# Semantic based Crawling for Automated Service Discovery and Categorization of Web Services in Digital Environment

Vaibhav V. Sawant
PG Research Scholar, Dept. of CSE, D. Y. Patil College of Engineering & Tech., Kolhapur (MH), India.

Dr. Vijay R. Ghorpade
Principal, D. Y. Patil College of Engineering & Tech., Kolhapur (MH), India.

**Abstract – A vast majority of web services exist without explicit associated semantic descriptions. As a result many services that are relevant to a specific user service request may not be considered during service discovery. In this paper, we address the issue of web service discovery given no explicit service description semantics that match a specific service request. Our approach to semantic based web service discovery involves semantic-based service categorization and semantic enhancement of the service request in digital environment. A solution for achieving functional level service categorization based on an ontology framework is proposed. Additionally, clustering is utilized for accurately classifying the web services based on service functionality. The semantic-based categorization is performed offline at the universal description discovery and integration (UDDI). The semantic enhancement of the service request achieves a better matching with relevant services. The service request enhancement involves expansion of additional terms (retrieved from ontology) that are deemed relevant for the requested functionality. In this paper semantic matching between content of downloaded web page and ontology is used to guide the crawler towards relevant information. The experimental results validate the effectiveness and feasibility of the proposed approach. In order to thoroughly evaluate the performance of proposed crawling technique, measures such as harvest rate, precision, recall, Fallout rate will be calculated.**

**Index Terms – service categorization, web service discovery, semantic classification, digital environment, ontology structure**

## 1. INTRODUCTION

The popularity of World Wide Web (WWW) is largely dependent on the search engines. Search engines are the gateways to the huge information repository at the internet. Search engine consist of four discrete components: Crawling, Indexing, Ranking and query-processing. A web crawler is a application program that goes around the internet collecting and storing data in a database for further analysis [1] [2]. Proposed web crawler begins with a website (Uniform Resource Locator) URL, called seed. Multiple seeds can be handled by crawler and URL list can be crawled parallel. The crawler visits the URL at the top of the list. On the web page it looks for hyperlinks to other web pages, it adds them to the existing list of URLs in the list. This methodology of the crawler visiting URLs depends on the rules set for the crawler. Crawled data is analyzed for further improvement in service discovery with proposed modules.

A large number of digitally distributed applications on the Web services architecture facilitate the creation of service-oriented architecture in web environment [3]. These web services communicate among each others for data growth of e-commerce, marketing and offer various functionalities.. Some of the web services are published and implemented in-house by various organizations for improving digital environment services. These web services are used for professional applications, or may be in the government and military. However, this structure requires careful selection of appropriate web services [4]. The Semantic web is known for being a web of Semantic Web Documents (SWDs) those are freely available on the semantic web and are described in Resource Description Framework (RDF) or any other syntax of semantic web [5]. Service providers are within the registry of web services that are specified by the predefined categories. Consequently, similar services can be listed under various categories. The emergence of digital business can be attributed to the natural existence of business ecosystems, along with the evolution of business network and information technology [6].

Today's search engines deal with SWDs poorly, since they have been developed to process text documents. Most make no attempt to parse web documents into appropriate tokens and none take advantage of the structural and semantic information encoded in a SWD. This paper proposes an semantic based web crawler architecture for effective crawling, parsing, analyzing and classification of semantic data with help of ontology structures. Web services digital network are service provider termed as servers and services utilized by users through their browsers. But, to provide the web services it is currently facing the following problems,

- Service provider marks the semantic web Uniform Resources Identifiers (URI) which will use the language to describe.
- To get online service page instances is very difficult as information retrieving is very difficult from the existing pages and those information will not annotate as the semantic services.
- Some services providers will work based on the semantic annotation, domain knowledge; this information will provide very less useful information. And still there are no proper methods to do this work and the existing building principal retrieving process information and services is not ubiquitous.

