Falcon Concept Search-Based Automated Service Discovery

R. Gowri [1], M. Malaimalavathani [2]
[1]*Associate Professor, Department of IT, Sri Manakula Vinayagar Engineering College, Puducherry, India.*
[2] *PG Student, Department of CSE, Sri Manakula Vinayagar Engineering College, Puducherry, India.*

[1] gowrithirugnanam@gmail.com, [2] vathani.ani@gmail.com

Abstract— **Service Discovery is a major challenge in semantic web services. A vast majority of web services exist without explicit associated semantic descriptions. Many services that are relevant to a specific user service request may not be considered during web service discovery. We address the issue of web service discovery given nonexplicit service description semantics that match a specific service request. To address the better matching and ranking problem we propose semantic enhancement of the service request and falcon relationship. Falcon is one of the best Novel based search engine. The approach to novel keyword based web service with Falcon involves novel keyword based service categorization with enhancement of the service request. The proposed approach gives a solution for achieving functional level service categorization based on an ontology framework. The novel based categorization is performed at the universal description discovery and integration. This enhancement of the service request achieves a better matching and ranking with relevant services.**

Keywords— **Web service discovery, semantic, falcon, UDDI.**

I. INTRODUCTION

Web services, as a key technology for realizing service-oriented architectures, promise to enable interoperability and integration between heterogeneous systems and applications. The discovery and selection of the appropriate services to fulfill a given request constitutes a fundamental task in such architectures. However, current industry standards for registering and locating Web services (WSDL, UDDI) aim at describing the structure of the service interface and of the exchanged messages, limiting the discovery process to essentially keyword-based search. Even though interoperability at the syntactic level is a necessary requirement, the identification and selection of appropriate services should be done in terms of the semantics of the requested and offered capabilities. To this direction, the Semantic Web, through the use of ontologies, provides the means to enrich the service descriptions with semantic information, allowing software agents to reason about the terms in these descriptions. This is a significant step for increasing the precision of the discovery process, as well as for minimizing the required human intervention.

Efficiently finding Web services on the Web is a challenging issue in service-oriented computing. Currently, UDDI is a standard for publishing and discovery of Web services, and UDDI registries also provide keyword searches for Web services. With the development of Semantic Web technologies, more and more Semantic Web data is generated, which is being used in Web applications and enterprise information systems. To effectively utilize the large amount of semantic data, efficient search mechanisms customized for Semantic Web data, especially for ontologies, have been proposed for both humans and software agents.

Web services should be semantically annotated to provide the best match to the service requestor as per his requirements. In order to address these problems, an efficient Web services on the Web based on their associated semantics is presented in this paper. In Semantic Web research, ontologies provide the foundation for machine-processable data and allow to exchange information between people and machines by both syntactic and semantic. Ontology represents knowledge about a particular domain. This knowledge includes entities in the domain, their property and relationship with each other. Entities in the ontology are termed *Concepts*.

The main objective is to develop an effective mechanism for Web service discovery by evaluating the ranking scores of the services contained in the ontology that satisfies the desired functionality of the user.

II. BACKGROUND

Web Services (WSs) are modular, self-describing, and loosely coupled software applications that can be advertised, located, and used across the Internet using a set of standards such as SOAP, WSDL, and UDDI.

XML defines a meta-language for describing the data. In XML applications, data is described by text-based tags that give information about the data itself as well as its hierarchical structure. Because XML syntax consists of text-based mark-up that describes the data being tagged, it is both application-independent and human readable. This simplicity and interoperability have helped XML achieve widespread acceptance and adoption as the standard for exchanging information between heterogeneous systems in a variety of applications, including Web services. XML forms the basis for all modern Web services, which use XML-based technologies to describe their interfaces and to encode their messages. WSDL, SOAP, and UDDI all use XML-based messaging that any machine can interpret.

