

Framework for the Development of An Optimized Solution for Quality of Service Delivery Using Fuzzy Logic and Genetic Algorithm

U.C. Ogude & L.N. Onyejebu

Department of Computer Science

University of Port Harcourt

Port Harcourt, Nigeria

ogudecyril@yahoo.com, nneka2k@yahoo.com

ABSTRACT

Selection of Web Services is one of the most important steps in the application of different types of Web Services such as Web Service composition systems and the Universal Description, Discovery, and Integration (UDDI) registries. The more available these Web Services are on the Internet, the wider the number of these services whose functions match the various service requests. Therefore, selection of services to suit the users' requirements has become an important, challenging and time consuming problem. Selecting Web Services with optimized solution largely depends on the Quality of Service (QoS) since it plays a significant role in selecting such services. The selection process based on QoS allows the user to specify their requirements not only based on functional attributes but also on non-functional attributes. The framework for an optimal system that will provide an optimal improvement using fuzzy and genetic algorithm during the selection process is proposed. Our architecture carries out the process of selecting an optimized solution stored or contained in the QoS database (KB). It is expected that its implementation will improve the selection process and optimize Quality of Service delivery in Web Services scenarios.

Keywords: Optimized Solution, Quality of Service QoS, Fuzzy Logic, Genetic Algorithm and Databases

African Journal of Computing & ICT Reference Format:

U.C. Ogude & L.N. Onyejebu (2015): Framework for the Development of An Optimized Solution for Quality of Service Delivery Using Fuzzy Logic and Genetic Algorithm. Afr J. of Comp & ICTs. Vol 8, No. 3. Pp 228-238.

1. INTRODUCTION

Over the years, internet computing like most other areas of computer science has witnessed momentous developments. Most of these developments have been triggered by the increase in the complexity of software. Software complexity has given rise to new paradigms and among these new technologies are the Service Oriented Architecture (SOA) and Service Oriented Computing (SOC). According to Rotem-Gal-Oz [9] Service Oriented Architecture is an architectural style for building systems based on interacting coarse-grained autonomous components called services. Each service exposes processes and behavior through contracts, which are composed of messages at discoverable addresses. It therefore implies that the major pillar of the Service Oriented Architecture (SOA) is a Service. Web Services are interfaces that describe a collection of operations that are network-accessible through standardized web protocols. The word service is used interchangeably with web service since services are being deployed on the web. A service according to [4] provides a specific function, typically a business function, such as booking a flight, analyzing an individual's credit history or processing a purchase order. A service can provide a single discrete function, such as converting one type of currency into another, or it can perform a set of related business functions, such as handling the various operations in an airline reservations system.

Web services are fast becoming a defacto for the implementation and deployment of applications that model both user and organizational needs. Service Oriented Computing on the other hand is seen as an emerging discipline promoting science, research, and technology, which are related to web services [3] [6]. The emergence of the Service Oriented Computing paradigm with its implicit inclusion of web services has caused a precipitous revolution in the platform of software engineering, web based selection and composition of web services, cloud computing, grid computing among others. Service computing has brought about a shift in the focus of application providers from the traditional approach of either building in-house applications or buying applications from vendors to the web approach. The major motivation and also rapid adoption of the Service Oriented Computing paradigm has been identified by Rosenberg [8] as the fact that businesses offer their application functionality as services over the internet in order to enable other companies or users to integrate and compose these business services into their applications. It is actually of economic significance when a person or organization can have access to a web service, pay for it and use it to achieve business goals.

Apart from the economic significance of using and deploying web services, several other reasons that have motivated the migration towards Service Oriented Computing and the consequent use of Service Oriented Architecture include the following: the need to integrate legacy systems into new applications, reusability of web services, scalability, flexibility and operational convenience.

2. WEB SERVICES

Web services are emerging technology that enables different applications running on different machines to exchange data. Using web service technology, an application can be implemented by web service composition by composing existing individual web services in accordance with the business process of the application. This means the application is provided to customers in the form of a value added composite web service. The long term goal of web service technology is to enable distributed applications that can be actively assembled via web service composition in accordance to changing business needs. When discussing web service technology, Quality of Service is a significant concern, as it is a critical factor which directly determines the success or failure of a web based application. An important and challenging issue of web service selection is how to deliver Quality of Service requirements. This includes user focused elements such as response time, execution cost, availability, reliability and reputation as well as how to find an optimized solution for the Quality of Service delivery. This in turn best fulfills users' expectations and achieves their satisfaction.