To solve these problems of the existing systems, an automatic framework for discovering, annotatating and categorizing content data domainwise is proposed. The proposed system will resolve functionality of the existing crawler improving the efficiency of the crawler.

The rest of the paper is presented in below sections - 2. Related work 3. Proposed system architecture 4. Implementation details 5. Experiment and results. Finally, in section - 6 work is summarized and concluded with further work.

## 2. RELATED WORK

The existing approaches are focusing on any one of the process that can be a services requester or service provider [1, 2].

The service users may come across three major issues – heterogeneity, ubiquity, and ambiguity, when searching for mining service information over the Internet. A service provider enters a digital environment by publishing a service entity, which will be stored in distributed service knowledge bases. Here, these service entities are stored in the form of service metadata [3]. Digital ecosystems transcend the traditional defined, collaborative environments from centralized, distributed or hybrid models into an flexible domain cluster demand-driven interactive environment [4]. Metadata abstraction focused crawlers are the focused crawlers that can abstract meaningful information from relevant Web pages and annotate the information with ontology markup languages [5].Existing service discovery approaches often adopt keyword matching technologies to locate the published web services. This syntax-based matchmaking returns discovery results that may not accurately match the given service request. As a result, only a few services that are an exact syntactical match of the service request may be considered for selection [6]. Thus, the discovery process is also constrained by its dependence on human intervention for choosing the appropriate service based on its semantics [9, 10]. The Semantic Web provides domain-knowledge-based classification tools [11]. H. Dong [12] proposed an association metric, with the purpose of optimizing the order of visited URLs for web crawlers. Another study is to organize online documents by linking their URLs to hierarchical ontology concepts, which are seen as thematic subsets [14]. Consequently, similar services can be listed under various categories [10, 13].

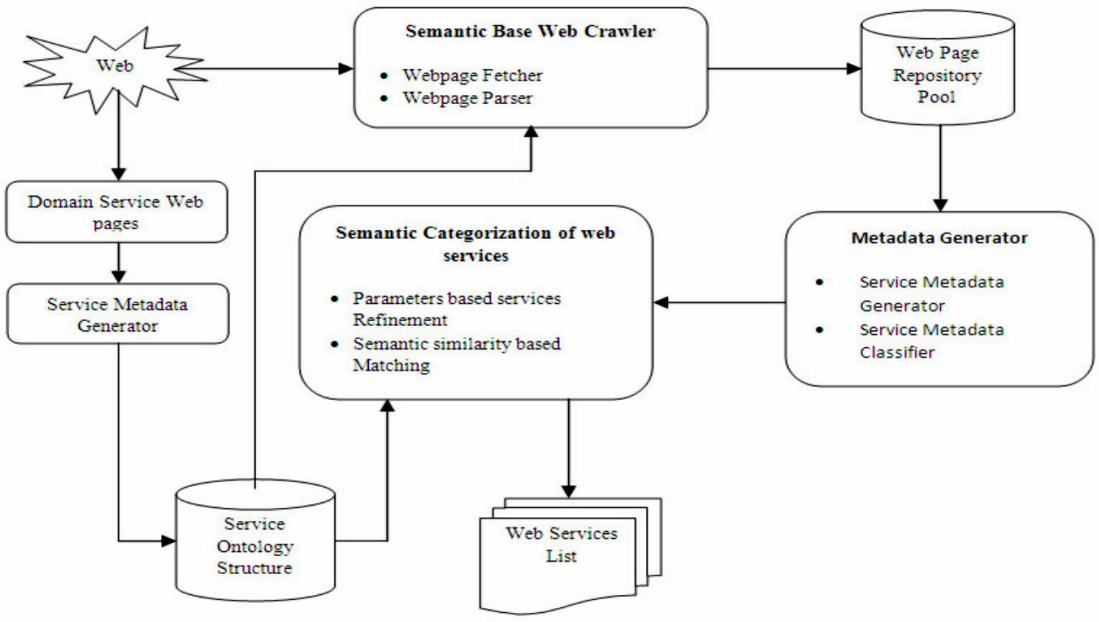


Figure.1: System Architecture

## 3. PROPOSED SYSTEM ARCHITECTURE

To overcome problems of focused crawler and automated discovery process discussed in the literature survey, a semantic based focused crawler is proposed for automated service discovery in digital web ecosystem. This crawler selects semantically similar web services using domain-knowledge in web ecosystems and improves the service retrieval result. The work is divided into following modules:

I. Implementation of semantics based crawler
II. Metadata generator & classification
III. Semantic categorization
IV. Performance evaluation

Figure 1 shows detailed architecture of proposed system. This system will work as under:

- The crawler will retrieve the information regarding service entities from the Web, which corresponds to the functionality of service discovery in digital ecosystems.
- The crawler will annotate the service information with the purpose of semantic and to store the semantic service information, which corresponds to the functionality of service annotation in digital ecosystems.
- The crawler will filter and classify the annotated service information by means of specific service domain knowledge, which corresponds to the functionality of service classification in digital ecosystems.

### 3.1 Implementation of Semantic web crawler

This module is based on semantic web crawler implementation where characteristics of domain based ontology and syntax based ontology is clubbed together and with use of depth first algorithm for crawling, websites are collected in repositories. This algorithm makes use of rule based association from WEKA libraries & travel through web documents for extracting meaningful data in HTML tags and collects in its repositories. The basic data flow diagram of this module is as follows:

3.1.1 Initialization: User initiates seed URL of websites to application server and specify the no. of WebPages to crawl, depth of crawling. Once this configuration has been completed the Policy Centre will send the depth (q) to the frontier for the web page crawling as shown in figure 4.

3.1.2 The webpage fetcher will start to obtain web pages in queue after it receives the URL. The web pages after crawling are sent to repository pool as depicted in figure 2.

3.1.3 When the policy centre receives the URLs from the webpage fetcher, it will determine whether they are within the crawling boundary & redundancy by comparing each document domain name. After that, the Policy Centre will allot doc id to the URL of website by their visiting priorities and send them to the pool as under:

a. The web page has hyperlinks to same page or other pages. Redirects to same page (inbounds) are omitted as they get in loop for crawler and those which redirects to another web pages (outbound) are considered in queue.

b. To formulate this, the web page is taken as a directed graph G= (V, E) where V is the set of nodes i.e. the set of all web pages and E is the set of directed edges in the graph i.e. hyperlinks. Let the total no. of pages on the web be n. The page rank formula for each page i denoted by p(i) is given by,

$$P(i) = \frac{1-d}{n} + d \sum_{Pj \varepsilon M(Pi)} \frac{Pi}{Pj}$$

Where d is dumping factor and for our case it is set to 0.85. The dumping factor is subtracted from 1. P(i) are the pages to consider from queue. M(Pi) is set of pages that link to, Pj is the no. of outbound links on page.

Once all web pages have been traversed then all its embedded tags will be removed and page will be saved in repository pool in plain texts. This pool consists of XML documents which include information snippets enclosed inside text tag and its parent URL.

