Fuzzy-XDDS: A Fuzzy Based Cross-Domain Services Matchmaker for Semantic Web Services

B. Meenakshi Sundaram
Asst. Prof, Department of Computer Applications
Karunya University
Coimbatore
Tamil Nadu, India

D. Manimegalai
Prof & Head, Department of Information Technology
National Engineering College
Kovilpatti
Tamil Nadu, India

Abstract—Web Services have been growing as an inventive mechanism to render various services to random devices over the World Wide Web (WWW). These Services can be interoperable through platforms and also neutral to all programming languages, which makes them suitable for accessing the Heterogeneous and different Environments. The consumer could not select the “right” Service Provider as a concern of the fast growth of various Applications of various Web Services and the plenty of available Service Providers. To help consumers, the UDDI (Universal Description, Discovery, and Integration) had proposed earlier, which catalogues of published the Web Services Description Language specifications of available Services. These catalogues are organized as per sequences of processes involved in the business domain. The Service Providers advertising their Services to the suitable categorical UDDI explorer. By using a well-formed API (Application Programming Interface) calls, the Service Requesters surf the UDDI by group and type wise to discover the available Web Services applicable to users’ query. The Semantic Web Services are presented to discuss the above mentioned problem and also to enable the publication of autonomous, discovery and Web Services Execution. A few recently proposed Fuzzy based Services Matchmaking Frameworks such as Fuzzy Matchmaking for Web Services, A Moderated Fuzzy Matchmaking for Web Services, Adaptive Fuzzy-Valued Service Selection, A Hybrid Approach to Semantic Web Services Matchmaking and Fuzzy Matchmaking Model for Semantic based Web Services have been studied thoroughly. This research work has revealed that the latest model called Fuzzy Matchmaking Approach for Semantic Web Services performing well in terms of Recall Rate and Precision Rate for selecting Web Services as compared with existing models. This model ranks services as per the significant weights of the developer’s query. However, from our study, this work has revealed that this model does not focus both the Services Ontology and Matching Services across domain boundaries while ranking and selecting Services for Service Requesters’ request and hence this model couldn’t provide required QoS to the Web Users. To address these major issues, this research work has proposed an efficient Fuzzy-based Cross-Domain Services Matchmaker (Fuzzy-XDDS) which is extending Service Discovery across Domain Boundaries. This model also has the ability to discover services across domain boundaries in a Flexible and Scalable manner as this is proposed for Cross-Domain Web Service Discovery which is based on the principles of Interoperability, Modularity and Scalability. The experimental results show that our proposed Fuzzy-XDDS performing well as compared with Fuzzy Matchmaker in terms of Precision Rate, Recall Rate and Imprecision Rate.

Index Term—Semantic Web service Selection; Fuzzy-XDDS; Cross Domain Service Selection; Fuzzy Matchmaker.

I. INTRODUCTION

The Web Services based Service-Oriented Computing paradigm and that of realization through various Web Services Technologies have provided a challenging solution for the seamless integration of various enterprise and creativity applications to produce value-added Services [8]. Industrial practices establishing a growing and rising interest in the Service-Based Web Systems Interoperability and Integration for supply chains, e-business, dynamic alliances, virtual organizations and extended enterprises [2–4]. A few critical steps in the processes of reusing the existing Web Service Description Language (WSDL) has specified services to build a large scale heterogenous applications is the discovery of relevant Web Services[9] as shown in Figure 1. wherein the Servers have published the Web Service Description Language which specifies the available services. The catalogs have organized according to types, groupings and categories of business activities. Service Providers advertise Web Services by adding their Web Service Description Language specifications to the appropriate Universal Description, Discovery, and Integration (UDDI) directory category. Through a well-defined API, the Web Service Requesters could be browsed the Universal Description, Discovery, and Integration (UDDI) directory by category to discover existing services to query. However, this approach of Category-Based Service Discovery is clearly become insufficient, because it depends on the shared common- sense application domain among the developers who have published and requested various services.

