International Journal of u- and e- Service, Science and Technology
Vol.9, No. 9 (2016), pp.139-154
http://dx.doi.org/10.14257/ijunesst.2016.9.9.15

SWEFT: Semantic Web Service Engine for Telecommunication – An Automatic Discovery and Composition through Genetic Approach

Lakshmana Kumar Ramasamy[1*] and M.S.Irfan Ahmed[2]

[1]*Department of Computer Applications,Hindusthan College of Engineering and Technology, Coimbatore, India*
[2]*Department of Computer Applications,Nehru Institute of Engineering and Technology,Coimbatore,India*
[1]*research.laksha@gmail.com,*[2]*msirfan@gmail.com*

Abstract

The web service developed by telecommunication domain is ineffective since they are demonstrated by syntactic description rather than semantic. The motivation of the research is to have semantic description with existing web services, and provides discovery, composition and invocation of web services automatically. The objective is to identify the discovery and composition concerns and devise a compositional approach that covers all concerns. So a new prototype named Semantic Web Service Engine for Telecommunication which automatically discover and composite a web service was proposed, enables semantic through upper ontology and maps Web Service Description Language to Ontology Web Language-Semantic. For composition, a genetic algorithm was proposed which can solve problems with great distinctiveness. This approach automatically discovers and generates the required composite semantic web services and considers all identified concerns concurrently, improves the accuracy for the service discovery and unifies the semantic representation of telecommunications without human intervention.

Keywords*: Service Discovery, Service composition, Semantic web services, Quality of service, Genetic algorithm*

1. Introduction

The World Wide Web (WWW) has been progressing into a largest scattered system to share information. The web service is made popular due to the development of WWW through different web protocols and service oriented architecture (SOA) approach. A web service is discovered by the service consumer by communicating the UDDI, the service registry where the web service description is written in Web Service Description Language (WSDL).The communication messages are carried over as payload of different web protocols like HTTP, SMTP *etc.,* by Simple Object Access Protocol. Thus web service control the whole connectivity and interoperability of heterogeneous applications and services. The increase in popularity of web services has triggered the problem to find and to compose the appropriate services from a public domain that exist with thousands of such services. In order to discover and compose appropriate web services, user has to search among the available set of abstract descriptions in the repository. Organization implementation is abstracted in the web service description. Human intervention is required to discover, compose, invocate and monitor the services since web services are deficient of machine interpretable semantics. And this human intervention restricts the

[*]Corresponding Author

ISSN: 2005-4246 IJUNESST

usage of services in complex business environment, where the former mentioned processes are to be automated .The idea of annotating services with ontology semantic descriptions enables the semantic web services that provides relaxation to these restriction [1].

Semantic enhanced web services provide a novel model for distributed computing. The mark-up is used by Semantic web that helps the data readable by machines. Through semantic web, computers have an ability to find, execute, composite and provide the information in an understandable way. The Semantic enhanced web services enables any computer that runs an application has ability to search, select and requests the needed web service. The architecture of the Web service has three components, service provider, service registry and service consumer. The syntactic Web Services used a Description Language which is an XML language, which acts as an intermediate that helps to find the right service in aspects of its input, output, Precondition and execution. This XML language helps the user to find the right service, provides the service location, and also the protocol for invoking the service, and it also helps to formulate the data to be sent to the service. The importance for bringing down the semantic enhancement for the Web Services is vital which helps to enable the automatic discovery and composition of Web Services without human intervention. There is a need for semantic web as it helps to represent and exchange the information for the web service and enables to, facilitate the execution of annotations on the web automatically .The descriptions on the semantic web provides the relationship between information over the Internet [1]. Ontology provides a technological structure which is a formal description for a collaborative conceptualization.

OWL-S provides an upper ontology helps to represent the functionalities and efficiency of web services in OWL. The OWL-S aids to discover invocate and composite the web services automatically without human intervention. [2]

In this paper, a new prototype was proposed named Semantic Web service Engine for Telecommunication (SWEFT) that automatically discovers and composes a web service. It enables automatic service discovery for the semantic through the upper ontology and mapped Web Service Description Language (WSDL) to OWL-S (Ontology Web Language-Semantic). This approach adopts a genetic algorithm (GA) for automatic service composition. First, the survey and analyzing the current mapping algorithm was carried out and then automatic discovery and composition efforts were identified and concerns as well as the existing gaps in the reviewed efforts. Second, the approach discusses about the mapping algorithm [19] that converts the web services annotations (WSDL) to semantic annotations (OWL-S) and automatic service discovery. Third, the prototype discusses about the automatic service composition approach based on the GA that covering all of the identified composition concerns.

The remainder of this paper is organized as follows:

Section 2 discusses related work of Semantic Mapping, Service Discovery and Service Composition. Section 3 describes the Proposed Architecture. Section 4 proposes Semantic Service Discovery. Section 5 proposes the GA-based composition approach while Section 6 presents the experiments and discussions. Finally Section 7 brings out the conclusion and the proposed future work.

2. Related Work

This section reviews past work and provides motivation for the study, including three subsections: Semantic Mapping, Service Discovery and Service composition.

Semantic Mapping and Discovery

Conversion of traditional web service description language (WSDL) to a semantic one (*i.e.*, OWL-S) is known as semantic mapping. The different approaches on semantic mapping are described as follows:

Tamer Ahmed *et. al.,* [2] proposed a mapping algorithm that provides the reconsideration for the traditional web services definition (*i.e.*, WSDL) through semantic annotations (*i.e.*, OWL-S). The proposed algorithm provides a new discovery system that depends on the semantic definitions of the web services to do its function. The algorithm uses a "local ontology repository" and "ontology search and standardization engine" as the backbone. Cassar *et. al.,* [3] suggested a method which uses probabilistic machine-learning techniques to abstract the underlying components from semantically enriched service annotations. The model is constructed using latent factors and helps to exhibit the different service annotations by a vector form. Through this model, heterogeneous service annotations can be discovered and compared with the similar homogeneous plane. The results obtained provides the scalability for service datasets which is variably large and produces a decisive way which provides the publishing and adding new services to the registry and representing those using latent factors after deployment of the system. Li *et. al.,* [4] proposed a strategy to find relationships for large-scale distributed networks semantically. The proposed model uses a reliable discovery protocol and a unique ordered knowledge abstraction. The proposed method proves that it will provide the cognizance effectively and helps to utilize the information present in the Internet.