WSDL (Web Services Description Language) is a standard to describe how to access a Web service and what operations [8] (methods) it performs. WSDL is an XML-based format for describing Web services and their capabilities in a standardised format.WSDL specifies what a request message must contain and what the response message will look like in unambiguous notation. In additional to describing message contains, WSDL defines where the service is available and what communication protocol is used to talk to service [6].

SOAP (Simple Object Access Protocol) provides a simple, standards-based method for sending XML messages between applications. Web services use SOAP to send messages between a service and its client(s). Because HTTP is supported by all Web servers and browsers, SOAP messages can be sent between applications regardless of their platform or programming language. Since Web Services will run in a heterogeneous environment [7], the protocols used to perform the data transfer between functions have to be independent of any runtime environment. SOAP is a protocol having these characteristics. This quality gives Web services their characteristic interoperability.

UDDI (Universal Description, Discovery and Integration) serves as a “Business and services” registry and are essential for dynamic usage of Web services. They are used to publish and discover information about a business and its web services. This data can be classified using standard taxonomies so that information can be found based on categorization. UDDI, mostly contains information about the technical interfaces of a business’s services.UDDI is a framework that defines XML-based registries where businesses can publish information about themselves and the services they offer. A protocol for publishing and discovering metadata about Web services, to enable applications to find Web services, either at design time or runtime.

III. LITERATURE SURVEY

1. Semantic Web Service Discovery

Semantic Web Services discovery is commonly a heavyweight task, which has scalability issues when the number of services or the ontology complexity increases, because most approaches are based on Description Logic reasoning. As a higher number of services becomes available, there is a need for solutions that improve discovery performance. Preprocessing stage can be added based on two SPARQL queries [2] that filter service repositories, discarding service descriptions that do not refer to any functionality or non-functional aspect requested by the user before the actual discovery takes place. This fairly reduces the search space for discovery mechanisms, consequently improving the overall performance of this task.

Semantic Web query language is a natural fit for performing SWS discovery and ranking processes in terms of user requests, because, essentially, these processes search for elements in some sort of persistent storage using selection and ordering criteria. However, current query languages present shortcomings with respect to the level of inference and computation needed for SWS discovery and ranking. The following are the background elements of their proposal in order to contextualize and further motivate their work.

A. Querying the semantic web

There are three main approaches for Semantic Web query languages:

Graph-based query language: These languages allow to fetch RDF triples based on matching triple patterns with RDF graphs.

Rule-based query language: They propose logic rules to define queries, supporting RDF reasoning systems.

DL-based query language: They allow to query Description Logic (DL) ontologies described in OWL-DL, being able to search for concepts, properties, and individuals.

B. Discovering and Ranking

Starting from a service repository (S) containing definitions either using OWL-S, WSMO, SAWSDL, or WSMOLite, for instance, that conforms the search space, the discovery process searches for these available service definitions, which are described in terms of domain ontologies (O), that match with a user request (U). This matchmaking is usually performed using logic reasoning techniques, such as DL reasoners, logic programming, or hybrid approaches. The resulting discovered services are a subset of the initial repository, where each instance of this subset is considered to be compliant with the user request, to some extent.

Concerning user requests for SWS discovery and ranking, there are several approaches on how to define them. Thus, in standard WSMO they are described as goals, where the functionality requested by a user is defined by means of capabilities and interfaces. They can be used to match corresponding services in the discovery stage taking into account preconditions, effects, inputs, and outputs, among other description elements pertaining to capabilities and interfaces [2].

2. Semantic Web search: Framework and Evaluation

The possibility of using Semantic Web data is to improve hypertext Web search. In particular, they used relevance feedback to create a 'virtuous cycle' [1] between data gathered from the Semantic Web of Linked Data and web-pages gathered from the hypertext Web. Previous approaches have generally considered the searching over the Semantic Web and hypertext Web to be entirely disparate, indexing, and searching over different domains. Evaluating the work over a wide range of algorithms and options, and show it improves baseline performance on these queries for deployed systems as well, such as the Semantic Web Search engine FALCON-S and Yahoo! Web search. Further showing that the use of Semantic Web inference seems to hurt performance, while the pseudo-relevance feedback increases performance in both cases, although not as much as actual relevance feedback. Lastly, the evaluation is the first rigorous 'Cranfield' evaluation of Semantic Web search.