Service lookup using UDDI

Fig. 1. Web Service Architecture Framework
Semantic Web Services (SWS) [10, 11] using Semantic Web Technology have presented to highlight the above identified problem and to facilitate the publication, discovery and execution of different Web Services. Further OWL-S, the Semantic Web Service Description Languages and WSMO [13] had proposed as abstractions of syntactic such as WSDL. The OWL-S describes the group of categories, inputs and outputs of Web Services in terms of concepts defined in the OWL ontology. To support Service Discovery, the Semantic Matchmakers [14, 15], it keeps the track of the descriptions of available Web Services from Service Providers and match them from Service Requesters [10, 16]. Intelligent Matching Techniques between publications and service requests described in OWL-S. Further the Fuzzy Matching Framework for Web Services uses fuzzy logic to classify the Web Services as fuzzy terms and rules [17, 15]. Other type of fuzzy-valued similarity measure applies a fuzzy-valued description of QoS parameters for selecting the most suitable services [19].

As discussed earlier, the Service Discovery does play a significant role in single domain SOAs. As Web services have become available or get removed, all Web Service Applications need to be up-to-date. The management of static depictions of these environments becomes unmanageable both within and across domains. This motivates a requirement for discovery of services across the domains whereas existing service discovery solutions works within domains. This research work has addressed this issue and proposed an efficient Fuzzy based Cross-Domain Services Matchmaker for Semantic Web Services.

II. RELATED WORKS

The web services are the business operations carried out by the enterprises. Every web service component needs a details includes: a service code, a business key, a service name, an optional description, a category Bag and the bindingTemplates, which contain one or more binding Template structures. The web services are stored in the public data registry called UDDI. The tModel is a data structure to store the web service component details while registering them with the public repository. Thereby to retrieve the component the interfacing signature need to be matched while querying. Thus, a semantic information-retrieval approach could be used to recognize and relevant specifications of WSDL based on the likeness of their descriptions with query.

The Vector-Space Model [24, 25] is a successful tool to do the Information-Retrieval. Here, the documents have been represented as \( t \)-dimensional Vectors and every element in the vector is having the distinct words enclosed in the document. The Signature Matching is considering the function types only and it has ignored their behaviors and hence the two functions with the same signature will be having completely opposite and different behaviors.

In Service Matching, the automatic discovery of Web Services is depending on the ability to find the best appropriate services. From the literature survey [23, 28–30], this problem has called as “Matchmaking” and its various processes are

- Semantic Matching
- Capability/Functional Matching
- Ranking of Services to select the best

A. Semantic Web Services

To speed up the service discovery process, composition and execution, every service is documented with appropriate description. OWL-S defines the ontology for semantic markup of Web Services. The main objective of this Semantic Web Service is process integration. In the functional/capability matching approaches, the author Gao and et.al., [30] have proposed a formal Capability Description Language (SCDL) which provides the primitives for a high level ontological description. Sycara [15] and Paolucci [14] of Carnegie Mellon University have proposed Semantic Web Services which is supporting capability matching and to manage the interaction between various Web Services. Ponnekanti and Fox [1] have focused on identifying and bridging incompatibilities of services using a testing approach. Woogle System [1] was clustered WSDL descriptions according to the names of the services operation parameters and it has leveraged these clusters to retrieve Web Services which may be delivered operations of interest to the requester. Chung and Kim [10] have proposed a matching architecture using Semantic Web Services for collaborative product. There have been some fuzzy matchmaking efforts which have aimed at bridging the gap between requiring full-fledged semantic specifications and relying simply on currently available WSDL specifications for Web Service Matching. Buche [1,33] has used a multi-view fuzzy querying, where the fuzzy sets have defined on a hierarchical domain of values called an ontology to query incomplete, imprecise and heterogeneously
A database with fuzzy selection criteria.

Bozdag has presented a fuzzy group decision making for selection among Computer Integrated Manufacturing Systems [1]. Chao [17] has studied the fuzzy matchmaking for Web Services and used fuzzy logic to classify and to represent the vague or imprecise data at the Web Service abstract level. However, the data is information has been arbitrarily classified. This cannot use for application since service consumers and providers have inconsistent classifications.