Nayak *et. al.*, [5] proposed the service discovery based on semantic is to enhance the discovery of the elusive services which are irrelevant. At present the Web service standards format supports only keyword based search. In these scenarios, many services which can fulfill the user's requirements are not obtained. So the basic requirement for efficient service discovery is to fetch the relevant information provided in the service annotation. In such cases, the optimal solution is obtained by introducing semantics in the present Web Service Description Language (WSDL). Kumar, Lakshmana R *et. al.*, [6] proposed a technique to enable the semantic to the Web Service through the upper ontology (*i.e.*, OWL-S) and have mapped OWL-S to UDDI registry. The proposed approach has also discussed about the issues faced while mapping OWL-S into UDDI registry. The given approach also provides the solution for improving the accuracy of the telecommunication network services description, discovery and matching, unifies the semantic representation of telecommunications network and Internet services. Khan *et. al.,* [8] investigated an application that moves from single tenant to multi-tenancy, it extensively increases the volume of homogenous as well as heterogeneous services, which makes the service discovery much more complicated. The proposed framework integrates Web Service Execution Environment (WSMX) and Internet Reasoning Service (IRS-III). Finally, the approach demonstrates that the required time and human efforts can be reduce in service discovery. Ngan *et. al.*, [9] provides in depth analysis of semantic Web service discovery, highlights the modern approaches, they have formed the semantic formalisms as the key, and they have used standards to perform a finding. For forming a model for semantic web service [26] discovery, they described the key factors and conditions used in agent-based computing. The proposed approach provides the research challenges to be focused for next-generation service discovery through dynamic multi-agent systems in composite environments.

Automated Semantic Service Annotation with Machine Learning, a mapping tool has been introduced byJohnston and Kushmerick [10]. WSDL file is converted as OWL-S file using the mapping tool ASSAM. ASSAM is confined to certain following boundaries. The tool does not provide any organization for the ontology in use which results more concepts. From the list, finding the possible concepts depends on the text search not on the meaning .And therefore the users are given with the option list of concepts that are not ranked by importance. Paolucci *et. al.,* [11] conversation of WSDL to OWL-S provides a transformation between WSDL and OWL-S. In this work, conversion of all XSD complex types and generated concepts and properties for each type takes place. This is a blind conversion of XSD types to concepts [1]. There is also no dependency or relationship between these concepts. This conversion leads to the use of many concepts and thus provides the loss of semantic web meaning. Sangers *et. al.,* [12] the natural language processing techniques was used by Semantic web services discovery. This helps to match the process between the needs of the user and semantic web service description through Web Service Modeling Ontology formalism. Service discovery is the most important task in the Web service model. Service discovery is the process of finding Web services provider locations which satisfy specific requirements of the user. So many methods are proposed for Semantic Web Service Discovery. Some of the methods are described as follows: Nawaz *et. al.,* [13] proposed a Publish Subscriber model for web service discovery. This model is divided into two phases - subscription phase and notification phase. In the subscription phase, a user registers in a registry and notifies the required web services. In the notification phase, a new web service is hosted on the registry. When a subscriber subscribes, the information along with the user's location and the requirements are stored in subscription knowledge base. This data stored is in the OWL format can be used later for matching services. The matched web service is selected by matching user needs to OWL-S annotations the process of matching is done in one of the six steps as Exact, Plug-In, Subsume, Enclosure, Unknown and Fail [22]. Then, OWLS description for newly created web services is added to the matching subscription types. Guo *et. al.,* [14] suggested a Layer based approach for semantic web service discovery. In this approach the search is divided into 3 levels by applying filters at each level. The 3 levels for matching the services are category matching, functionality matching and quality of service matching. In category matching, the category of service is stored in the Service Profile. The data is checked against the service category of the user request. If there is a match the web service is allowed to move to the next service layer. The service functionality matching degree is calculated in the service functionality matching layer. The attributes – has-Input, has-Output, has-Precondition and has-Result are checked against the user service request. The properties are defined in the Service Profile. The final work is to compute the quality of the service (QoS) matching degree. QoS is valued which is based on the response time and reliability of the service discovery system. Service Matching Degree is calculated from the above three steps and the advertisements that best suit the request are presented in the form of a list. Johnsen *et. al.,* [16] proposed a new method for web services discovery in heterogeneous network. Different networks use different protocols. Service discovery gateways are used so that the different networks can continue to use different protocols. Hetero generous communication is achieved using service discovery gateways that can translate across different service discovery mechanisms known as Service Advertisements in MANETs (SAM). It regularly queries all services in the Web Service-Directory. Available services are then checked in the gateway's local service cache. As and when a service is deleted from a domain, it is removed from the local cache.

Rajendran and Balasubramanie [16] proposed a web service discovery method based on Quality of Service (QoS) parameters. Such as are response time, availability, throughput and time. The proposed method contains an intermediate to rank the various web services available depends on the certificates received from service publishers. Two main entities in the proposed method are verifier and certifier. The service publisher phase is accountable to register, update and delete the web service information in registry. The Service publisher is obtained by the service providers with QoS values related to business and performance of web services. The discovery agent Verifies and certifies these QoS values. The service functionality is then published to the UDDI registry. The service consumer searches the UDDI registry for a service through the discovery agent. This agent aids in finding the best service available based on the QoS parameters. Time required for choosing the best web service with respect to the QoS values eventually decreases. QoS verification is the process of validating the information described in the service interface [25]. The certification process makes use of this verification result.

Semantic Service Composition

In semantic web service composition, according to the user conditions, the computers will automatically find, execute and invoke the right web service for the users to fulfill their task. The need for this automation is to perform the task over the internet without the aid of user involvement which will save the time. Rao *et. al.,* [17] Forms the model for the semantic web service discovery and composition by using the certain constraints and inconsistencies without the need of the user. This model uses Web Ontology Language (OWL) based on Graph Plan Algorithm which can provide better results and helps to check the given state can be disjunctive refinement from another state.