3. Mapping from WSDL to OWL-S

A mapping algorithm is introduced which represents the first step in the first phase of the proposed discovery mechanism. This algorithm aims to redefine the conventional web services using semantic markups. This does not only mean the process of converting the conventional web service description language (WSDL) to a semantic one (i.e. OWL-S), but it also means the standardization of this definition by using the concept of ontology to describe any type of data in the service. Consequently, the proposed algorithm contains an important component called the ontology search and standardization engine (OSSE) that helps in the standardization process. OSSE's function is based on searching for a suitable ontology in the "local ontology repository"[5].

4. Semantics-Based Automated Service Discovery

Semantic-based categorization of web services is performed at the UDDI that involves semantics augmented classification of web services into functional categories. The Semantically related web services are grouped together even though they may be published under different categories within the UDDI. Service selection then consists of two key steps: 1) parameters-based service refinement; and 2) semantic similarity-based matching. The web service input and output parameters contain the underlying functional knowledge that is extracted for improving service discovery. Parameter-based service refinement exploits a combination of service descriptions and input and output to narrow the set of appropriate services matching the service request, by combining semantics with syntactic characteristic of a WSDL document. The refined set of web services is then matched against an enhanced service request as part of Semantic Similarity-based Matching. The service request is enhanced by adding relevant ontology concepts, which improves the matching of the service request with the web services.

3. Falcons Concept Search

Falcons Concept Search, a novel keyword-based ontology search engine, as part of the Falcons system. It retrieves concepts whose textual description is matched with the terms in the keyword query and ranks the results according to both query relevance and popularity of concepts. The popularity is measured based on a large data set collected from the real Semantic Web. Each concept returned is associated with a query-relevant structured snippet, indicating how the concept is matched with the keyword query and also briefly clarifying its meaning.

IV. EXISTING SYSTEM

A majority of the current approaches for web service discovery call for semantic web services that have semantic tagged descriptions through various approaches, e.g., OWL-S, Web Services Description Language (WSDL)-S. However, these approaches have several limitations. First, it is impractical to expect all new services to have semantic tagged descriptions. Second, descriptions of the vast majority of already existing web services are specified using WSDL and do not have associated semantics. Also, from the service requestor’s perspective, the requestor may not be aware of all the knowledge that constitutes the domain. Specifically, the service requestor may not be aware of all the terms related to the service request. As a result of which many services relevant to the request may not be considered in the service discovery process.

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. Thus, the discovery process is also constrained by its dependence on human intervention for choosing the appropriate service based on its semantics.

Issues

When the large number of web services and the distribution of similar services in multiple categories in the existing UDDI infrastructure, it is difficult to find services that satisfy the desired functionality.

Such service discovery may involve searching a large number of categories to find appropriate services.

Therefore, there is a need to categorize web services based on their functional semantics rather than based on the classifications of service providers.

V. PROPOSED APPROACH

The limitations of existing approaches, an integrated approach needs to be developed for addressing the two major issues related to automated service discovery: 1) semantic-based categorization of web services; and 2) selection of services based on semantic service description rather than syntactic keyword matching. Moreover, the approach needs to be generic and should not be tied to a specific description language. Semantic-based categorization of web services is performed at the UDDI [3] that involves semantics augmented or classification of web services into functional categories. The semantically related web services are grouped together even though they may be published under different categories within the UDDI. Service selection then consists of two key steps: 1) parameters-based service refinement; and 2) semantic similarity-based matching.

In our proposed approach, semantic-based categorization of web services is performed at the UDDI that involves semantics augmented classification of web services into functional categories.

