A Web Service with Self-Learning Ontological Process (WS – SLOP)

J. Ranjani
PG Scholar, SRM University, Kattankulathur & AP, Department of MCA, Sri Sai Ram Engineering College, Chennai

G. Niranjana
AP-SG, Department of CSE, SRM University, Kattankulathur, Chennai

Abstract - Ontology is a specification that is designed for reuse across multiple applications and implementations. Ontology has become the effective modeling for various applications and significantly in the semantic web. The difficulty of extracting information from the web is avoided by the ontology. It was created mainly for visualising information, has driven the birth of the semantic web. It contains much more resources than the web and will attach machine-readable semantic information to these resources. Self-Learning Ontological Process on a set of predefined sources must address the problem of various, largely unrelated concepts. The web services consist of basically two components, Web Services Description Language (WSDL) descriptor and free text descriptors. The WSDL descriptor is evaluated using two methods, namely Term Frequency/Inverse Document Frequency (TF/IDF) and web context generation. The proposed Self-Learning Ontological Process integrates TF/IDF and web context generation and applies validation using the free text descriptor service, so that, it offers more accurate definition of ontology. The web service ranking model assign the rank for a collection of web service documents which leads to the self learning, enhancement ontology for web services.

Keywords: Ontology, Semantic web, WSDL Descriptors, Free Text Descriptors, Web Context Generation, TF/IDF, Web Context Generation, Web Service, UDDI

I. INTRODUCTION

Ontology defines a common terms for researchers who need to share information in a domain. It includes machine understandable definitions of basic concepts in the various field and relations among them.

Ontology can be a

- "Theory of existence"'.
- An explicit pattern of conceptualization.
- A body of knowledge describing some domain.

Requirement to build up the ontology

- To distribute familiar understanding of the structure of information among people or software agents
- To permit reuse of field information
- To make field assumptions unambiguous
- To divide field information from the operational information
- To examine field knowledge

To distribute familiar understanding of the structure of information among people or software agents is one of the more universal goals in constructing ontology. For example, suppose numerous diverse Web sites contain product information or provide product e-commerce services. If these Web sites share and distribute the similar basic ontology of the conditions they all use, then computer agents can extort and cumulate information from these different sites. The agents can use this cumulative information to answer user queries or as input data to other applications.

Permitting reuse of field information was one of the dynamic services at the back new flow in ontology research. If one set of researchers develops such ontology in detail, others can simply use again it for their domains. If we need to construct a big ontology, we can incorporate several existing ontology, describing portions of the huge field. We can also use again a general ontology extend it to describe our domain of interest.

Making explicit field assumptions unambiguous underlying an accomplishment makes it possible to change these assumptions easily if our information about the field changes. An explicit specification of domain information is useful for novel users who must study what terms in the field stand for.

Dividing field information from the operational information is another common use of ontology. We can explain a work of configuring manufactured goods from its mechanism according to a necessary requirement and execute a program that does this configuration independent of the goods and components themselves.

Analyzing domain knowledge is possible once a declarative pattern of the terms is available. Prescribed assessment of terms is extremely expensive when mutually attempting to reclaim accessible ontology and extending them.

Implementing ontology is analogous to defining a set of statistics and their structure for other programs to use. Problem-solving methods, field self-determining applications, and software agents use ontology's and information bases built from ontology as facts.

Self-Learning Ontology, which has newly emerged as the vital technology for ontology building, involves routine recognition of concepts significant to a field and relations between the concepts.

Web service ontology

A major prerequisite for web service ontology is high quality. In particular, the ontology should provide a elevated modeling self-expression so that a huge diversity of services can be modeled. Understandable semantics and a rich formalization of these semantics would ensure the maintenance for difficult reasoning tasks performed by services. Then, another appraise of quality is whether the captured generic information is flexible for use in parallel domains. It would also be desirable that a mapping between different evolving generic ontology's can be achieved in order to ensure a basis for their comparison. Ontology for web services should present large domain treatment and they should include the information that describes large and dynamic web service collections.

- Improves the accurateness of web searches by penetrating for concepts as a substitute of keywords.
- Allows systems that were autonomously urbanized to work mutually to exchange information.
- Facilitates the use of services to collect, process, and exchange information.
- Helps attempt problematical questions whose answers do not exist in on a single web service.