2.1 Service Oriented Architecture

Service Oriented Architecture can be better understood when web services are understood. Ouzzani [5] describes a web service as a piece of software that can be defined, described and discovered by eXtended Markup Language (XML) artifacts. A core concise definition of a web service is given by W3C to be a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine process able format (specifically WSDL, which stands for Web Service Description Language). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using Hyper Text Transfer Protocol (HTTP) with an eXtensible Markup Language (XML) serialization in conjunction with other Web-related standards. Figure 2.4 illustrates the various technologies involved with web services. Web Services are interfaces that describe a collection of operations that are network- accessible through standardized web protocols. When a required operation is not found, several services can be combined to get a composite service that performs the desired task. [7].

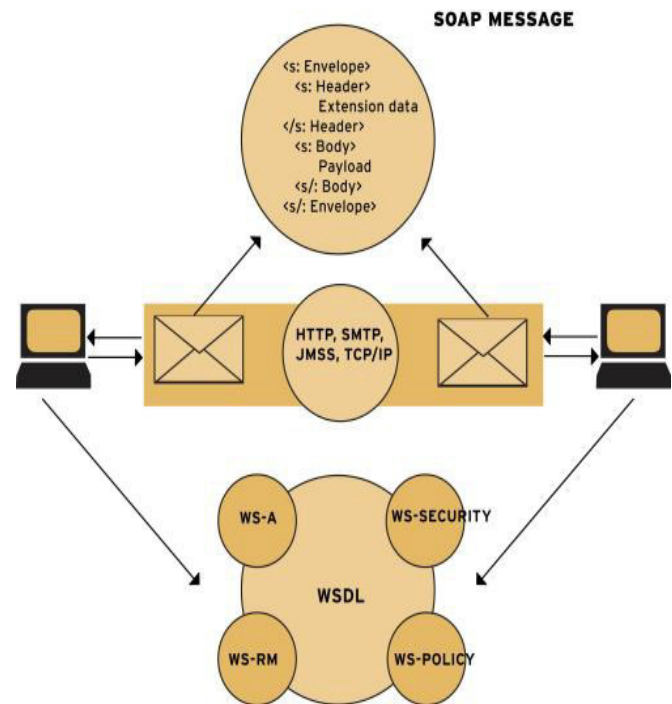


Fig. 1: A web service illustrating needed technologies [1],

Information flow between the various actors in a web service is illustrated in figure 2.5. The flow is simple; a service provider advertises his services on the repository which can either be the UDDI or is referenced in the UDDI. A client finds the needed web service based on published functionalities on the UDDI through a service provider. Sometimes, a particular service does not meet the client's needs, that is, when the client needs to compose several services. The client accesses and uses the service through the service provider.

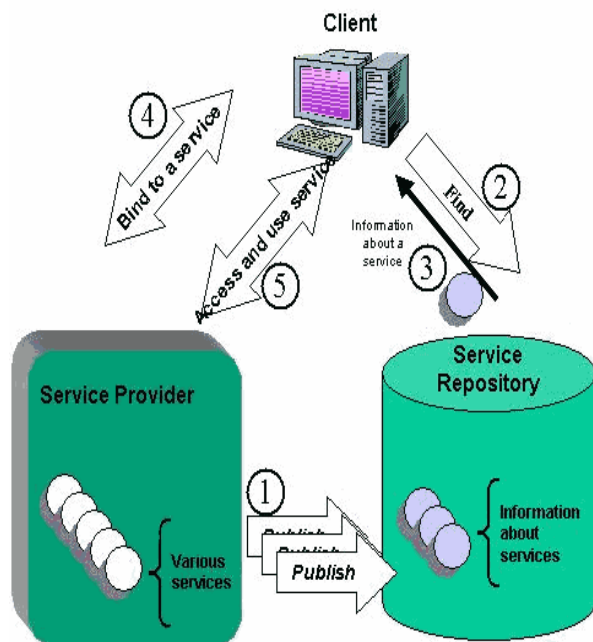


Fig. 2: Communication between the components in a web service

2.2 A Review of the Existing Systems

The number of parameters is one of the main concerns for fuzzy systems, especially when it is desired to increase the number of inputs and rules, since for the standard fuzzy system the number of parameters increases exponentially when the numbers of inputs or rules are increased, and computational complexity increases accordingly.

A detailed and in depth analysis of existing systems are reviewed in order to determine the gap to be filled, we reviewed the following:-

- (i) The fuzzy approach build or model decisions based on the rules stored or contained in the knowledge base (KB) for service selection process. Therefore, decision is based on information or data. In most real-world settings, decision-relevant information is incomplete, uncertain and imprecise.
- (ii) The number of rules increases exponentially as the number of system variables, upon which the fuzzy rules are based, is increased.

2.3 Limitations in the Existing Systems

Fuzzy systems are rule based systems (knowledge based systems). The rule base of a fuzzy system is composed of fuzzy IF-THEN rules that are similar to the rules used by humans in their reasoning.