The author Huang [18] has proposed a moderation mechanism to allow that the service classification which can be moderated through the exercise of reaching consensus process in order to reflect consumer’s expectations and bridge the differences between a collection of Web Service Consumers and Provider. In the Fuzzy Matching for Web Services, the semantic information of web services did not include.

The author Min Liu have proposed a Similarity-Based Fuzzy Matchmaking Approach for improving the precision rate and the recall rate for material selection services. However, we observed that the requirement for discovery services across domains is currently unmet. This is one of the serious issues, which is addressing by our proposed work.

B. Fuzzy Matching Approach for Semantic Web Services

In this section, we are discussing the existing Fuzzy Matching approach for Semantic Web Services [2].

The Fuzzy Matching Framework (FMF) has composed of Web Service Deploying Module, Web Service Registering Module and Web Service Calling Module.

The Web Service Register Tool reads the WSDL documents written with WSDL and stored in service company decides a task to outsource through Web Services and calls the available Web Service or Services when a company decides a task to outsource through Web Services and calls the available Web Service or Services when a

In Web Service Calling Module, FMF searches, matches and calls the available Web Service or Services when a company decides a task to outsource through Web Services and calls the available Web Service or Services when a

C. Fuzzy definition for Semantic Attributes of Web Services

A fuzzy set \( f_i \) is called a Fuzzy Set, if it is defined on a domain of values \( O \), which is a subset of values belonging to the ontology \( O \). For all the couples of values \( x \) and \( y \) belonging to \( S(f_i) \), with \( y \) more specific than \( x \), the underlying semantics [2] is defined as follows:

Given \( \mu_f \), the membership function of \( f_i : O \to [0, 1] \):

1. \( \mu_f(x) \geq \mu_f(y) \) represents a semantics of restriction for \( y \) compared to \( x \).
2. \( \mu_f(x) < \mu_f(y) \) represents a semantics of reinforcement for \( y \) compared to \( x \).

D. Matching Model

To improve the current matching mechanisms, the authors Zhou, Z and, Sellami[1] have proposed the model which exploits the Fuzzy Logic and ontology to explain and to represent the data of Semantic Web Services. i.e we use the fuzzy description in Universal Description Discovery and Integration (UDDI) to allow consumers to search with imprecise queries. It enhances the matching of Universal Description Discovery and Integration (UDDI) and discovers the hidden dimension to label Web services for searching. Figure 3 shows the proposed model of Zhou, Z, Sellami[1] and the following is their model.

![Fig. 3. Fuzzy Matchmaking Framework for Semantic Web Service Execution Environment [2]](image-url)
In this model, all Web services described with OWL-S will be registered in UDDI by their service providers. The Fuzzy Classifier is asking for the databases of Web Services that have registered in UDDI and uses OWL to explain the heterogeneous database of each Web Service. Fuzzy Classifier explains the heterogeneous database by OWL and represents the abstract description of Web Services by Fuzzy Logic. Fuzzy Classifier classifies Web services as the illustration of various Fuzzy Terms. Hence, Fuzzy Classifier records fuzzy values of each Web Service for specific fuzzy terms in UDDI for searching.

In Fuzzy Engine, it defines fuzzy terms and related membership functions that have based on the Web Services. Based on the OWL interpretation, fuzzy classifier calculates the fuzzy value of Web Services for different fuzzy terms. When the query is addressed, fuzzy matching parses query sentence and request UDDI and fuzzy engine to return suited Web services to requesters.

E. Fuzzy Classifier

The objective of fuzzy classifier is to represent Web services as fuzzy terms that are related with different fuzzy sets. Web service is classified by fuzzy classifier according to data of Web service to represent Web services with fuzzy description. In this model [1], the authors define the explanation about slot names by OWL. They have used OWL Jess KB as a reasoning tool to parse OWL document which interprets the relationship among heterogeneous databases. After realizing the meaning of each slot, they have calculated the slot value with related membership functions defined by fuzzy rules. Fuzzy engine returns the fuzzy value to the classifier. Fuzzy classifier then records the fuzzy values in UDDI for matching. Fuzzy Classifier is recording the different fuzzy values of each Web service. It is integrating UDDI with fuzzy logic and hence, it allows consumers to search Web services with vague queries in UDDI.