Preferences for planning with Golog, the agent programming is proposed by sohrabi *et. al.,* [18]. First – order language is used to express the preferences of user and the descriptions of the web services. The modified version of Golog uses the first-order language. The functional and non functional properties of web services are described by situation calculus and first order language. The author proposes the translation of semantic of OWL-S is done by situation calculus and to generate the preferred solutions the composition system uses the preferences of the user. Zhang *et. al.,* [7]. Proposed a review on Semantic Web service composition. The approach discusses about the research of Semantic Web service composition through the semantic reasoning as core by AI planning, Logical Reasoning, Graph Search and Matching methods as well as other measures. The review focuses about how to discover the accurate and efficient Web service based on semantic and how to provide the composition plan that fulfils the users' needs. In [19] the prototype builds the composite web service through model driven approach for the web service languages enhanced semantically. The prototype employs in 4 modules. The composite model is created during the first phase, which contains all the needed intelligence for the discovery and selection of the web services. In the next phase, the apt web services are processed for the discovery which purely implies on the semantic annotations. During the third phase, a detailed composite model is created. Finally, different annotations of the composition prototype are used for composing the web service.

In [20-23], an optimized framework is formed for the service composition. The approach works by taking 3 inputs, the domain specific composition rules, the description of the business objectives and the description of the business assumptions. The first step is Backward Chaining in which the chain backwards was created by composition rules from the business aims till the first stage is achieved. The next step is known as forward chaining; here services are added with the composition structure to finish the first stage. The augmentation of the earlier results helps to control the flow for the composition. Finally, it is known as Dataflow Interface where data flow is clubbed with the composition structure.

3. SWEFT Architecture

SWEFT Framework contains two engines: Service Discovery Engine and Service Composition Engine. Below figure represents the model of SWEFT Framework.

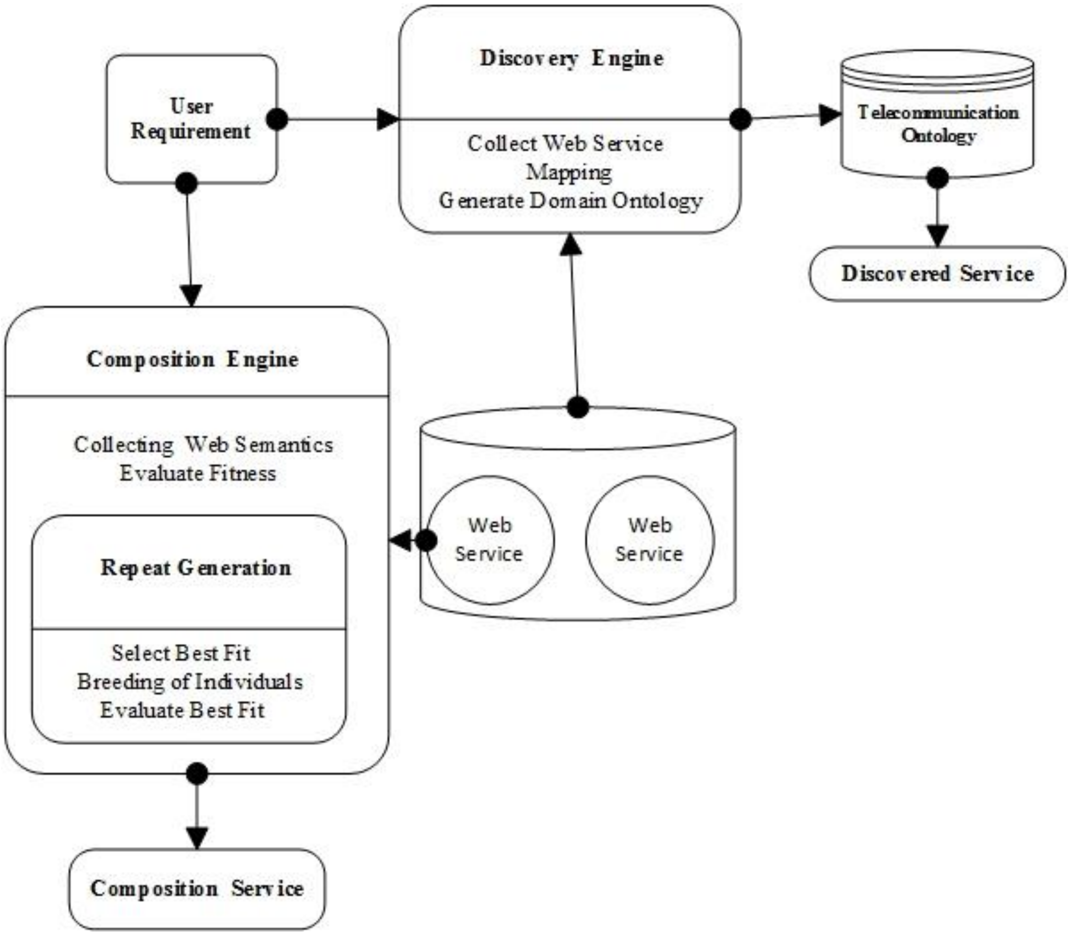

Figure 1. SWEFT Architecture

Based on the user requirements the discovery engine collects all the web service, map WSDL to OWL, generates domain ontology and discovery services. The composition engine uses genetic algorithm to compose the services.

4. Automatic Service Discovery

In semantic web service discovery, the number of Services to be executed should have the ontological annotation for each service. So the first step is mapping the WSDL files into their corresponding OWL-S files [27] which provides semantic enhancement.

Algorithm 1: Mapping of WSDL to OWL-S

Input: WSDL File
Output: OWL-S File
Load WSDL File
Read WSDL File
    If XSD type is Complex then
        Extract name of the XSD Complex type
        Create ontology class for XSD Complex type
        For each XSD Complex type
            Extract name and type of XSD element
            Create Ontology data and Object Property for XSD element
        End for
End if
Extract WSDL operation
For each WSDL Operation
Create Ontology class
End if
Generated Ontology is stored in service repository

The above algorithm is an mapping algorithm of WSDL to OWL-S. Figure 2, shows the flow of WDSL to OWL conversion.