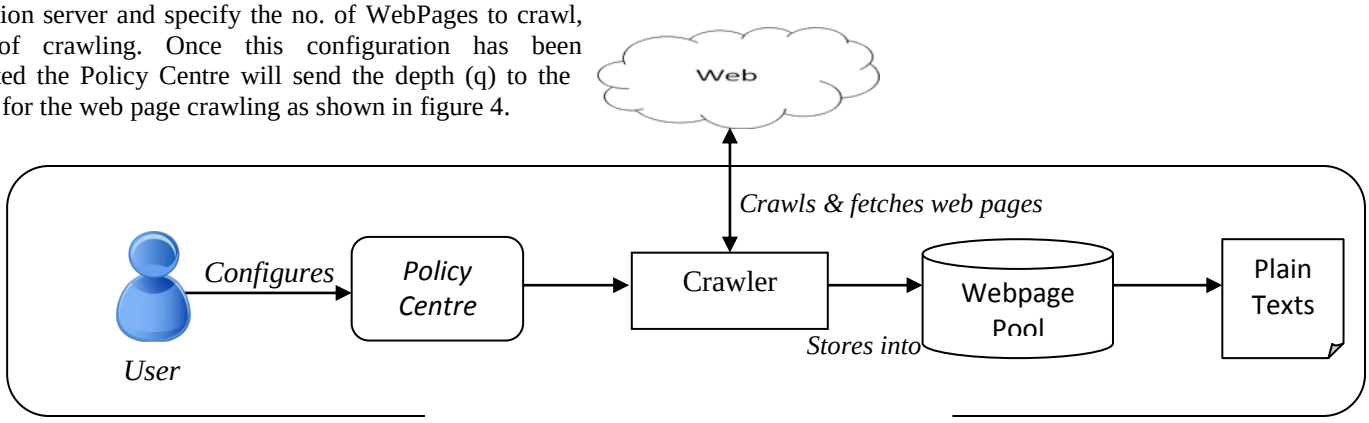


Figure 2: Flow diagram of module 1

3.1.1.1 Working of Crawler

User inputs seed URL for each thread crawler. A multithreaded crawler is implemented so that parallely multiple websites can be crawled and the web pages would be downloaded fast for further processing as shown in figure 3. Frontier manager holds a set of visited and waiting to crawl URL's. There should be some thread responsible for crawling web pages. So multithreaded crawler spawns a new thread all the time we want to crawl the website until the list is empty. Also we have maintained a list of already downloaded pages in persistent storage and checked before adding to the queue. If the page is already downloaded then its been cross checked once again with its metadata like file size, contents and its freshness then whether to crawl it again or not is been decided by the thread at run time. By default there is no limit on the depth of crawling. But we have limited the depth of crawling. For example, assume that you have a seed page "A", which links to "B", which links to "C", which links to "D". So, we have the following link structure:

A –> B –> C –> D

Since, "A" is a seed page; it will have a depth of 0. "B" will have depth of 1 and so on. You can set a limit on the depth of pages that crawler crawls. For example, if you set this limit to 2, it won't crawl page "D".

3.2 Metadata generation using semantic web crawler

This module is used to generate metadata with help of semantic web crawler. This module works as follows:

3.2.1 The webpage parser will obtain the processed web page information from the Webpage pool, extract the meaningful information snippets from each web page and pass them to the service metadata generator.

3.2.2 The service metadata generator will annotate the delivered information snippets with the ontology markup languages, in order to create service metadata. Service metadata will then be stored into the service metadata local database. Service metadata generator will send a message of creation to the service metadata classifier.

3.2.3 On receiving the message the service metadata classifier will compute the similarities between the generated metadata and each bottom-level ontology concept of a compatible ontology. If a similarity is above a threshold value, the corresponding concept can be regarded as being relevant to the metadata. Then, service metadata generator will associate metadata with the concepts. If similarity is below threshold those are treated as irrelevant to metadata.