F. Fuzzy Matching

Based on fuzzy operations, it is parsing Fuzzy Operations as “AND” “OR” “NOT” from the query and these are as called operators. The objective of fuzzy matching was to obtain the suited Web Services. With the fuzzy matching mechanism, it could obtain the target services from UDDI and filter those services with the description of Model through the fuzzy operations. Fuzzy matching is functioning as the interface for communication between UDDI, Fuzzy Engine and Consumers.

G. Fuzzy similarity for Web Service Objects

Min Liu[1] defined a goal Service as a network that is traveled by an object, undergoing transformations caused by Services. Object characterizations have described as envelope instances. Services required special conditions for objects to be processed and how the objects are then transformed. Let \( X_1 = e_1 : E_1 + (\text{Int}_{1}, \text{Ext}_{1}) \) and \( X_2 = e_2 : E_2 + (\text{Int}_{2}, \text{Ext}_{2}) \) be elementary object conditions. We are interested in \( \text{Sim}(X_1, X_2) = \text{Sim}(e_1 : E_1 + (\text{Int}_{1}, \text{Ext}_{1}), e_2 : E_2 + (\text{Int}_{2}, \text{Ext}_{2})) \).

Web Services have various conditions that must be satisfied before it is executed. If a pre-condition is not met, the service will not be executed. In other words, the service cannot be invoked and the post-condition will not be met. For similarity computations, the authors [1] ignore these assumption ie the authors assume them as true. During the construction of a goal service, the condition has to be taken into account though. Testing the condition may be seen as a service that provides information about the object. Consequently:

\[
\text{Sim}(e_1 : E_1 + (\text{Int}_{1}, \text{Ext}_{1}), e_2 : E_2 + (\text{Int}_{2}, \text{Ext}_{2})) = \text{Sim}(e_1 : E_1 + (\text{True}, \text{Ext}_{1}), e_2 : E_2 + (\text{True}, \text{Ext}_{2}))
\]

The requirements of the object pattern can be interpreted as a restriction on attribute values for the attributes of the associated envelope. The object pattern \( e : E + (\text{True}, \text{Ext}) \) describes an object that has envelope E. Some of the attributes of E have values assigned, as specified by the (partial) envelope instance e. For the other attributes X the values are bound by the condition Ext. Let us now consider the contribution of an attribute x to the similarity of two service objects. Attributes can take values from their associated domains within the restrictions from the conditions Ext1 and Ext2. This is seen as conditional probability expressed:

\[
\text{Prob}(X_1|X_2) = \text{Prob}(e=|X_1|e=X_2).
\]

It therefore follows that

\[
\text{Prob}(X_1|X_2) = \begin{cases} 1 & \text{if } v \in A \\ 0 & \text{otherwise} \end{cases}
\]

For each pair of the service profiles, the degree of match is calculated by using the weighted average of the matching scores between the pairs of the profile types, the input parameters, the effects of service, and the output parameters. Mathematically, the degree of match between a pair of service profiles is

\[
D = \sum_{i} W_i d_i^{p}\sum_{i} W_i
\]

where, \( D \) is the degree of match between two service profiles, \( W_i \) and \( d_i^{p} \) is respectively weight and the matching scores between the profile types, the input parameters, the effects of services and the output parameters.

Web services often take several input parameters. In this case, the matching score between the two sets of input parameters is the un-weighted average of the matching scores between each sequential pair of the input parameters. Mathematically, it is defined as

\[
d_i^{p} = \sum_{j} d_{ij}^{p}/N
\]

where, \( d_{ij}^{p} \) is the degree of match between the jth
pair of input parameters, \( N \) is the total number of input parameters. To eliminate the case of strong incompatibility between input parameters, \( d_p \) must be taken as zero if any of \( d_i \) equals zero. Based on the above computing process, the degree of match between the query service and the advertisement services from the registry center has calculated.