The under listed are the drawback and limitation in the existing systems:

- (i) it is hard to obtain optimal fuzzy set due due to the number of generated rules as the number of system variables are increased,
- (ii) the fuzzy set doesn't have learning capability, it depends on a pre-defined set of rules which depends on the numbers of inputs that are to be processed, and
- (iii) the problem of defining the rule base:
- (iv) redundant rules: whose actions are covered by other rules
- (v) wrong rules: badly defined, thereby upsetting the system performance, and
- (vi) conflicting rules: worsens the system performance when co-existing with other rules in the RB (rule base).

These limitations make the system performance less efficient and effective due to fact that the service selection results always contain:

- a. Irrelevant and unjustified QoS values
- b. Non-existing web services and
- c. Wrong and inconsistent QoS values

Hence, the system performance is poor and unacceptable.

These limitations is the central driving force behind the design of a more efficient and effective system where two techniques are combined in a manner that overcomes the limitations.

2.4 Research Gap

A typical scenario to be used in illustrating the problem of Quality of Service based Selection of Web Services is drawn from the academic community attend a conference planner. This is the case of a user who plans to attend an international conference.

The user needs to carry out several tasks which would naturally include:

- i. Getting advert information for the conference,
- ii. Registering for the conference either through the bank or by a cashless transfer procedure,
- iii. Booking a flight for the conference and
- iv. Making hotel reservations and
- v. The need for car rental services.

From the illustration just given, it is obvious that one service cannot fulfill the needs of this conference attendee. The need for web service composition arises when a single web service cannot meet the desires of a web service user. This is applicable both for an individual user and for business to business and enterprise level application integration; composition of web services plays an important role.

Also suppose a large number of user web services with different criteria, such as, reputation, reliability, availability, cost and response time are available for every service component, the task of Quality of Service based service composition is to select the optimal user services for each of them.

An important and challenging issue of web service selection is how to meet Quality of Service requirements. This includes user focused attributes such as reputation, reliability, availability, cost and response time as well as how to provide optimal solution for Quality of Service results for the composites. This in turn best fulfills users' expectations and achieves their satisfaction. The objective of the web service selection is to maximize the Quality of Service values. Selected web service should have high reputation, reliability and availability whereas the cost and response time should be less. Therefore, the problem addressed in this work is to find an optimized solution for service selection bearing in mind the satisfaction of the user's Quality of Service.

3. RESEARCH DIRECTION

The aim of this study is to develop an efficient and effective system that will help users select services with respect to their preference and QoS-based optimization using fuzzy and genetic algorithm. A service design environment that is an integral part of the entire optimization and selection process will then be enabled.

3.1 The Quality of Service Model

There are many measures available for different QoS criteria; however, we consider the following five generic quality criteria for services (QoS parameters): reliability, availability, reputation, execution duration, and execution price. The QoS model that we propose is composed of five criteria as parameters for the quality model: reliability, availability, reputation, execution duration, and execution price. Each one is presented below and a difference is made between the QoS of single service and the composition.

Reliability

The quality of reliability is a measure of the service invocation trustworthiness. It is defined as the ratio between the numbers of service invocations that comply with the negotiated QoS over the total number of service invocations.

Availability

The quality of availability of a web service is the probability that the service operation is accessible. This is defined by the proportion of the service's uptime and downtime. The quality of availability is the probability that the service can be accessed and used. It means that this quality is obtained by the number of times the service answers a request divided by the number of total requests.

Reputation

The reputation quality is the measure of its trustworthiness. It depends on the user's experience using the service. Different end users can have different opinions about the same service. Reputation can be defined as the average ranking given to the service by end users [10]

Execution Duration

The execution duration measures the execution time between the moment the request is sent and the moment the results are received. It is defined as the fee to be paid to the service provider by the service requester for executing a particular service. The cost is always associated with the value of the service functionality, i.e. the more complex the function it provides, the higher the service price cost.

Execution Price

The Execution Price is the amount that a service requester needs to pay for executing a service [10] [11] [12]. .

Fuzzy System

A basic generalized layout of a fuzzy system unit can be seen in figure 2.16 which comprises four principal components: fuzzification, knowledge base, inference engine, and defuzzification.

3.2 Fuzzification

It means converting a crisp value of process variable into a fuzzy set. In order to make it compatible with the fuzzy set representation of the process state variable.

Fuzzification interface also involves the following functions:

- i. Measures the value of input variables,
- ii. Performs a scale mapping that transfers the range of values of input variables into corresponding universes of discourse,
- iii. Performs the function of fuzzification that converts input data into suitable linguistic values which may be viewed as labels of fuzzy sets.