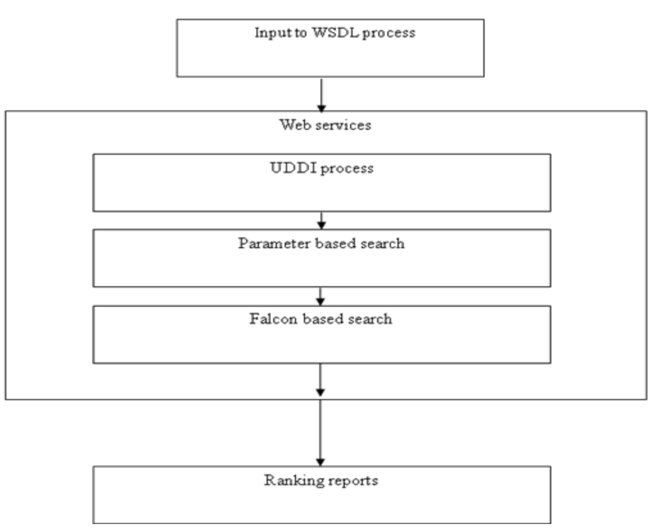

Fig 1: Architecture diagram

Ranking of Semantic Relationships

Semantic relationship among ontology concepts is generally ranked based on three parameters including relevance, specificity, and the span [3] of the relationship.

Relevance (Rel): Concepts may be associated with each other with reference to multiple domains that are specific to user applications. The associated domain for a particular concept may be expressed as a high-level concept in an upper ontology.

Specificity (Sp): The concepts are classified based on their position in the concept hierarchy. Concepts in the lower level of the hierarchy are specific concepts where as the higher level concepts are termed as generic concepts.

Span (S): The span of the relationships expressing the semantic association conveys the strength of linkage among concepts. The span, specified to restrict the scope of the user request, includes the coverage and the depth of the associated concepts. Coverage includes the concepts at the peer level of the considered concept where as the depth includes level of descendants to be included.

Hyperclique Patterns Discovery

Hyperclique patterns are based on the concepts of frequent item sets A hyperclique pattern is a new type of association pattern that contains items that are highly affiliated with each other. Specifically, the presence of an item in one service description vector strongly implies the presence of every other item that belongs to the same hyperclique pattern.

LSI (Latent Semantic Indexing)

LSI [3] is utilized over a set of WSDL documents and the terms in the service description and parameters. After analyzing a base set of web service documents, it finds relations between web service terms including service description and parameters. Given a term query, LSI translates it into concepts, and finds matching documents and corresponding web services.

Semantic Categorization of Web Services

This approach starts with the semantic categorization of UDDI wherein combining ontologies with an established hierarchical clustering methodology, following the service description vector building process. For each term in the service description vector, a corresponding concept is located in the relevant ontology. If there is a match, the concept is added to the description vector. Additional concepts are added and irrelevant terms are deleted based on semantic relationships between the concepts. The resulting set of service descriptions is clustered based on the relationship between the ontology concepts and service description terms. Finally, the relevant semantic information is added to the UDDI for effective service categorization.

Parameters-Based Service Refinement

By using parameter-based service refinement service is selected from the relevant category of services. Web service parameters, i.e., input, output, and description, aid service refinement through narrowing the set of appropriate services matching the service request.

The relationship between web service input and output parameters may be represented as statistical associations. These associations relay information about the operation parameters that are frequently associated with each other. To group web service input and output parameters into meaningful associations, applying a hyperclique pattern discovery approach. These associations combined with the semantic relevance are then leveraged to discover and rank web services.

Semantic Similarity-Based Matching

The parameter-based refined set of web services is then matched against an enhanced service request as part of Semantic Similarity-based Matching. This process involves enhancing the service request. This approach for web semantic similarity-based service selection employs ontology-based request enhancement and LSI based service matching. The idea of the proposed approach is to enhance the service request with relevant ontology terms and then find the similarity measure of the semantically enhanced service request with the web service description vectors generated in the service refinement phase. For evaluating this similarity, employ LSI-based technique.