III. IDENTIFIED PROBLEM

The above discussed Service Discovery solutions will be working only within domains. This type of Service Discovery plays a significant role in SOAs because of the dynamic Service Environment, which is shown in the Fig. 4. As Services become available as shown in the Fig. 4, removing or changing, all applications need to up-to-date Information. The Fuzzy Matchmaking Approach for Semantic Web Services had proposed recently and it is performing well in terms of Precision Rate and Recall Rate for selecting Web Services as compared with existing models. This model ranking services according to the relevance of the developer’s query. However, from our study, this work has revealed that this model does not focus both the Services Ontology and Matching Services across domain boundaries while ranking and selecting services for service requesters’ request and hence this model could not provide required QoS to the Web Users.

Fig. 4. Functional View of the Discovery Process within a Single Domain

To address these major issues, this research work has proposed an Efficient Fuzzy based Cross-Domain Discovery Services Matchmaker which is extending Service Discovery across Domain Boundaries. The Model developed under our Cross Domain Discovery Service (XDDS) effort fill this gap.

Thereby the dynamic discovery is across all type of domains and is associating Non-Hierarchical and Hierarchical relationships.

IV. PROPOSED WORK

A. Fuzzy-XDDS Matchmaker

The Cross Domain Discovery Service (XDDS) is describing in this research work fills the gap, which we identified as limitation of Discovery Process within a Single Domain by enabling dynamic discovery and it is associating relationships including Non-hierarchical, Hierarchical and coalition. Under the XDDS effort, this research work has designed a cross domain discovery service as shown in the Fig. 5 that this is providing dynamic and assured discovery in a flexible and scalable way and it is developed a prototype to demonstrate practicality of this proposed Model. This Model has designed in such a way that this is interconnecting with numbers of Local Discovery Process Domain and the Messages and Information of all these Local Discovery Process Domain can be linked with Cross-Domain Discovery Services Matchmaker which is called as Global Discovery Service. The detailed working nature of this model is explained in the following section.

B. Architecture of Cross-Domain Discovery Services Matchmaker

The Architecture of Cross-Domain Discovery Services Matchmaker is shown in the Figure. 5. The Discovery Process within a Single Domain which has to communicate with the local service substrate, it is instantiating service proxies and uses adapters for supporting a number of various protocols. This is implemented proxies to discover and implemented adapters for HTTP and UDDI.

For communication with other Discovery Process within a Single Domain, it uses Discovery Process within a Single Domain service proxies. This prototype involved to interact all Discovery Process within a Single Domain(s) through the SOAP. i.e. this discovery process itself has implemented as a Semantic Web Service and launched in Apache Axis2 under Java 1.6.1[1,35] hosted in the Axis2[1] web services container.

The heart of the Discovery Process within a Single Domain(s) is the layered protocol stack and it is used to identify the identity management. The main objective of this Layered Protocol stack is to convert messages from the local service substrate, which will be technology specific, which is suited for crossing domain boundaries. The collection of core messages can be described through abstract protocols for identity, discovery, and brokering. The brokering protocol is responsible to route all the requests through the network of XDDS nodes. Requests can be either identity management related or discovery or originate from a brokering proxy. This will allow all the clients to invoke various services in other XDDS-enabled domains. The identity protocol is enabling the Discovery Process within a Single Domain(s) to export a few selected identity mappings from the local identity service to other XDDS nodes.

Fig. 5. Fuzzy-XDDS Matchmaker Architecture
C. Cross-Domain Discovery Services Matchmaker Specification

The Cross-Domain Discovery Services Matchmaker message protocol is an XML-based one which is describing the syntax of messages communicated between Discovery Processes within a Single Domain(s) through the Cross-Domain Discovery Services Matchmaker. The protocol is consistent with open standards, e.g., XML, UDDI, WS-Security, XML Signature and SOAP. The protocol represents XML message exchanges through two basic message forms – XDDS requests and XDDS responses. By default, all requests generate responses and generic acknowledgement responses are returned in error cases instead of error responses.

Messages include control information, such as Discovery Process within a Single Domain(s) identities used for routing purposes, classification markings, and message integrity and provenance trails that allow enforcement of integrity and anti-spoofing. To simplify messaging formats, the protocol uses the same message types during referral and replication modes and treats the replication request analogous to a query response in the referral mode. Furthermore, the application specific discovery protocols, e.g., UDDI and HTTP, are encapsulated in the XDDS messages in restricted form, allowing the same XDDS message structure to be used with multiple application specific protocols. The XDDS protocol allows expression of restrictions on message exchanges through both XML schema and XSLT restrictions.