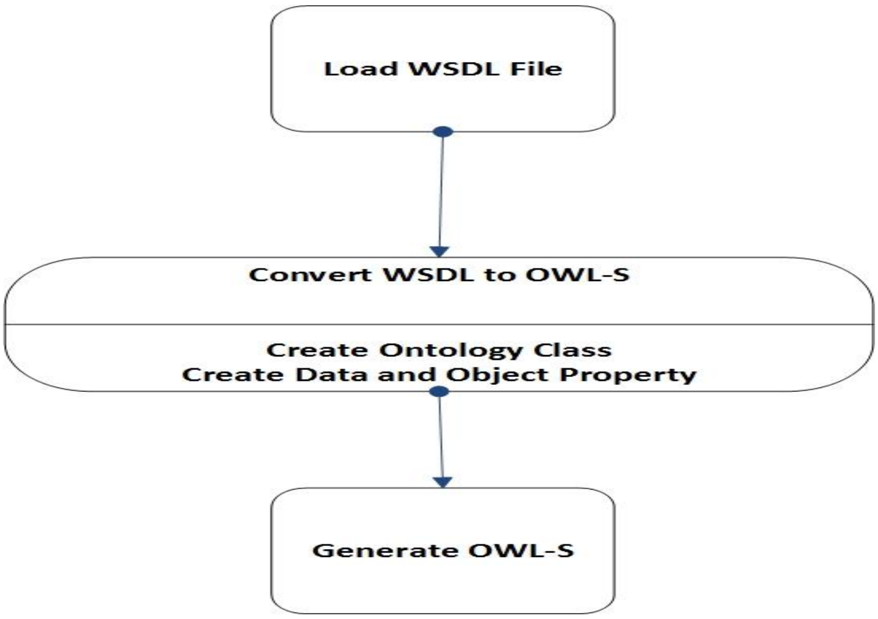

Figure 2. Flow of WSDL to OWL Conversion

The Mapping algorithm executes by using WSDL files as input from the service providers. After loading WSDL file the converter converts WSDL to OWL-S. WSDL type conversions play the vital part in mapping algorithm, especially XSD types. In general there are 2 categories: Simple XSD types and complex XSD types. The simple XSD are converted to OWL-S as they are. But the complicated XSD are translated into OWL ontology concepts using Protege OWL Java API. It extracts all the XSD Complex types, for each type creating the OWL class. It extracts all the XSD elements from XSD Complex type, for each element creating the OWL Data and Object Property. The generated ontology is stored to service repository. Figure 4, shows the genetic algorithm for semantic service composition.

5. Automatic Service Composition

This section explains genetic based web service composition. This paper does not attempt to propose a new type of or design for the genetic algorithm (GA). In this work, the prototype adopts the GA and follows its typical process to solve an application problem that's define. To overcome the encountered problems, the approach discovers that the design must include four independent fitness functions, and thus, a special selection mechanism is used as the novelty of the problem.

5.1. Introduction to the Genetic Algorithm (GA)

The GA is the popular method in evolutionary computation and is commonly used to find the optimal or the near-optimal solution for problems with notable large search spaces [22]. The GA applies concepts from biological principles to simulate the evolution process in nature, namely, ''survival of the fittest'' or ''natural selection''. In the GA evolution, a candidate solution to the problem to be solved is known as a ''chromosome'' or ''individual''. A chromosome consisting of genes and is encoded by several formats, with Bit string encoding being the most common. There are a number of chromosomes in each GA generation, and all of the chromosomes within a generation are referred to as the population. The number of chromosomes contained in a GA generation and the number of generations are fixed and assigned by the user of the GA at the design time. The important element in the GA is the fitness function, used to find the fitness and goodness of the chromosomes for the problem to be solved. During evolution, the chromosomes are chosen for crossover according to their fitness values are known as parents, and the products of crossover are referred to as children. The evolution process iteratively repeats until it reaches the end condition.

5.2 Configuration and Fitness Function

The creation of chromosomes in the first generation is entirely random, but certain limitations apply to every chromosome. The first limitation is the length (size) of the chromosome (*i.e.*, the number of services in a chromosome). The degree of the tree is the second limitation. In the selection mechanism, the operator picks the parent chromosomes for crossover and mutation from the current population. After passing through these three operators, the generated children become the members of the next generation. Once the selection operator operates twice, two individuals (parents) are received. The next step is to find the crossover decided by the crossover probability between two parent chromosomes. If the crossover probability for the parents does not copulate, they will move to the next step. To perform crossover, the crossover operator determines a node randomly for the parent trees at level 2 and exchanges these nodes. Each chromosome from the crossover stage must determine whether it needs to mutate, depending on a mutation probability [24]. The crossover probability is typically higher than the mutation probability; otherwise, the mutation probability is always a small value because the GA turns into a random search if mutation occurs excessively. To perform mutation, at the level 2 the mutation operator determines a tree node and randomly exchanges this node with another activity. The four types of mutation patterns are described as follows: (1) a structure exchanges with another structure, (2) a service mutates into another service, (3) a service becomes a structure with services, and (4) a service or structure disappears.

In general, the most crucial element in the GA is the fitness function, determines the performance and efficiency of the GA, whether it is successful or not. The proposed and conventional GAs difference is that the proposed GA contains four autonomous fitness functions for scoring the chromosomes. The fitness value from one of the four fitness functions evaluates a chromosome in terms of its unique viewpoint and implies a different meaning. In the traditional GA application, if several objective or fitness functions emerge, they are typically normalized (merged) into a single function.

Algorithm 2: Genetic Algorithm for Service Composition

Sort all the web services according to their Qos value
Initialize chromosomes
t=1; maxIteration=100;
While (t<maxIteration)
Selection
Cross Over
Mutation
Compute Fitness Value
t=t+1;
End While
Select the best from the population
Composite Web service

6. Experiments

The experiments have accomplished several experiments to evaluate the implemented algorithm. The programming language used to do the evaluation is Java and the algorithm is executed on desktop PC with Pentium 2.2 GHz dual core CPU and 3 GB of RAM. The useful JAVA API for XML Processing (JAXP) known as the Document Object Model (DOM) is adopted to bi-directionally transform the XML document as an in-memory tree structure (and vice versa), which matches the user needs. Another exploited API known as OWL-API is used to operate the OWL-DL ontology and perform inference on it. Table 1, lists the experimental environment at a glance.