WSDL and UDDI

Self learning ontologies focused only on a few domains previously. It develops from the existing ontology only. UDDI entries improve interoperability, approval and adoption of web services. But these UDDI entries have major drawbacks like, they are visible and available to all or they may have irrerelevant entries or if a user wants to access a service, they require registration.

UDDI registries can store only limited description about the service. Self learning ontologies can be used as an alternative solution for classifying and utilizing the services. However, the growing number of accessible web services makes it difficult to categorize web services using single domain ontology or a set of existing ontologies created for other purposes.

The proposed Web service Self-Learning Ontological Process is based on the two types of descriptions:

- Web Service Description Language (WSDL) - describes "how" the service should be used and
- A textual description of the web service in free text - describes "what" the service does.

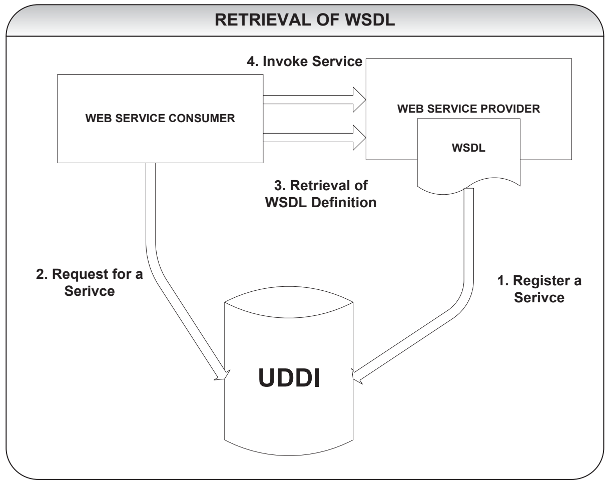

Figure 1. Service Consumer Accessing WSDL Document

Self Learning Ontology process analyzes a web service using TF/IDF, web context extraction and free text descriptor verification. These three methods have different perspective on a web service. Since these methods have different perspective of a web service, these processes provides more precise description of the ontology and yields good results.

- TF/IDF – Examines the service from internal point of view. It examines what concept in the text illustrates the content of the WSDL document.
- Web Context Extraction – Most common and frequent concept represents the response of the web search queries based on WSDL content, i.e., it illustrates the WSDL document from an external point of view.
- Free Text Description Verification method - Used to resolve inconsistencies with the current ontology.

An ontology can be bootstrapped when all these examinations agrees the new concept or a change in relationship between the ontology concepts. Descriptors are used to define the relationship between the ontologies. SLOP can assist in ontology construction and reduce the maintenance effort significantly. SLOP enhances automatic structuring of an ontology that can assist in, classifying, expanding, and retrieving related services, without any prior guidance required by previously developed approaches.

II. ARCHITECTURE OF SELF-LEARNING ONTOLOGY PROCESS

The proposed self learning ontology model is based on the continuous analysis of WSDL documents and generates a self learning ontology model which uses relationships among the concepts.

The innovation of the proposed self learning model centers on:

- The combination of the use of two different extraction methods
  - TF/IDF
  - Web based concept generation
- Analysis of external service descriptors can be done by verifying the results using of free text descriptors
- Ontology construction
- Web service ranking

The following figure describes how to construct a self-learning ontology from WSDL document.

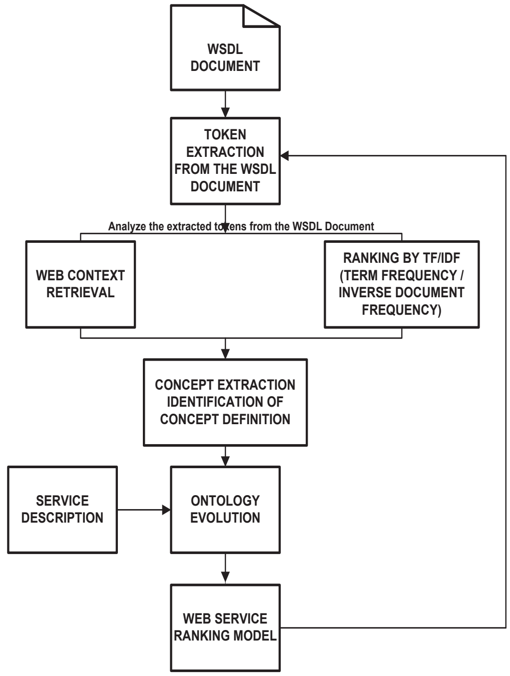

Figure 2. Architecture of a Web service Self Learning ontological process.