D. Procedure of Fuzzy Cross-Domain Discovery Services Matchmaker

The following briefly describes different steps involved in the proposed framework.

- The first step is to generalize contents into fuzzy terms by employing fuzzy classifier which represents a Service or a sub-service as descriptive fuzzy terms based on the data related to web services. These terms are loosely structured as a hierarchy via fuzzy rules and the predefined fuzzy sets. The high-level terms are embedded and published in UDDI are available for matchmaking.
- When a service consumer initiates a query, the fuzzy classifier checks and transforms crisp terms used in the query to fuzzy terms. The fuzzy matchmaking mechanism is triggered by relating objects in the request with other objects represented in ontologies. Capability matchmaking is processed first. If there is a capability match in terms of their parts or types, fuzzy reasoning, is then used to map the fuzzy request to the appropriate data of services.

The following section describes Framework and Fuzzy Rules of proposed Fuzzy-XDDS.

This proposed Fuzzy Matchmaking Framework is representing the Web Services Information through Fuzzy Logic, which is facilitating to optimize the discovery process. It comprises various components like Fuzzy Engine and Fuzzy Classifier, Universal Description, Discovery and Integration, Web Ontology Language – Semantic (OWL-S) and Fuzzy Converter. The Procedure of the Fuzzy-XDDS is given below.

Fuzzy based Matching Technique

Fuzzy.XDDS.FuzzyEngine()
{
Invoking Web Services()  
Invoking Web ServicesTo.UDDI.OWL-S();  
Invoking WebServices.To.FuzzyClassifier()  
FuzzyTerms(UDDI.OWL-S());  
FuzzyEngine(FuzzyClassifier)
FuzzyMatchmaking(UDDI.OWL-S())
FuzzyConverter(FuzzyEngine)
}
  
FuzzyConverter  UsersReq()  
Users  MatchedServices.Matchmaking(User)

Domain 1()
{
ServicesAdvertisements.ServiceRequest()  
SemanticDescription();  
Extraction.FunctionElements()  
Fuzzy.Transformation();  
AttributesReduction()  
SemanticMatching();  
ResultServices(User)
}

Invoking Web Services()
Invoking Web ServicesTo.UDDI.OWL-S();  
Invoking WebServices.To.FuzzyClassifier()  
FuzzyTerms(UDDI.OWL-S());  
FuzzyEngine(FuzzyClassifier)
FuzzyMatchmaking(UDDI.OWL-S())
FuzzyConverter(FuzzyEngine)
The Fuzzy Rules of Fuzzy-XDDS is given below. It is converting equivalent Fuzzy Terms for Users OWL-S(advertisement:UDDI)

\[
\text{If (User Request) Initiates} \\
\{ \\
\quad \text{FuzzyClassifier :} \\
\quad \quad \text{(Terms : Term1, Term2, ... Terms n)} \\
\quad \text{FuzzyConverter :} \\
\quad \quad \text{FT1} \leftarrow \text{ChangeCustomerReqToFuzzyTerms()} \\
\quad \quad \text{WS} \leftarrow \text{FuzzyEngineExecute(FT1)} \\
\}
\]

E. Case Study

For illustrating the above proposed Fuzzy Rules, we have considered one of the Web Services Flight-Booking Service model. For demonstrating this purpose, we have considered a term called satisfaction. This term represented as satisfaction\( \hat{Q} \), which is deriving from the following Fuzzy Rules.

- Cheap: This is giving cost details of the Ticket of a Flight. It is referring as Cheap(\( \hat{Q} \))
- Departure Time: This is specifying the Departing Time of a Flight. i.e DepartureTime(\( \hat{Q} \))

and similarly we have calculated Travel Time(\( \hat{Q} \)) and ArrivalTime(\( \hat{Q} \))

V. EXPERIMENTAL PERFORMANCE EVALUATION

The proposed Fuzzy based Cross-Domain Discovery Services Matchmaker (Fuzzy-XDDS) work has investigated the various issues of web services matchmaking by considering Location of Services, Software Signatures, Web Services Semantics and Syntax and the essential data of services.