Table 1. Environment of the Experiments

Hardware	Pentium 2.2 GHz dual core CPU and 3 GB of RAM
Operating System	Windows 7
Programming language	Java 7.0
JAVA APIs	JAXP DOM and OWL-API
Service Domain	Telecommunication

Mapping Results

The results for Semantic Mapping of Concepts are represented below. Figure 3, 4, 5, shows the implementation of semantic mapping

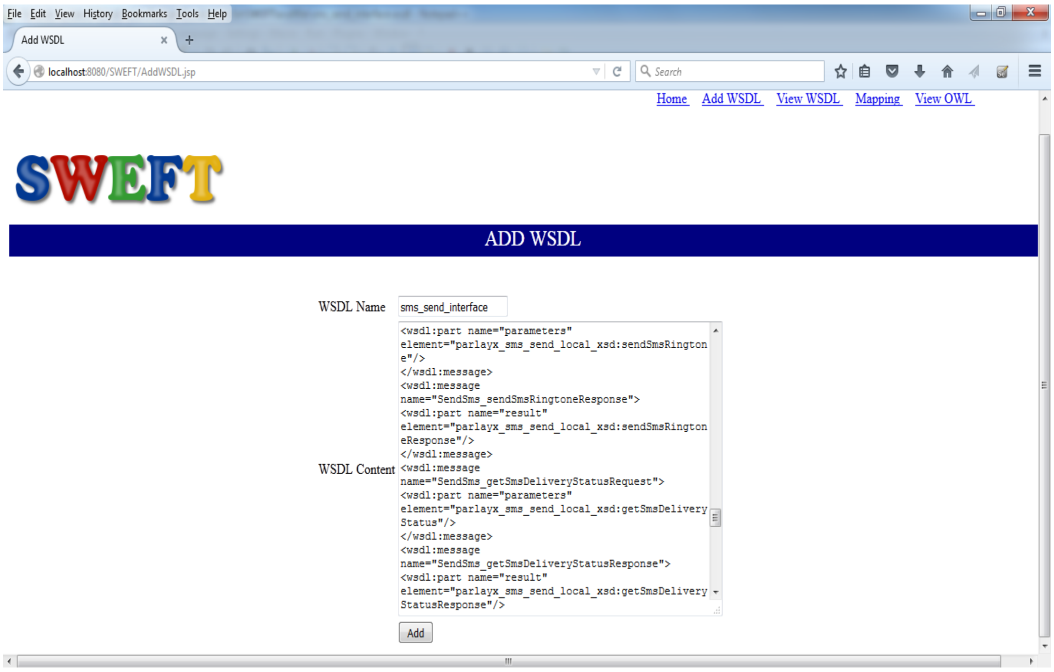

Figure 3. Semantic Web Service Search Engine WSDL Add Page

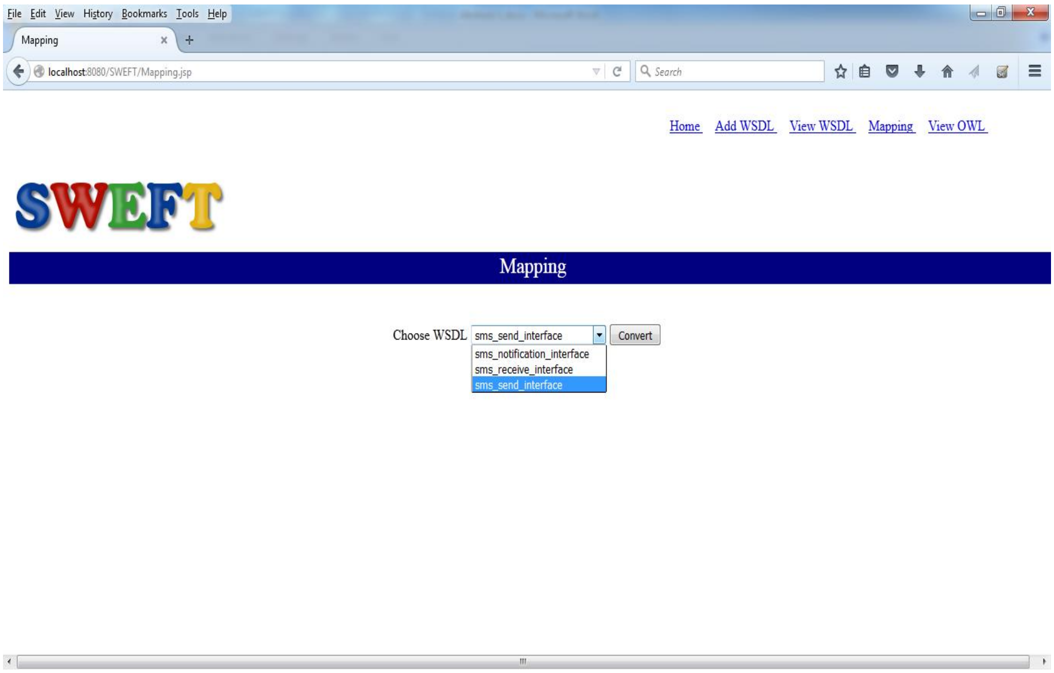

Figure 4. Conversion of WSDL to OWL File

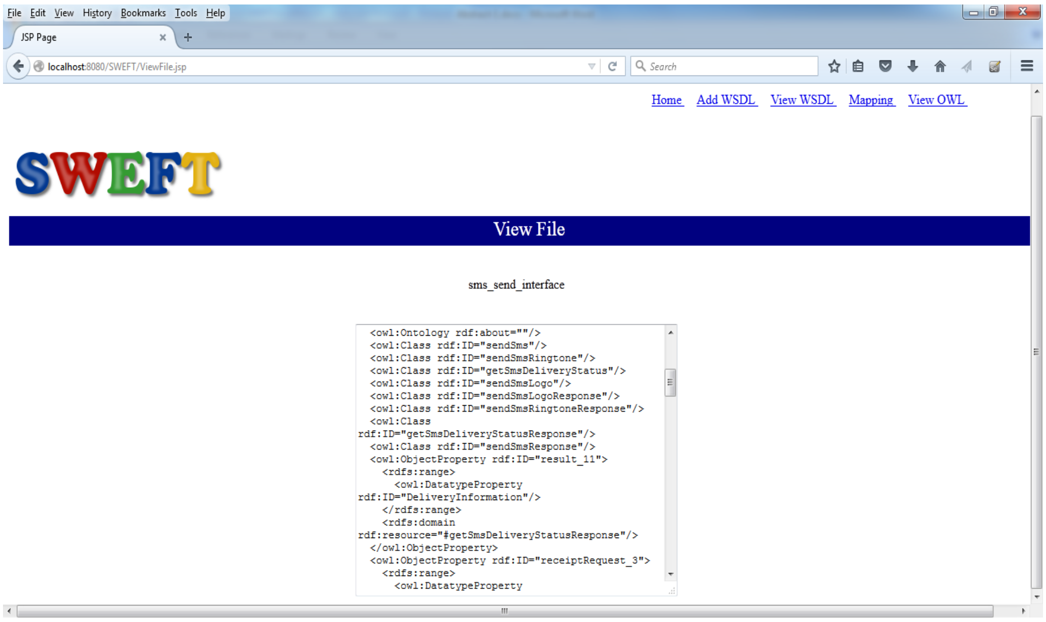

Figure 5. Mapped OWL File

Service Composition Results