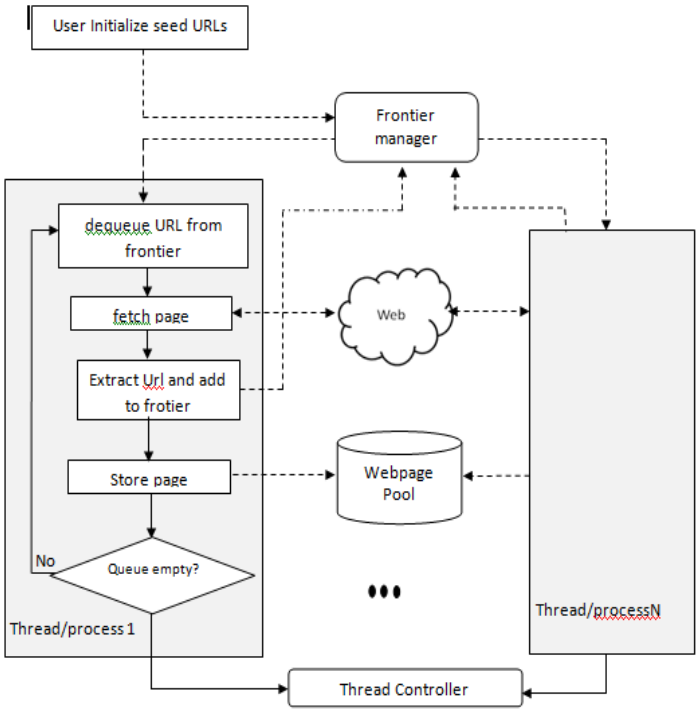


Figure 3: Multithreaded crawler

After we get the web repository pages, then pre-processing and metadata generation is applied with below algorithm :

```
for each document in webdb_repository
do
    Remove stop-words
end for
for each remaining word in the dataset
do
 Perform Stemming using Lovins stemmer and store in a
 vector (WordList)
end for
for each word in the Wordlist
do
       Calculate TF and store the result in a weight matrix
end for
for each element in weight matrix Set the threshold 'c'
     for each term
       If concept < c then
                discard the term along with its weight
       end if
     End for
End for
```

Threshold in our case is set to 0.8 means semantic similarity is 80% between the words. It is important to select the significant keywords that carry the meaning and discard the words that do not contribute to distinguishing between the documents.

Term frequency of each word in a document (TF) is a weight which depends on the distribution of each word in documents. TF matrix generation is done by using:

$$TermFrequency(TF) = \frac{\text{No. of times term t appears in a document}}{\text{Total number of terms in the document}}$$

### 3.3 Semantic categorization

The classification and annotation service will work based on the user query. It will get the Metadata from the Services Metadata database and classify the results of web pages, means the user request resources will be matching the fetched pages data and each page data will be annotated on the user request and provide the rank to the pages and those page links will provide response to the user. It process the collected metadata from the metadata generator and metadata from service ontology structure data to compare the semantic similarity of data and identifying the services category and will provide an web services list as output.

Check similarity between the semantic web services of vectors. Algorithm for similarity checking of semantic web service

```
Checksimilarity(List_of_webservices)
Begin
Get List_of_webservice {<SWS1, SDV1>, <SWS2,
 SDV2>  ..... < SWSn , SDVn>}
For each tuple do the following process

If ( SWS1.name = SWS2.name) then
   If (SWS1.Description=SWS2.Description) then
       begin
       For each input( Ii )of SWS1.inputs
       For each input( Ij )of SWS2.inputs
      Compare_Input( Ii , Ij )
      For each Output( Oi )of SWS1.outputs
      For each Output( oj )of SWS2.outputs
      Compare_output( oi , oj )
      Add the similar web services into a group
   End
End
```

Form this clusters of similar service categories will be mapped and respected category will be referred each time. Proposed work gives two fold benefits; firstly, only semantically similar results are retrieved which reduces the number of results extracted and secondly, due to improved focused searching irrelevant results is pruned.

### 3.4 Result evaluation

For ontology generation WEKA libraries are used and required packages are embedded with swing components. Threshold is set which is 0.85 & 100 websites have been crawled so far for testing. one of the characteristic of crawler is freshness and for our proposed crawler it is tested to be correct. Redundancy is also eliminated with caching logic within crawler i.e. crawler will not crawl the same webpage unless it is fresh.

In order to evaluate the performance of our proposed semantic focused crawlers, 4 even indicators from the field of information retrieval are undertaken: harvest rate, precision, recall and fallout rate.