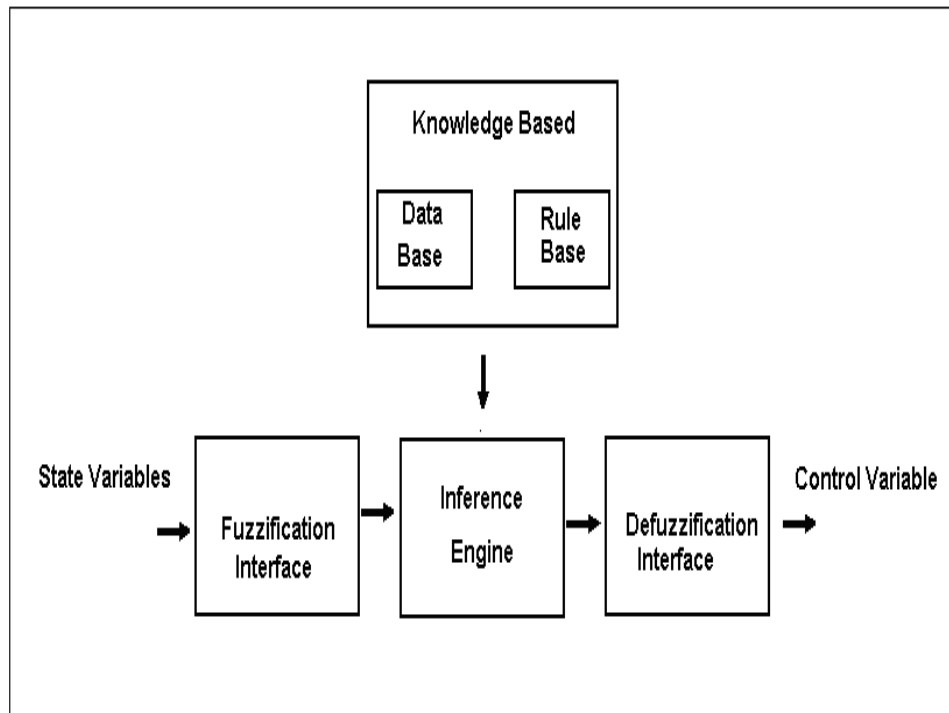


Fig. 3: Block Diagram of Fuzzy System Unit

3.3 Genetic Algorithms

The motivation for using genetic algorithms for the design of the algorithm employed in this work is due to the intractability of the mathematical model used in the problem addressed in this work. Before giving a review of methods that have been used by several authors for web service compositions and optimizations, a brief description of some important concepts surrounding genetic algorithms is given.

The choice of genetic algorithm for solving the problem described in this work was motivated by the fact that finding all the compositions for all service combinations of web services is a NP hard problem. Genetic algorithms are a heuristic method based on survival of the fittest. Genetic algorithms were invented by John Holland in the 1960s and were developed by Holland and his students and colleagues at the University of Michigan in the 1960s and the 1970s. Holland's genetic algorithm is a method for moving from one population of chromosomes to a new population by using a kind of "natural selection" together with genetic inspired operators which include crossover, selection, and mutation.

According to him some of the advantages of genetic algorithms include the following:

- i. Genetic algorithms optimize with continuous or discrete variables.
- ii. They do not require derivative information.
- iii. They can carry out simultaneously searches from a wide sampling of the cost surface.
- iv. They have the capacity to deal with a large number of variables.
- v. They are well suited for parallel computers.
- vi. They optimize variables with extremely complex cost surfaces and can jump out of a local minimum.
- vii. They have the capability of providing a list of optimum solutions, not just a single solution.
- viii. They encode the variables so that the optimization is done with the encoded variables.
- ix. Genetic algorithms work with numerically generated data, experimental data, or analytical functions.

These advantages are intriguing and produce stunning results when traditional optimization approaches fail. However, genetic algorithms are not a fix-it-all method. For optimization problems with few variables, straight forward cost function and small search spaces, traditional optimization approaches still outperform genetic algorithms [2]