Table 2, shows the genetic algorithm parameters. For different execution the prototype have to change the value. The population size is varied from 50 to 1000. The crossover and mutation probability is varied from 0.1 to 0.9. Based on these parameters the execution time and fitness values are changed. The algorithm is executed number of times with different parameters.

Table 2. Initial GA Parameters

Crossover Probability	0.7
Mutation Probability	0.1
Population Size	50

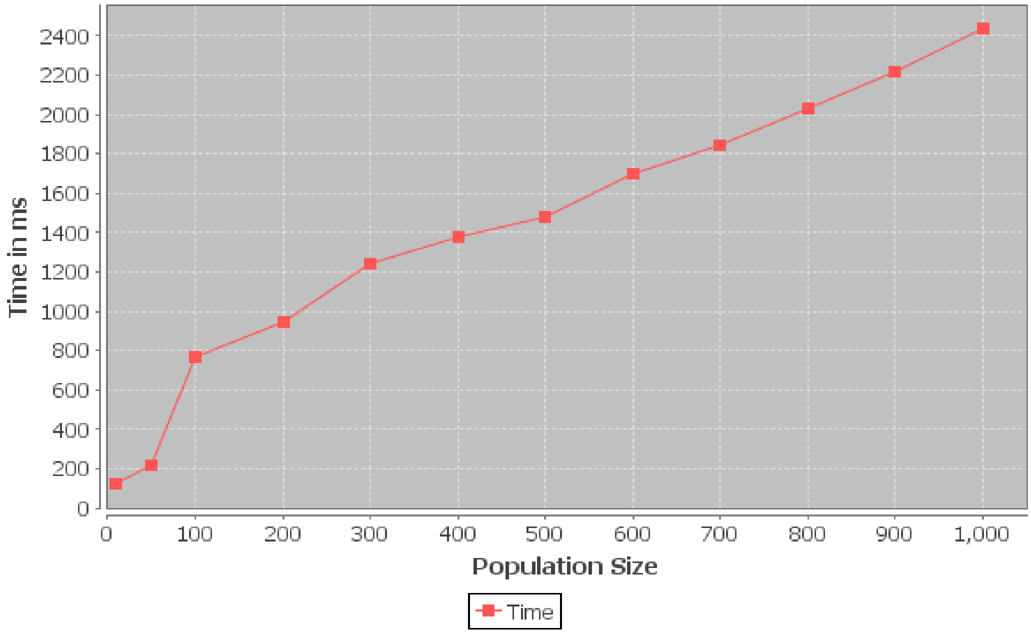

Figure 6. Execution Time for Different Population

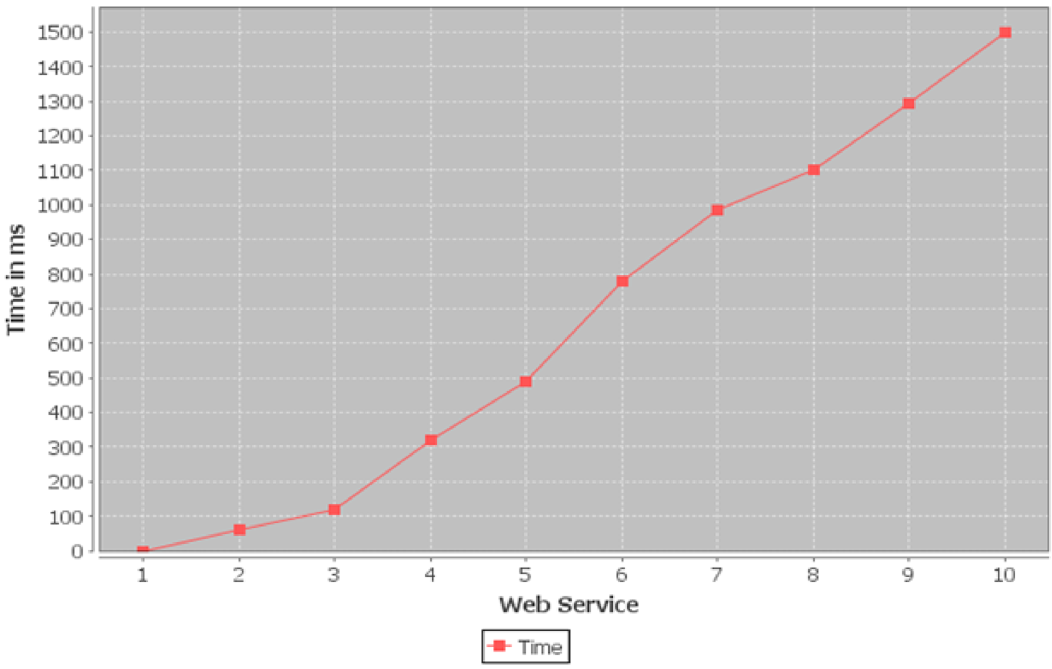

Figure 7. Execution Time for Different No of Web Service

Figure 6, and 7, shows the execution time of proposed approach with different population size and different number of web service.