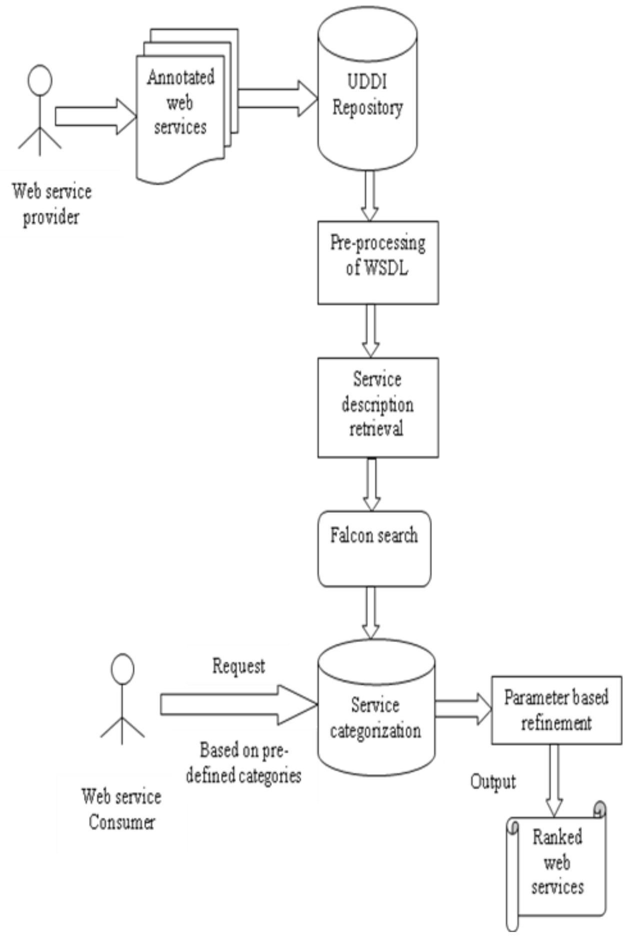

Fig 2: Proposed methodology

VI. OUR CONTRIBUTION

We present Falcon based search engine which involves novel keyword search concept and service categorization with enhancement of the service request. Ontology's are recommended to be selected to filter the concepts returned. We rank ontology's based on the ranking concepts.

Ranking

A. Concept Ranking

In the system, the ranking score of a concept *c* is concerned with two factors, i.e., its relevance to the keyword query *q* and its popularity

RankingScore (c, q) = TextSim(c, q) · Popularity(c)

1) Query Relevance

On the one hand, a virtual document is constructed for each concept. On the other hand, a keyword query can be treated as a short document. Thus, the problem of calculating the relevance of a concept to a keyword query could be transformed into the problem of calculating similarity between two documents. They use the vector space model and the term frequency weight to represent documents, i.e., each document is represented as a vector where each component corresponds to the frequency of a term in the document. In particular, the weights of the terms extracted from the local name and label of the concept in question are additionally multiplied by 10.0, based on our previous experience of using virtual documents in ontology matching. Then, weights are further refined by the well-known inverse document frequency measure, i.e., a higher weight is assigned to a term in a virtual document if the term occurs in fewer documents in the whole data set because such a term is considered to be a more distinctive feature. Finally, the relevance of a concept *c* to a keyword query *q*, TextSim(c, q), is defined as the cosine of the angle between the vector form of the virtual document of *c* and the vector form of *q* [4].

2) Popularity

The existing approaches failed to investigate the use of concepts in practice. To develop a new Web application, in order to maximize the interoperability of different applications, one best practice is to reuse concepts that have been widely used by existing applications. Therefore, the system gives higher ranks to popular concepts.