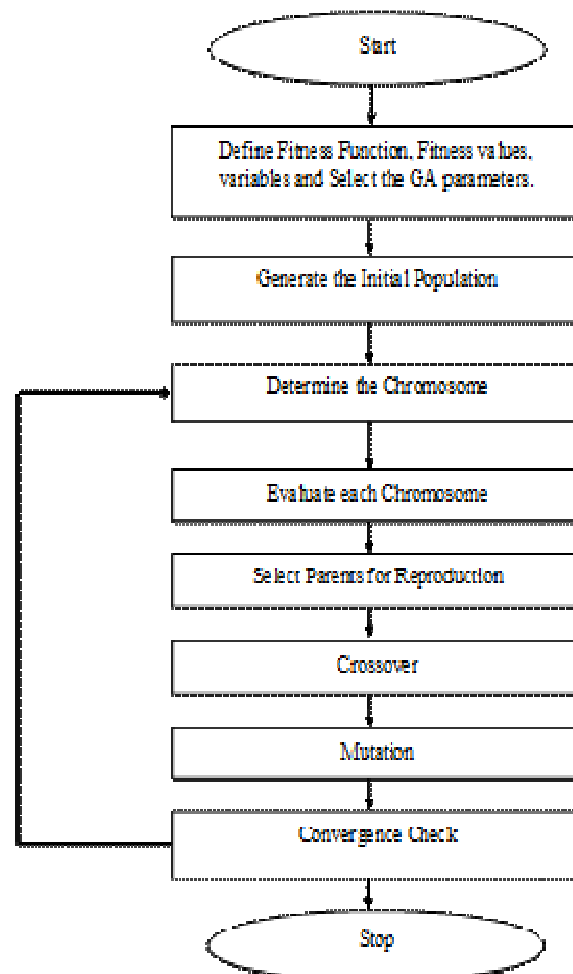


Fig. 4: Flowchart for Genetic Algorithm

4. SYSTEM ARCHITECTURES

4.1 Existing Systems Architecture

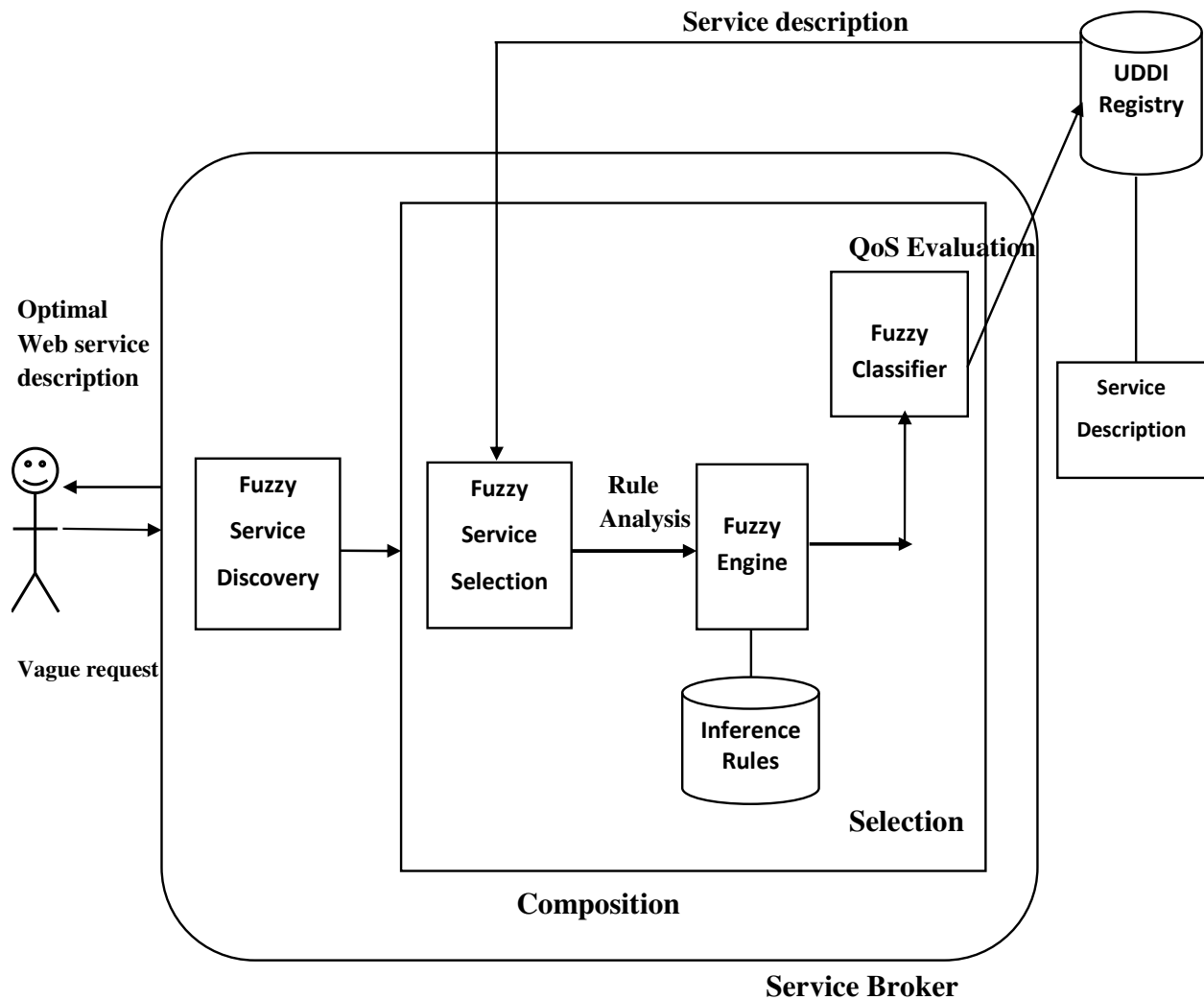


Fig. 5: Existing Architecture of the web service selection by Priya *et al.* (2014)

4.2 Analysis of Existing System

- i. The user input vague request to Fuzzy Service Discovery module to find the list of services that are available in the registry.
- ii. The registry will return the collection of service description to fuzzy service selection.
- iii. The returned service may be available or not available to the user (which depends on the service provider).
- iv. If the services are not available, because of the user preference matching the service and on seeing the service down, the user gets dissatisfied.
- v. The user move to next preference matching the services.
- vi. This process is time consuming and results in more cost and poor response time.