First the WSDL document is registered with UDDI registry. Tokens are extracted from WSDL using TF/IDF analysis. Rank is assigned for WSDL document based on the frequency or number of hits. Web contents are also extracted from the WSDL document.

The new concept will be extracted from TF/IDF analysis and web context extraction. Now ontology is evolved from the extracted content and it refers service description of the document. Finally, ranking model for a web service is provided for the web service consumer.

III. OPERATION OF SLOP

The overall self Learning ontology process is described in Fig. 2. There are four five steps in the process.

Token extraction

This extract tokens representing relevant information from a WSDL document. This step extracts all the name labels, parses the tokens, and performs initial filtering.

Analyze the extracted WSDL tokens in parallel

The second step analyzes the extracted WSDL tokens using two methods.

- The frequency of tokens appearing in WSDL documents are analyzed using TF/IDF analysis
- Token set can be used a query for web context extraction
- The result will be clustered according to the textual descriptors which identifies the perspective of the web service

Concept Extraction

TF/IDF and web context method identifies the descriptors for the concept extraction. These descriptors are very useful to identify the concept names that could be used in the ontology evolution step. Also, these descriptors are also assisted in the process of finding relationship between the concepts.

Concept Evocation

Descriptor appears in both TF/IDF and web context are identified in the concept extraction step. Possible concept names utilized by the ontology evolution process can be identified by these descriptors. These descriptors also assist in the process of finding the relationship between concepts.

Ontology Evolution

Ontology evolution step develops the required ontology that is identified concept extraction step. It can also modify the relationship between them. External descriptor acts as a moderator for any conflict appears between the new concept and ontology. Free text descriptors are used to cross check the correct implication of the idea in the new concept. According to the most common context descriptors among concepts, the relations are defined as a continuing process.

Web Service ranking model

Since ranking adaption is a challenging task for the web consumer, a model has been adapted which predicts ranking for the cluster of web service documents. According to the priority most frequently visited web services are categorized. The entire process will be continued after ontology evolution for the next web service document with their concepts and relations. The processing order the web service document may be in any arbitrary order.

3.1 Token extraction

The major elements of a web service docu0ment are types, operations and binding. These elements are broken down into identifiable elements and it can be reused or combined effectively.

Steps to extract tokens

1. Represent the descriptor which contains set of tokens for each service
2. Parse the documentation of a service to extract the textual term of the token
3. Descriptor signifies the web service document
4. Web service tokens are the sequence of tokens that can be identified by capitalizing the first letter of each word in the sequence

Example: GetDomainsByRegistrantName

The descriptors are divided into separate tokens. It is worth mentioning that we initially considered using predefined WSDL documentation tags for extraction and evaluation but found them less valuable since web service developers usually do not include tags in their services. The tokens are filtered using a list of stop words, removing words with no substantive semantics

Analyzes the extracted WSDL tokens in parallel

The processes included in context analysis are TF/IDF analysis and Web Context extraction. In information retrieval, TF/IDF analysis is commonly used to generate a set of keywords from a large amount of documents. The same analysis is applied in WSDL descriptors also to find the tokens. Generating a huge volume for each and every document and the unique unwanted terms are thrown out from the corpus.

The TF/IDF process analyzes the most and less frequent terms appeared in each WSDL. The frequency is a very useful measure that evaluates the importance of the term in the document. Using that frequency only a document can be classified according to the domain. This method is applied for the WSDL descriptors. A term is identified as a least frequency term that has more number of occurrences in both document and in the corpus also.
Set of tokens identified in TF/IDF process can be used a query for a search engine, group the results according to the textual descriptors and clusters the set of descriptors that identifies the web service context. This process is developed by extracting tokens from the WSDL descriptor of each service. Every set of token will be sent to the web search engine and descriptor set is generated by clustering the search results of each web page for each set of tokens.

3.3 Perception Derivation

Concept derivation identifies a possible concept definition that will be refined next in the ontology evolution. The idea derivation is performed based on situation intersection. An ontology perception is defined by the descriptors that come into view in the connection of both the web perspective results and the TF/IDF results.

Descriptor set is derived from the TF/IDF results based on extracted tokens from the WSDL text. The context is to begin with clear as a descriptor set extracted from the web and representing the same text. As a result, the ontology concept is represented by a set of descriptor which belongs to both sets.