3.4.1 Harvest Rate: Harvest Rate in the information retrieval is to measure the crawling ability of a crawler, which can be mathematically represented as,

$$\text{Harvest Rate} = \frac{\text{Number of associated metadata}}{\text{Number of generated metadata}}$$

3.4.2 Precision: Precision in the information retrieval is used to measure the preciseness of a retrieval system.

3.4.3 Recall: Recall in the information retrieval refers to the measure of effectiveness of a query system.

3.4.4 Fallout Rate: Fallout Rate is considered for calculation of non-relevant concept associated by whole collection of irrelevant metadata.

$$\text{Fallout Rate} = \frac{\text{Number of associated and non relevant metadata}}{\text{Number of nonrelevant metadata}}$$

## 4. EXPERIMENTAL SET-UP & RESULTS

An application (figure 4) is developed to carry all execution tasks of specified modules. The application server tool has been developed at local side that will allow authenticated users to use it. When user logins & enters the absolute URL of website, whose robots allow to crawl their website, then those web pages are traversed, fetched and downloaded in local disk through the interface engine. The interface is also used to display the extracted web page data enclosed in HTML tags as per the user defined depth. The collected web docs are separated in tabs so that all its crawled information can be viewed.

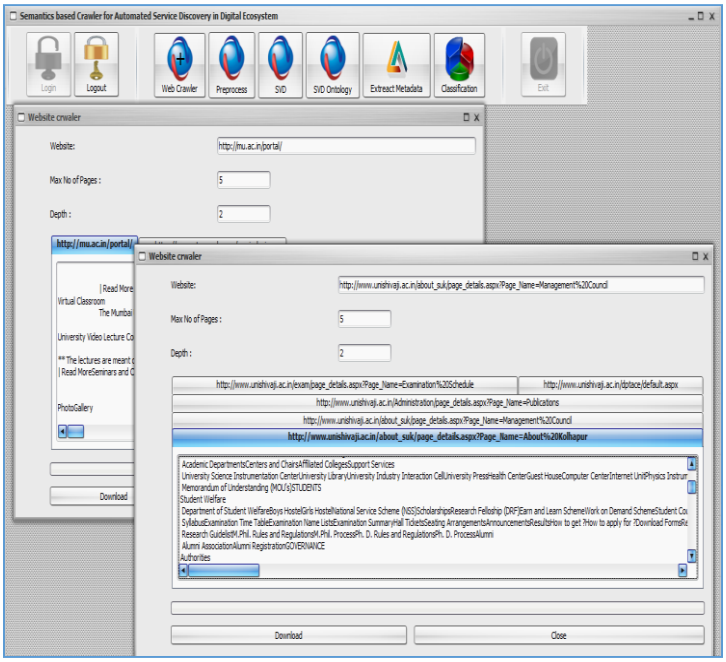


Figure 4: Policy configuration & multihreaded crawling

Nextly, this data is pre-processed and ontology is generated as under shown in figure 5,

```
1115:salvia:herb:0.39669370651245117
1116:elms.org.uk:elms.org.uk:-2.0
1117:hhwtravel:hhwtravel:-2.0
1118:yw:ing:0.2822052538394928
1119:in/148932728496725/plac:search/106078429431815/plac:1.0
1120 Replace Word :in/148932728496725/plac:search/106078429431815/plac:1.0
1120:eglur.co.uk:eglur.co.uk:-2.0
1121:garfieldweston.org:garfieldweston.org:-2.0
1122:categori:categori:-2.0
1123:recent:follow:0.4248262345790863
1124 Replace Word :recent:follow:0.4248262345790863
1124:host:improv:0.34996622800827026
1125:transport:trade:0.3415008783340454
1126:ashtrai:ashtrai:-2.0
1127:beeciifolkd:beeciifolkd:-2.0
```