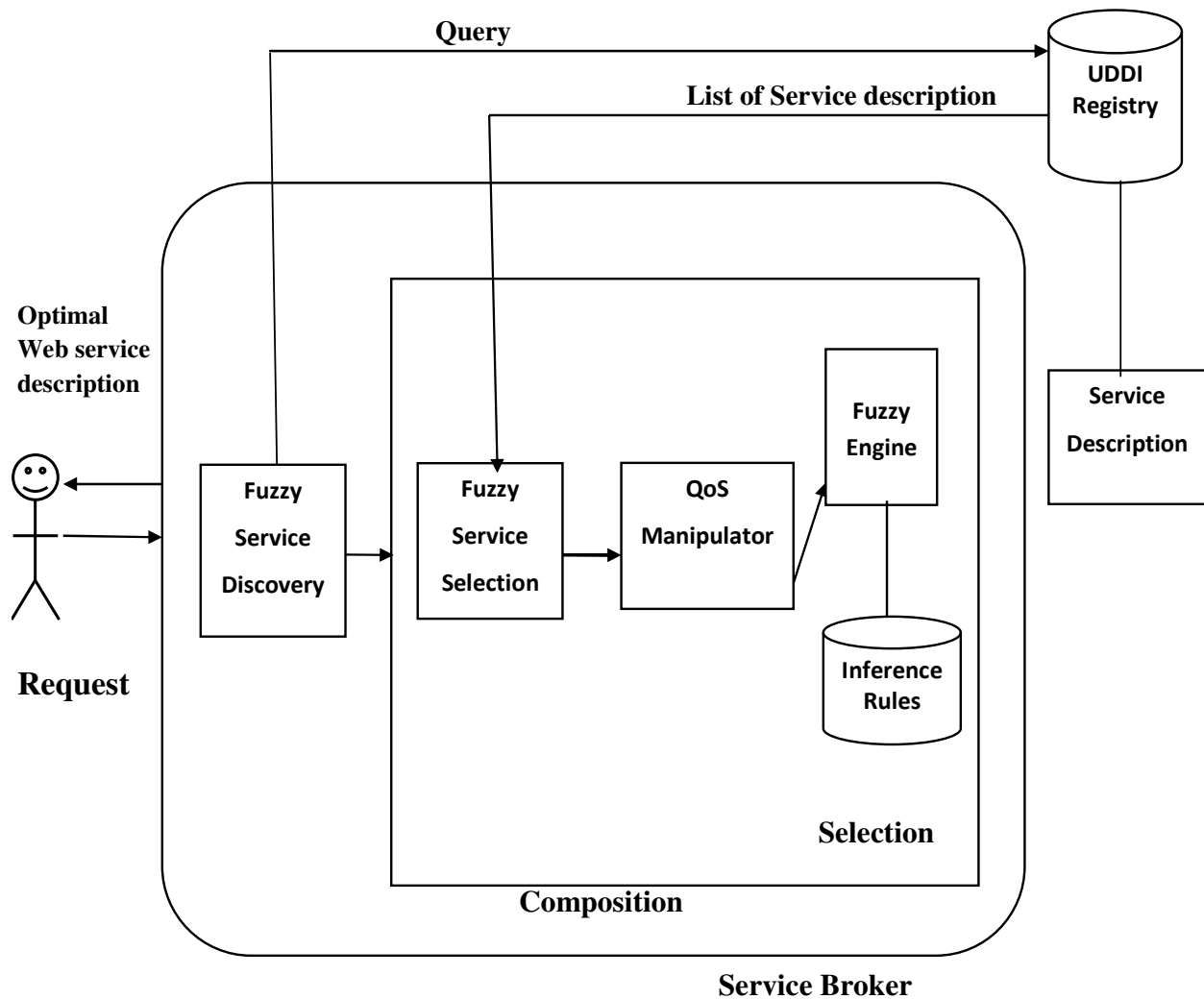


Fig. 6: Analyzing Existing Systems

4.3 The Proposed System Architecture/Framework

Several researchers have applied fuzzy approach for Quality of Service based web service selection and composition. But they did not use any optimization technique to find an optimized solution to the problem. In this work, the key point is to employ genetic algorithm (GA) in modulating, improving and optimizing the fuzzy system to find an optimized solution. Our Proposed System Exhibits two (2) important characteristics or uniqueness :-

- i. Fuzzy System:-
Model user preference and the desired QoS parameters of the service with fuzzy approach (IF-THEN rules).
- ii. GA System:-

Optimized the control structure or decisions-making and search for an optimized solution (best rules) to the given problem.