For a concept *c*, let Docs(c) be the set of RDF documents where *c* is instantiated. A concept *c* is instantiated in an RDF document *d* if either *c* is a class and *d* contains an RDF triple whose predicate is rdf:type and whose object is *c*, or *c* is a property and *d* contains an RDF triple whose predicate is *c*. The popularity score of *c* is calculated as follows:

Popularity(c) = log $(|\text{Docs}(c)| + 1) + 1$.

In the system, popularity scores are evaluated based on a large data set collected from the real Semantic Web, which includes not only conceptual-level RDF documents (ontologies) but also a lot of instance-level RDF documents. Therefore, it is possible to characterize the use of concepts in practice.

B. Ontology Recommendation

In the system, according to the proposed mode of user interaction, several ontologies are recommended to be selected to filter the concepts returned. Now, ranking ontologies based on the ranking of concepts. For a keyword query, the ontologies that the concepts returned come from are regarded as candidates for recommendation. For each ontology candidate, its ranking score is evaluated by adding up the ranking scores of those concepts returned and contained in this ontology. Finally, up to nine top-ranking ontologies are recommended. The underlying criterion is an ontology and more likely to be recommended if the concepts in the ontology that are matched with the terms in the keyword query are more popular on the Semantic Web.

VII. MODULES

A. WSDL Process

In this module the design and implementation is described via web based services. Both semantic-based service categorization and parameter-based service refinement depend on the service description in the WSDL file. Additionally, we consider keyword-based search for service discovery in the WSDL process.

B. Service Categorization

In this module, the semantic categorization of UDDI wherein we combine ontologies with an established hierarchical clustering methodology, following the service description vector building process. For each term in the service description vector, a corresponding concept is located in the relevant ontology. If there is a match, the concept is added to the description vector. Additional concepts are added and irrelevant terms are deleted based on semantic relationships between the concepts. The resulting set of service descriptions is clustered based on the relationship between the ontology concepts and service description terms. Finally, the relevant semantic information is added to the UDDI for effective service categorization.

C. Service Refinement

In this module, the service selection is processed from the relevant category of services using parameter-based service refinement. Web service parameters, i.e., input, output, and description, aid service refinement through narrowing the set of appropriate services matching the service request. The relationship between web service input and output parameters may be represented as statistical associations. These associations relay information about the operation parameters that are frequently associated with each other. To group web service input and output parameters into meaningful associations, we apply a hyperclique pattern discovery. These associations combined with the semantic relevance are then leveraged to discover and rank web services.

D. Service Matching

In this module, the parameter-based refined set of web services is then matched against an enhanced service request as part of Semantic Similarity-based Matching. A key part of this process involves enhancing the service request. Our approach for web semantic similarity-based service selection employs ontology-based request enhancement and LSI based service matching. The basic idea of the proposed approach is to enhance the service request with relevant ontology terms and then find the similarity measure of the semantically enhanced service request with the web service description vectors generated in the service refinement phase.

E. Ranking of Falcon Relationship

In this module, the parameters for ranking semantic relationships in the context of semantic-based service categorization. Concepts may be associated with each other with reference to multiple domains that are specific to user applications. The associated domain for a particular concept may be expressed as a high-level concept in an upper ontology.

VIII. CONCLUSION

In this work, The Semantic Based Automated Service Discovery involves an ontology guided categorization of web services into functional categories for service discovery. The semantic enhancement of the service request achieves a better matching with relevant services with falcon based search concept. An efficient matching of the enhanced service request with the retrieved service description also achieved by utilizing Latent Semantic Indexing (LSI).The service discovery is achieved by matching the service request with an appropriate service description and efficient ranking concept is obtained based on their popularity of semantic Web as well as their relevance to keyword queries**.** By these approaches, we achieve the better matching and the ranking results.