We have highlighted and utilized fuzzy logic to classify the web services. In this research paper, we have contributed the following

- Identification of a Hidden Dimension
- Representation of Web Services Data based on Fuzzy Set Theory
- Representation of Fuzzy Sets and Fuzzy Rules

For Fuzzy Sets and Rules, we have used OWLJESSKB, which is logic reasoner for the World Wide Web Consortium’s Ontology Web Language [2]. The Ontology Web Language has implemented using the Java Expert System Shell which is called as JESS. To implement this research work, we have used JAX-RPC.

To evaluate our proposed system, we have used Flight Booking System as a Model. The purpose of this implementation is to study whether the consumer is able to identify a right and suitable flight or not. To implement our work, we have defined various fuzzy classification rules such as cheap, comfortable, best airline and etc for integrating airlines databases having various size records ranging from 5 to 60. To evaluate the Discovery Performance of Web Services, we have implemented and compared the results of our proposed Fuzzy-XDDS with existing Fuzzy Matchmaking Approach in terms of Imprecision Rates, Precision Rate and Recall Rate. These measures have calculated as follows.

\[
\text{Precision} = \frac{\text{[Relevant Web Services]} \cap \text{[Retrieved Web Services]}]}{\text{[Retrieved Web Services]}}
\]

\[
\text{Recall} = \frac{\text{[Relevant Web Services]} \cap \text{[Retrieved Web Services]}]}{\text{[Relevant Web Services]}}
\]

\[
\text{Imprecision Rate} = \frac{R_{user} \neq R_{searched} / \text{TSS}}{}
\]

Here \( R_{user} \) is the number of services required by users and \( R_{searched} \) is the number of searched services and TSS is the Total Search Space.

A. Results and Discussions

From the Fig. 7, it is established that the Precision Rate of our proposed Fuzzy-XDDS is higher than that of the existing Fuzzy MatchMaker. It is also revealed that he Recall Rate of
proposed Fuzzy-XDDS is higher than that of the existing Fuzzy MatchMaker, which is shown in the Fig. 8.

From the Figure 9, it is noted that the Imprecision Rate of the proposed Fuzzy-XDDS is less than that of Fuzzy Matchmaker and we can conclude that the proposed Fuzzy-XDDS is able to generate an acceptable result.

VI. CONCLUSION AND FUTURE WORK

This work implemented and studied a few Fuzzy based Services Matchmaking Frameworks such as Fuzzy Matchmaking for Web Services, A Moderated Fuzzy Matchmaking for Web Services, Adaptive Fuzzy-Valued Service Selection, A Hybrid Approach to Semantic Web Services Matchmaking and Fuzzy Matchmaking Approach for Semantic Web Services thoroughly and revealed that the Fuzzy Matchmaking Approach for Semantic Web Services performing well in terms of Precision Rate and Recall Rate for selecting Web Services as compared with existing models. However, this work does not focus both the Services Ontology and Matching Services across domain boundaries while Ranking and Selecting Services and hence this model couldn’t provide required QoS to the Web Users. To address these major issues, this research work has proposed an efficient Fuzzy based Cross-Domain Services Matchmaker (Fuzzy-XDDS) which is extending Service Discovery across Domain Boundaries. This model also has the ability to discover services across domain boundaries in a Scalable and Flexible manner as this is designed for Cross-Domain Service Discovery based on the principles of Modularity, Interoperability and Scalability. The experimental result shows that our proposed Fuzzy-XDDS performing well as compared with Fuzzy Matchmaker in terms of Precision Rate, Recall Rate and Imprecision Rate.

ACKNOWLEDGMENT

I wish to thank Dr. D. Manimegalai for her valuable support and guidance throughout the work and Prof. Jayasri for her valuable suggestions in every part of the work and all other colleagues who used to comment on the idea to carry over this research work. I also wish to acknowledge the valuable contributions of the numerous researches and industries, who participated in this research area by providing varieties of tools and techniques.

REFERENCES