Find an optimized solution-best service to the user's preference during the selection process.

3.5.1 Proposed System Design

The proposed architecture for the web service selection must be remodeled to meet the user's request. The composition of QoS properties is done with the help of fuzzy rule and fitness function based service selection. Figure 7 shows the proposed system architecture of the optimized web service selection:

- i. End user gives vague request (query relevant services) to the Fuzzy service selection broker
- ii. Fuzzy service selector analyze the given request by passing it to fuzzy inference engine
- iii. Fuzzy Inference engine calls the knowledge base and evaluate the QoS criteria's for the listed service

- iv. Genetic Algorithms then optimizes the fuzzy system parameters.
- v. Finally, there is optimization of composition unit works based on Genetic Algorithms. Fuzzy rules, which are created based on user request together with user constraints, constitute one fitness function.
- vi. GA parameters are defined for the system. QoS criteria's related to different compositions are evaluated and an optimized solution is selected in accordance with user request and convenience.

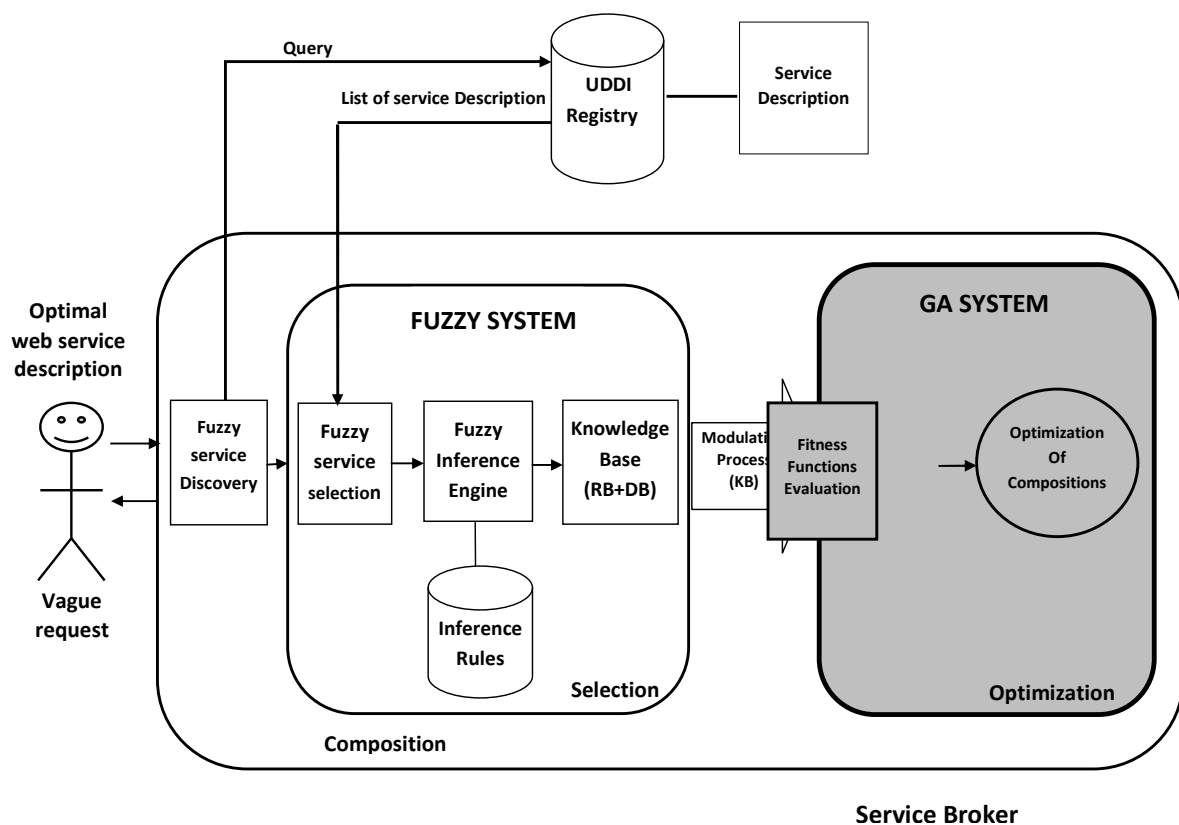


Fig. 7: Proposed Architecture of the optimized web service selection.

4.4 Optimal Improvement

The design of the parameters of a fuzzy system can be analyzed as an optimization problem. To overcome the limitations, genetic algorithm (GA) offer a possibility to solve these problem. Therefore, to overcome the drawbacks and limitations, genetic algorithm (GA) is proposed for designing and optimizing the parameters of the fuzzy system to obtain an optimized solution.