Figure 6, shows exponential increase of the execution time versus the population size. Varying population size the execution time also increases. The y axis represent the execution time and x axis represents various sizes of populations (10, 50,100,200 up to 1000).

Figure 7, shows increase of the execution time versus the number of web services.

Through Figure 7, the execution time is largely different in the case of different service numbers. The using time of becoming large as the number of services increases.
Figure 8, shows the fitness value for 50 populations

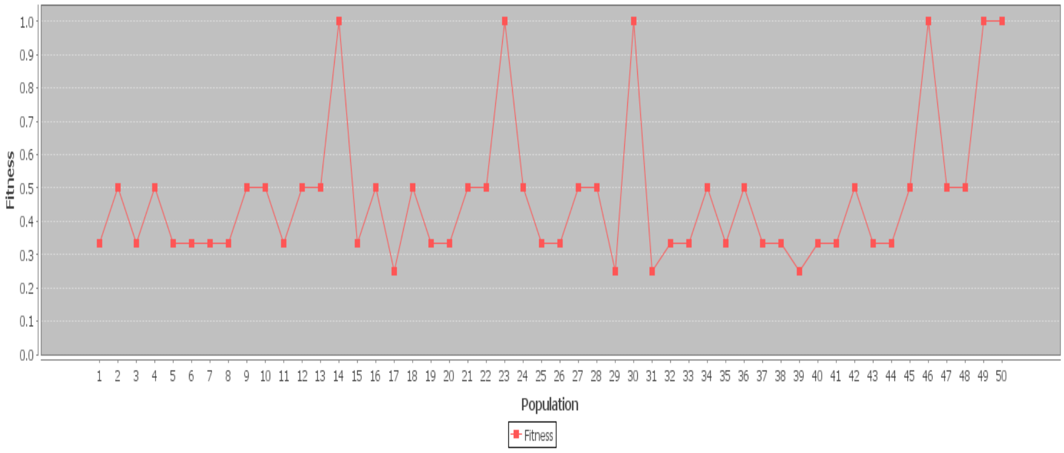

Figure 8. Fitness Value

Initially, in the proposed approach it’s necessary to set the value of the maximal number of generations to 10 and then to calculate the fitness for the cases where the size of the initial population is: 50 and increase the population size up to 1000 for each population size and the mutation rate is fixed as 0.1, 0.2 and 0.3.

7. Conclusion and Future Directions

Every day, the semantic web comes with different advent. This method provides a unique approach of mapping the semantic descriptions with the Web Service Description files. The important aspect of this algorithm is that, it provides the automated mapping scheme which is simple and proves that it reduces the time interval. This paper also proposes GA for thoroughly automating the design-time service composition process. The purpose of this paper is to identify a realistic automatic approach that covers all of the relevant composition concerns. It shows how the user could find the most suitable web service composition using genetic algorithms. Some new ideas for generating chromosomes, selection and crossover functions are proposed. The obtained results demonstrate that the advantages of the new ideas are welcomed to overcome the local optimums.

Therefore, it concludes that using genetic algorithms in such problems has improved computation time. The proposed approach is GA-based though is quite different from conventional GA-based approaches because it contains four distinct fitness functions for addressing the modeled composition problem.

In the proposed research, we worked on semantic web service discovery and composition in the wired environment with constant clients. In the future, we have a plan to extend this to non-wired environment. More over the prototype proposed can’t be implied for the mobile services. There is a huge environment changes when compared with non-wired infrastructure such as bandwidth, communication between client and server, issues on power supply. So we strongly believe that it provides the new path to implement semantic web services to the wireless world.

