, Tom, Luc Clément, David Ehnebuske, Andrew Hately, Maryann Hondo, Yin Leng Husband, Karsten Januszewski et al. "UDDI Version 3.0."Published specification, Oasis 5 (2002): 16-18.
[32]Fielding, Roy. "Representational state transfer." Architectural Styles and the Design of Netowork-based Software Architecture (2000): 76-85.
[33]Fielding, Roy, Jim Gettys, Jeffrey Mogul, Henrik Frystyk, Larry Masinter, Paul Leach, and Tim Berners-Lee. "Hypertext transfer protocol–HTTP/1.1." (1999): 12.
[34] Berners-Lee, Tim, James Hendler, and Ora Lassila. "The semantic web."Scientific american 284, no. 5 (2001): 28-37.
[35] Yu, Liyang. "Linked open data." In A Developer's Guide to the Semantic Web, pp. 409-466. Springer Berlin Heidelberg, 2011.
[36] Page, Kevin R., David C. De Roure, and Kirk Martinez. "REST and Linked Data: a match made for domain driven development?." In Proceedings of the Second International Workshop on RESTful Design, pp. 22-25. ACM, 2011.