REFERENCES

[1] "Relevance feedback between hypertext and Semantic Web search: Frameworks and evaluation", Harry Halpin, Victor Lavrenko, Web Semantics: Science, Services and Agents on the World Wide Web 9 (2011) Elsevier

[2] "Improving semantic web services discovery using SPARQL-based repository filtering", José María García , David Ruiz, Antonio Ruiz-Cortés, Web Semantics: Science, Services and Agents on the World Wide Web 17 (2012) Elsevier

[3] “Semantics-Based Automated Service Discovery”, Aabhas V. Paliwal, Student Member, IEEE, Basit Shafiq, Member, IEEE, Jaideep Vaidya, Member, IEEE, Hui Xiong, Senior Member, IEEE, and Nabil Adam, Senior Member, IEEE, 2012

[4] “Falcons Concept Search: A Practical Search Engine for Web Ontologies”, Yuzhong Qu and Gong Cheng, Ieee transactions on systems, man, and cybernetics—part a: systems and humans, vol. 41, no. 4, july 2011

[5] “Towards SWSs discovery: mapping from WSDL to OWL-S based on ontology search and standardization engine”, Tamer A. Farrag, A. I. Saleh and H. A. Ali, Department of Computers and Systems, Faculty of Engineering, Mansoura University, Egypt, ieee transactions on knowledge and data engineering, vol. x, no. x, x 20xx

[6] “Web services: Benefits, challenges, and a unique, visual development solution” Written by Erin Cavanaugh. Altova.

[7] “Introduction to Web Services” by Hartwig Gunzer, March 2002.

[8] “Analysis on the Study of QoS-Aware Web Services Discovery” T. Rajendran and Dr. P. Balasubramanie JOURNAL OF COMPUTING, VOLUME 1, ISSUE 1, DECEMBER 2009

[9] “Semantics-Enabled Service Discovery Framework in the SIMDAT Pharma Grid” Changtao Qu, Falk Zimmermann, Kai Kumpf, Richard Kamuzinzi, Val´erie Ledent, and Robert Herzog, *Member*, *ieee*, ieee transactions on information technology in biomedicine, vol. 12, no. 2, march 2008

[10] “Semantic web services” Sheila A. McIlraith, Tran Cao Son, and Honglei Zeng, *Stanford University*, ieee intelligent systems 2001

[11] “Searching Semantic Web Objects Based on Class Hierarchies”. In: Bizer, C., Heath, T., Idehen, K., Berners-Lee, T. (eds.) LDOW 2008. CEUR-WS, vol. 369. CEUR-WS.org (2008)

[12] “Structural and Role-Oriented Web Service Discovery with Taxonomies in OWL-S”, Georgios Meditskos and Nick Bassiliades, Member, ieee, ieee transactions on knowledge and data engineering, vol. 22, no. 2, february 2010

[13] “Dynamic Discovery and coordination of Agent- based Semantic Web Services”, Michael N. Huhns *huhns@sc .edu* 2004 ieee internet computing

[14] “Agent-Based Semantic Service Discovery for Healthcare: An organizational Approach” César Cáceres, Alberto Fernández, Sascha Ossowski, and Matteo Vasirani**,** *University Rey Juan Carlos* 2006 IEEE

[15] “Information Retrieval and Knowledge Discovery Utilizing a BioMedical Patent Semantic Web”. Sougata Mukherjea, Bhuvan Bamba, and Pankaj Kankar ieee transactions on knowledge and data engineering, vol. 17, no. 8, august 2005

[16] “Semantic Web Content Analysis: A Study in Proximity-Based Collaborative Clustering” Vincenzo Loia, Witold Pedrycz, and Sabrina Senatore ieee transactions on fuzzy systems, vol. 15, no. 6, december 2007

[17] “ManuHub: A Semantic Web System for Ontology-Based Service Management in Distributed Manufacturing Environments” M. Cai, W. Y. Zhang, and K. Zhang ieee transactions on systems, man, and cybernetics—part a: systems and humans, vol. 41, no. 3, may 2011

[18] “Semantic Discovery of Web Services through Social Learning” Sajib Kumar Mistry, Mosaddek Hossain Kamal, Dilip Mistry 2012 Published by Elsevier.