References

[1] A. B. E. Sisi, "Fast Mapping Algorithm from WSDL to OWL-S", International Journal of Information Technology and Computer Science (IJITCS), vol. 6, no. 9, **(2014)**, pp. 24.
[2] T. A. Farrag, A. I. Saleh and H. A. Ali, "Toward SWSs discovery: mapping from wsdl to owl-s based on ontology search and standardization engine", Knowledge and Data Engineering, IEEE Transactions on, vol. 25, no. 5, **(2013)**, pp. 1135-1147.
[3] G. Cassar, P. Barnaghi and K. Moessner, "Probabilistic matchmaking methods for automated service discovery.Services Computing", IEEE Transactions on, vol. 7, no. 4, **(2014)**, pp. 654-666.
[4] J. Li, H. Wang, S. U. Khan, Q. Li and A. Y. Zomaya, "A fully distributed scheme for discovery of semantic relationships. Services Computing", IEEE Transactions on, vol. 6, no. 4, **(2013)**, pp. 457-469.
[5] C. Nayak, S. Nayak and S. Ghosal, "Semantic Driven Web Service Discovery (SDWSD) System", International Journal of Computer Applications, vol. 69, no. 23, **(2013)**.
[6] L. R. Kumar and M. I. Ahmed, "SEMANTIC ENHANCED UDDI USING OWL-S PROFILE ONTOLOGY FOR THE AUTOMATIC DISCOVERY OF WEB SERVICES IN THE DOMAIN OF TELECOMMUNICATION", Journal of Computer Science, vol. 10, no. 8, **(2014)**, pp. 1418.
[7] J. Zhang, P. Lin, P. Huang, B. Kuang and H. Huang, "Review on Semantic Web Service Composition", Advances in Information Sciences and Service Sciences, vol. 5, no. 8, **(2013),** pp. 1172.
[8] A. N. Khan, S. Asghar and S. Fong, "Evaluation of semantic web services: before and after applying in telecommunication", Journal of Emerging Technologies in Web Intelligence, vol. 3, no. 2, **(2011)**, pp. 120-135.
[9] L. D. Ngan and R. Kanagasabai, "Semantic Web service discovery: state-of-the-art and research challenges", Personal and ubiquitous computing, vol. 17, no. 8, **(2013)**, pp. 1741-1752.
[10] A. Heß, E. Johnston and N. Kushmerick, "Assam: A tool for semi-automatically annotating semantic web services", In The Semantic Web–ISWC 2004, Springer Berlin Heidelberg, **(2004)**, pp. 320-334.
[11] M. Paolucci, N. Srinivasan, K. P. Sycara and T. Nishimura, "Towards a Semantic Choreography of Web Services: from WSDL to DAML-S", In ICWS, **(2003)** June, pp. 22-26.
[12] J. Sangers, F. Frasincar, F. Hogenboom and V. Chepegin, "Semantic web service discovery using natural language processing techniques", Expert Systems with Applications, vol. 40, no. 11, **(2013)**, pp. 4660-4671.
[13] F. Nawaz, K. Qadir and H. F. Ahmad, "SEMREG-Pro: a semantic based registry for proactive Web service discovery using publish-subscribe model", In Semantics, Knowledge and Grid, 2008. SKG'08. Fourth International Conference on, IEEE, **(2008)** December, pp. 301-308.
[14] G. Wen-Yue, A. H. Cheng and C. Hong, "Semantic web service discovery algorithm and its application on the intelligent automotive manufacturing system", In Information Management and Engineering (ICIME), 2010 The 2nd IEEE International Conference on, IEEE, **(2010)** April, pp. 601-604.
[15] F. T. Johnsen, T. Hafsøe, A. Eggen, C. Griwodz and P. Halvorsen, "Web services discovery across heterogeneous military networks", Communications Magazine, IEEE, vol. 48, no. 10, **(2010)**, pp. 84-90.
[16] T. Rajendran and P. Balasubramanie, "An optimal agent-based architecture for dynamic web service discovery with qos", In Computing Communication and Networking Technologies (ICCCNT), 2010 International Conference on, IEEE, **(2010)** July, pp. 1-7.
[17] J. Rao, D. Dimitrov, P. Hofmann and N. Sadeh, "A mixed initiative approach to semantic web service discovery and composition: SAP's guided procedures framework", In Web Services, 2006. ICWS'06. International Conference on, IEEE, **(2006)** September, pp. 401-410.
[18] S. Sohrabi, N. Prokoshyna and S. A. McIlraith, "Web service composition via the customization of Golog programs with user preferences", In Conceptual Modeling: Foundations and Applications, Springer Berlin Heidelberg, **(2009)**, pp. 319-334.
[19] R. Grønmo and M. C. Jaeger, "Model-driven semantic Web service composition", In Software Engineering Conference, 2005. APSEC'05. 12th Asia-Pacific, IEEE, **(2005)** December, pp. 8.
[20] L. Zeng, A. H. Ngu, B. Benatallah, R. Podorozhny and H. Lei, "Dynamic composition and optimization of web services", Distributed and Parallel Databases, vol. 24, no. 1-3, **(2008)**, pp. 45-72.
[21] L. Aversano, M. D. Penta and K. Taneja, "A genetic programming approach to support the design of service compositions", International Journal of Computer Systems Science & Engineering, vol. 21, no. 4, **(2006)**, pp. 247-254.
[22] D. Mukhopadhyay and A. Chougule, "A survey on web service discovery approaches", In Advances in Computer Science, Engineering & Applications, Springer Berlin Heidelberg, **(2012)**, pp. 1001-1012.
[23] G. Baryannis and D. Plexousakis, "Automated Web Service Composition: State of the Art and Research Challenges", ICS-FORTH, Tech. Rep, 409, **(2010)**.
[24] Y. Syu, Y. Y. FanJiang, J. Y. Kuo and S. P. Ma, "Towards a genetic algorithm approach to automating workflow composition for web services with transactional and qos-awareness", In Services (SERVICES), IEEE World Congress on, IEEE, **(2011)** July, pp. 295-302.
[25] M. A. Amiri and H. Serajzadeh, "QoS aware web service composition based on genetic algorithm", In Telecommunications (IST), 2010 5th International Symposium on IEEE, **(2010)** December, pp. 502-507.
[26] S. Rho, N. Chilamkurti and K. E. Defrawy, "Agent societies and social networks for ubiquitous computing", Personal and Ubiquitous Computing, vol. 17, no. 8, **(2013)**, pp. 1667-1669.

[27] G. Wen-Yue, A. H. Cheng and C. Hong, “Semantic web service discovery algorithm and its application on the intelligent automotive manufacturing system”, In Information Management and Engineering (ICIME), 2010 The 2nd IEEE International Conference on, IEEE, **(2010)** April, pp. 601-604.

Authors

Lakshmana Kumar Ramasamy, He is currently working as Assistant Professor cum Technical Trainer in Hindustan College of Engineering and Technology. He is research scholar in Anna University, Chennai in the area of Semantic web Services. He has published several papers in National and International Conferences. He holds membership in several bodies like IEEE, ACM, International Association of Computer Science and Information technology, Computer Science Teachers Association, University of New South Wales Australian School of Business Society, British Computer Society –Information Retrieval Specialist Group. His areas of specializations are Distributed Computing ,Semantic Web Services and Java technologies.

M.S.Irfan Ahmed, He received his PhD in Computer Science and Engineering with a commended status from Algappa University, India. He has been an approved research guide for various universities, which includes Anna University, Bharathiyar University, Periyar University, Periyar Maniammai University and Karpagam University. He has published around 25 papers in International Journals and Conferences. His research interest includes Trustworthy Computing, Networking and Data mining *etc.*, He has long tenure of 20 Years of teaching experience and around 4 years of Research experience. He has fetched aid from different government organizations such as DRDO, CSIR, DST, DIT, and AICTE for conducting projects, conferences, workshops and seminars. He has received the Best faculty award for the year 2012 by Nehru Group of Institutions in the discipline of computer science in the senior category.