Figure 5: Ontology generated metadata

## 5. CONCLUSION & FURTHER WORK

This paper presents a framework for automatic service discovery in a huge digital web environment. It proposes an efficient approach to build a service ontology structure and classifying the web service into category for user service improvisation. In order to achieve the goal of semantic service crawling, discovery and classification this paper presents a framework model consisting of a semantic based web crawler, semantic classification and categorization of web services for experiment. Main features of the proposed work is to semantically discover the service information from the web pages by parsing, annotating and storing their service information, which will be used for classification & categorization of the web service based on specific service ontology domain knowledge. This approach defines a format for service metadata and service concept, which enables the function of similarity computation and association between metadata and concepts [15].

## REFERENCES

[1] Gupta, P.; Johari, K., "Implementation of Web Crawler", Emerging Trends in Engineering and Technology (ICETET), 2009 2nd International Conference, vol., no., pp.838,843, 16-18 Dec. 2009 doi: 10.1109/ICETET.2009.124.

[2] Udapure, T., Kale, R., & Dharmik, R., "Study of Web Crawler and its Different Types", IOSR Journal of Computer Engineering, 16(1), 2014.

[3] Aabhas V. Paliwal, Basit Shaafiq, Jaideep Vaidya, Hui Xiong, Nabil Adam, members IEEE, "Semantics-based automated service discovery", IEEE Transactions On Services Computing, Vol. 5, No. 2, June 2012, pp. 260-275.

[4] H. Boley and E. Chang, "Digital Ecosystems: Principles and semantics", in Proc. IEEE DEST, Cairns, Australia, 2007, pp. 398-403.

[5] Li Ding, Tim Finin, Anupam Joshi, "Swoogle: A Semantic web search & Metadata Engine", DARPA, 2009.

[6] E. Chang and M. West, "Digital ecosystem-A next generation of the collaborative environment", in Proc. iiWAS, Yogyakarta, Indonesia, 2006, pp. 3-24.

[7] Chintan Patel, Supekar K., "OntoKhoj: A semantic web portal for ontology searching, ranking and classification", 19th ACM Conf. Information and Knowledge Management, Nov. 2009, pp.652-668.

[8] H. Dong, F. K. Hussain, and E. Chang, "A service search engine for the industrial digital ecosystems", IEEE Trans. Ind. Electron., vol. 58, no. 6, pp. 2183-2196, Jun. 2011, DOI: 10.1109/TIE, 2009.

[9] H. Dong, F. K. Hussain, and E. Chang, "State of the art in semantic focused crawlers", Proc. ICCSA, Yongin, Korea, 2009, pp. 890-904.

[10] H. Dong, F. K. Hussain, and E. Chang, "A survey in semantic web technologies-inspired focused crawlers", in Proc. 3rd ICDIM, East London, U.K., 2008, pp. 934-936.

[11] J. L. M. Lastra and M. Delamer, "Semantic web services in factory automation: Fundamental insights and research roadmap", IEEE Trans. Ind. Informat., vol. 2, no. 1, pp. 1-11, Feb. 2006.

[12] P. Malone, "DE services in Ecosystem Oriented Architectures," in Digital Business Ecosystems, F. Nachira, P.Dini, A. Nicolai, M. L. Louarn, and L. R. Lèon, Eds: Eur.Commission, 2007.

[13] H. Dong, F. K. Hussain, and E. Chang, "State of the art in metadata abstraction crawlers", in Proc. IEEE ICIT, Chengdu, China, 2008, pp. 1-6.

[14] M. Yuvarani, N. C. S. N. Iyengar, and A. Kannan, "Scrawled: A framework for an enhanced focused web crawler based on link semantics", in Proc. IEEE/WIC/ACM Int. Conf. WI, 2006, pp. 794-800.

[15] V. V. Sawant, Dr. V. R. Ghorpade, "Automatic Semantic Classification and Categorization of web services in Digital Environment", in ICCCT 2014, IEEE Hyderabad Section, 11-13 Dec 2014.