The most important key area to be optimized is the Knowledge Base (KB). The Knowledge Base of a fuzzy system consists of the Rule Base and Data Base.

- (i) Rule Base (RB) - a rule base containing a number of fuzzy IF-THEN rules,
- (ii) Data Base (DB) - a database which defines the membership functions of the fuzzy sets used in the fuzzy rules.

5. CONTRIBUTIONS TO KNOWLEDGE

To draw an accurate, reasonable and reliable conclusion in a fuzzy system, the knowledge base plays an important role and is the heart of the system, “heart – stores all data, information, rules and constraints used by fuzzy inference engine for solving difficult problems or tasks”. Once a fuzzy system is built, we are faced with a large number of parameters which need to be optimized and tuned in order to improve the system performance in terms of the results (conclusions) obtained. The key quality of this study is to achieve and obtain the optimal structure of a fuzzy system. A genetic algorithm (GA) is applied for optimization in a way that it determines fuzzy sets, their shape and their optimal number. Optimizing the entire knowledge base of the fuzzy system has a finer dimensionality and efficiency, and is therefore more likely to contain optimal solution. With these considerations, there is an obvious improvement in the system performance in terms of the results (conclusions) obtained. Therefore in this study, we seek to develop an architectural design for integrating and synthesizing of fuzzy system using a genetic algorithm. This will be able to achieve a trade-off between execution time and the optimized knowledge base thereby improving the system performance in terms of efficiency and good response time which provide the delivery of optimized solution for Quality of Service.

6. FUTURE WORK

In the paper, we presented the several quality of service parameters, existing QoS Models, their challenges/limitations as well as our proposed Architecture for optimal web service selection. The architecture constitutes user, who requests for service available in the registry, pool of service providers in the registry and a system. The system constitutes a broker and a registry. The broker employs a fuzzy and genetic algorithm system based service selection. This fuzzy and genetic algorithm system takes QoS parameter input from the details stored in the registry and finds the weight for each service by composing the QoS parameters. Future work will delve into the implementation procedure for web service selection and results from the implementation and evaluation will be provided. Automated interfaces for web service selection system, based on users' queries will also be shown.

REFERENCES

- [1] Ardagna, D., and Pernici, B. (2007). Adaptive service composition in flexible processes. *IEEE Transactions on Software Engineering*. Vol. 33(6): 369–384.
- [2] Canfora, G., Penta, M. D., Esposito, R. and Villani, M. L. (2005). An Approach for QoS-aware Service Composition based on Genetic Algorithms. *Proceedings of the Genetic and Computation Conference (GECCO'05)*, 1069-1075
- [3] Dustdar, S. and Papazoglou, M. P. (2008). *Services and Service Composition – An Introduction*. *IT Information Technology* 2: 86-92.
- [4] Ort, E. (2005). *Service-Oriented Architecture and Web Services: Concepts, Technologies, and Tools*. A white paper, Retrieved from <http://cic.javerianacali.edu.co/wiki/lib/exe/fetch.php?media=materias:soa2.pdf> on November 15 2014.
- [5] Ouzzani, M. (2004). *Efficient Delivery of Web Services*. PhD Thesis, Virginia Polytechnic Institute and State University.
- [6] Papazoglou, M. P., Traverso, P., Dustdar, S. and Leymann, F. (2007). *Service-Oriented Computing: State of the Art and Research Challenges*. *Computer*, Vol. 40, 38-45.
- [7] Rodriguez-Mier, P., Mucientes, M., Lama, M. and Couto, M. I., (2010). Composition of Web Services through Genetic Programming. *Evolutionary Intelligence*, Vol. 3(3-4): 171–186.
- [8] Rosenberg, F. (2009). *QoS-Aware Composition of Adaptive Service-Oriented Applications*. PhD Dissertation. Technical University, Vienna, Austria.
- [9] Rotem-Gal-Oz, A. (2010). *What is SOA Anyway? Getting from Hype to Reality*. A White Paper, Retrieved on November 15th 2010 from <http://www.rgoarchitects.com/Files/SOADefined.pdf>
- [10] Zeng, L., Benatallah, B., Dumas, M., KAlagnanam, J. and Sheng, Q. Z. (2003). Quality driven web service composition. *Proceedings of the 12th International Conference on World Wide Web*, 411-421.
- [11] Cardoso, J., Sheth, A., Miler, J., Arnold, J. and Kochut, K. (2004). Quality of Service for workflows and web service processes. *Journal of Web Semantics*, Vol. 1(3), 281-308.
- [12] Lau, H., Chan, T. M. and Tsui, W. T. (2008). Item-Location Assignment Using Fuzzy Logic Guided Genetic Algorithms. *Institute of Electrical and Electronics Engineers (IEEE) Transactions on Evolutionary Computation*, Vol. 12(6), 765-780.