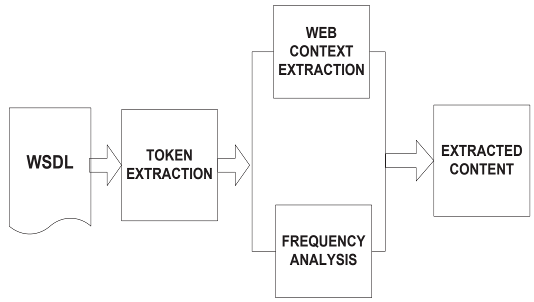

Figure 3. Content Extraction

The context detection algorithm contains the following major phases:

- Selecting contexts for each set of tokens,
- grade the contexts, and
- Declaring the present contexts.

The result of the token mining is a record of tokens obtained from the web service WSDL. The input to the algorithm is based on the given name descriptor tokens extracted from the web service WSDL. The option of the context descriptors is based on probing the web for relevant documents according to these tokens and on combining the results into feasible context descriptors. The production of the position phase is a set of highest ranking context descriptors. The set of context descriptors that have the peak number of references, both in number of web pages and in number of appearances in the WSDL, is declared to be the context and the credence is defined by integrating the value of references and appearances.

3.4 Ontology Evolution

The ontology development expands the ontology as necessary according to the newly recognized concepts and modifies the relationships between them. The external web service textual descriptor serves as a moderator if there is a conflict between the current ontology and a new concept.

Latest concepts can be tartan beside the free text descriptors to verify the correct explanation of the concept. After the ontology evolution, the whole procedure continues to the next WSDL with the evolved ontology concepts and relations.

Ontology evolution process

- Construction of a latest concept based on refinement the possible identified concepts.
- Evolution of the relation between concepts, examine the overlapping context descriptors between diverse concepts. In this case, to use descriptors that were incorporated in the union of the descriptors extracted by both the TF/IDF and the Web context methods. Preference is given to descriptors that become visible in both concept definitions more descriptors that emerge in the context descriptors.

Web Service Ranking model

Web Service Ranking is moderately more demanding. It desires to adapt the model which is used to anticipate the rankings for a set of web service credentials. The web service documents are usually labeled with several significance levels, which seem to be handled by a multiclass taxonomy or regression, this process categorizes the web service queries based upon most correlated searched web service. Most visited web service is categorized based upon precedence.

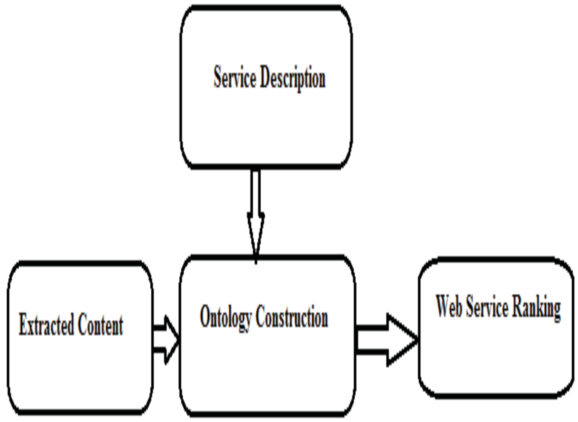

Figure 4. Web Service Ranking

IV. EXPECTED RESULT

The token extraction has been experimented as follows:

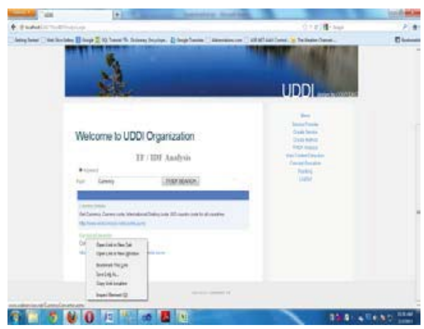

Figure 5. Analysis of TF/IDF

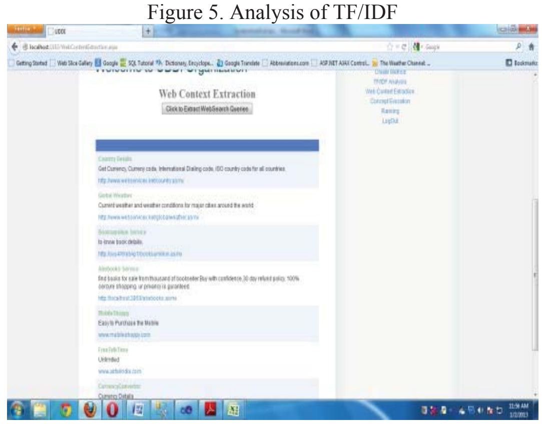

Figure 6. Generation of Web Context

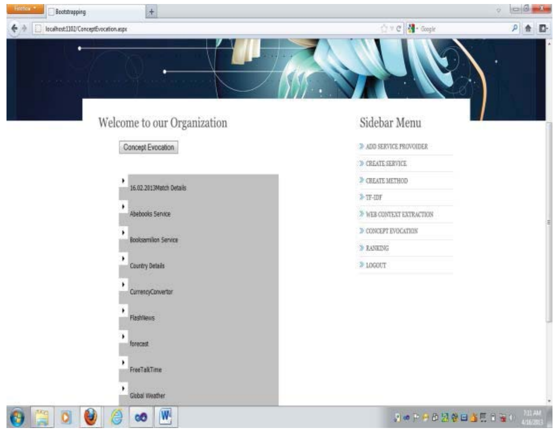

Figure 7. Ontology generation

V. CONCLUSION

This paper proposes an approach for self learning ontology based on web service descriptions. The approach analyzes web services from various perspectives. Web services usually consist of both WSDL and free text descriptors. The self learning ontology process is based on WSDL and verifying the process based on the web service free text descriptor.

Self learning Ontology is used to focus on unlimited domain so that any type of web service can be created, maintained and supplementary in the existing ontology. This come up to facilitates ontology production procedure which allows the automatic construction of an ontology that can help in expanding, classifying and retrieving appropriate services without any direction as a contrary to the existing methods.

The self learning ontology process in our model is performed automatically, enabling a stable update of the ontology for each new web service. Using web service ranking model, large number of frequently visited web service can be easily identified. Thus, the involuntary production, enrichment and adaptation of ontology, is highly preferred.

REFERENCES

[1] Wensheng Wu1, AnHai Doan1, Clement Yu2, and Weiyi Meng, “Bootstrapping Domain Ontology for Semantic Web Services from Source Web Sites”, Proceedings of the 6th international conference on Technologies for E-Services, January 2008

[2] Y. Chabeb, S. Tata, and D. Belad, “Toward an Integrated Ontology for Web Services,” Proc. Fourth Int’l Conf. Internet and Web Applications and Services (ICIW ’09), 2009.

[3] Adam Funk, Kalina Bontcheva, “Ontology-Based Categorization of Web Services with Machine Learning”, 7th international conference on Language Resources and Evaluation, May 2010

[4] Anita Mohebi, Master of Science, Ryerson University, “An Efficient Qos-Based Ranking Model for Web Service Selection with Consideration of User’s Requirement”, 2012

[5] Aviv Segev, Member, IEEE, and Quan Z. Sheng, Member, IEEE, “Bootstrapping Ontologies for Web Services”, IEEE Transactions On Services Computing, Vol. 5, No. 1, January-March 2012

[6] Dongkyu Kim Et. Al, “Practical Ontology Systems for Enterprise Application”, LNCS 3818, pp. 79 – 89, Springer-Verlag Berlin Heidelberg 2005

[7] Ehrig. M, Staab. S, and Sure. Y, “Bootstrapping Ontology Alignment Methods with APFEL”, Proc. Fourth Int’l Semantic Web Conf. (ISWC’05), 2005.

[8] Georgios Petasis, Anastasia Krithara, Alfio Ferrara, Vangelis Karkaletsis, “Bootstrapping Ontology Evolution: A Generic Approach Relying on Ontology-based Information Extraction”, May 2012.

[9] Natalya F. Noy, Michel Klein, “Ontology Evolution: Not the same as schema evolution”, Knowledge and Information Systems, Springer-Verlag, 2004.

[10] Wensheng Wu1, AnHai Doan1, Clement Yu2, and Weiyi Meng, “Bootstrapping Domain Ontology for Semantic Web Services from Source Web Sites”, Proceedings of the 6th international conference on Technologies for E-Services, January 2008

[11] G. Zhang, A. Troy, and K. Bourgoin, “Bootstrapping Ontology Learning for Information Retrieval Using Formal Concept Analysis and Information Anchors,” Proc. 14th Int’l Conf. Conceptual Structures (ICCS ’06), 2006.

[12] http://www.smhi.se/en/services/professional-services/agriculture/web-services-for-farmers-1.7819

[13] http://www.sungard.com/campaigns/fs/globaltrading/marketmap/real-timedata/webservices.aspx