A NEW ONTOLOGY-BASED APPROACH FORWEB SERVICES SELECTION

Ilhem Feddaoui[1] Hela Limam[2] Hashim Jarrar[3] Jalel Akaichi[4]

[1]ECE Paris_Lyon
[2]Institut Supérieur d'Informatique, Université de Tunis El Manar, Tunisia and Laboratoire BestMod, Institut Supérieur de Gestion de Tunis, Tunis, Tunisia
[3]Department of Information Systems, Bisha University, Saudi Arabia
[4]Department of Computer Science, Bisha University, Saudi Arabia

ABSTRACT

In the contemporary online environment, the increasing prevalence of distributed and accessible web services presents a challenge, as users struggle to locate services that are suitable for their specific needs. In this context, it is essential to categorize web services that offer the same functionality to facilitate the selection of appropriate options. The imprecision of current web service selection methods can be attributed to the disregard for users' feedback and past selections. Furthermore, users encounter difficulties in selecting the most suitable web services when using conventional keyword-based search strategies. Our research introduces a novel hybrid approach that combines syntactic and semantic methods for selecting appropriate web services based on collaborative filtering, ontology-based querying, and QoS. Experiments demonstrate the effectiveness of our proposed selection technique in accurately recommending the required web services.

KEYWORDS

Ontology, Query, QoS, Collaborative Filtering, Web service.

1. INTRODUCTION

The growing number of web services poses a challenging task in the process of selecting appropriate options [16]. Users encounter difficulties in discovering suitable web services, which highlights the need for solutions that enable users to find the most satisfactory web services within their specific context. Consequently, the interest in semantic web service discovery has increased in recent years. One of the challenges of semantic web service discovery is that two semantically different concepts could have identical representations, which could result in low precision. Furthermore, given that multiple web services could offer similar functionality, ranking these web services is crucial. Quality of Service (QoS) is another criterion introduced as a determining factor [15].

Numerous studies have utilized Quality of Service (QoS) parameters to aid in the selection of web services [1] [2] [3]. However, these methods have certain limitations, such as an insufficient number of non-functional characteristics, and some methods [4] [5] do not consider precision in web service recommendations or the perspectives of both web service consumers and providers regarding qualities. Standard definitions of QoS parameters are essential for web service publications to be utilized effectively by both parties. However, user history is often neglected in the selection process. Therefore, semantic web service selection can significantly enhance the

David C. Wyld et al. (Eds): SPTM, AIS, SOENG, AMLA, NLPA, IPPR, CSIT – 2024
pp. 229-240, 2024. CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.141120

discovery process by integrating semantics into web services representation using ontologies [17] that provide a vocabulary and taxonomy of a specific domain, defining objects, classes, attributes, and relationships.

Many approaches [7] [8] [9] use ontology to solve selection issues, but they have limitations, particularly in the acquisition of functional and non-functional requirements for significant amounts of data sources. Additionally, traditional selection methods prioritize users' functional requirements, neglecting non-functional ones. As a result, more dynamic ontologies are required, capable of managing concepts beyond human capacity.

In this paper, we present a new approach for web service selection based on ontology, collaborative filtering, and QoS, consisting of three phases. In the first phase, an ontology is employed to analyze the query and identify the domain and QoS. In the second phase, web services are selected by domain, and their similarity to the query is calculated. Finally, in the third phase, web services are chosen based on their QoS. Section 2 discusses prior studies related to web service selection based on QoS and ontology. Section 3 describes the new approach in detail, and Section 4 presents experimental results and analysis. The paper concludes in Section 5, offering future research directions.

2. RELATED WORKS

In this section, a literature review of syntactic and semantic approaches for web service selection is provided, which is structured into two main parts. The first part gives an overview of Web service selection approaches based on Quality of Service (QoS). The second part presents recent research works in the field of ontologies for web service selection.

Various works have focused on QoS as a means for selecting desired web services. One such tool is the Social Spider Algorithm (SSA), which is proposed as an optimization strategy for choosing web services in a composition scenario [1]. Modeling QoS parameters of individual and composite web services, as well as the composition technique, are important elements in the problem of online services selection optimization. The mathematical modeling of predominant QoS factors has been performed using Markov model and Weibull analysis [2]. The behavioral aspects are studied using a scenario that is simulated using Colored Petri Nets (CPN).

In contrast to using a single factor that resulted in higher reputation with higher cost, the authors' investigation led to the choice of a service with a higher reputation but a lower usage cost. Collaborative filtering and QoS are used in conjunction to pick web services [3], where the selection process is divided into two stages: collaborative filtering based on memory and QoS for the first stage, and collaborative filtering based on model and QoS for the second stage. Another collaborative filtering and QoS-based technique for web service discovery is proposed for two phases [4]. In this case, the first phase is not utilized to filter out web services with QoS values, whereas the second phase is used to make recommendations for web services based on the QoS provided by the customer, which describes the non-functional requirements for the services.

To achieve QoS consensus and to resolve disagreements over QoS attributes when choosing online services, the QoS Consensus Moderation Approach (QCMA) has been developed [5]. The ranking of web services is done using a new mechanism [6], where the suggested approach computes the value of the web services' relevancy function while doing normalization. There has been less research work in the field of semantic web services that target to enhance or upgrade web service ontology qualitatively or that can facilitate the use of ontology and improve the selection process.

Multiple approaches have been proposed for web service selection, with ontology-based and QoS-based approaches being some of the popular methods.

In [7], an ontology-based approach for personalized RESTful web service discovery was introduced, which employs an ontology profile to personalize search results for different users. The approach uses a services relationship ontology to highlight similarity and complementarity relationships between services and a collaborative filtering method to filter returned services based on the opinions of similar users.

In [8], a job recommendation system was developed using a dynamic ontology that considers data from job searchers and data posted on websites. The system provides time-saving recommendations for suitable jobs for both employers and job seekers.

In [9], a novel method was presented for creating a QoS ontology and its QoS-based ranking algorithm for assessing web services. The QoS ontology can assist with not only documenting QoS information in detail but also simplifying the expression of QoS offers and requests at various degrees of expectation by different service participants.

In [10], the authors proposed an expansive and adaptable ontological standard called OWL-Q for semantic QoS-based web Service description. The authors examined every component of OWL-Q and concluded that more rules are necessary to provide property inferencing and constraint enforcement.

In [11], the authors presented a web service selection approach based on QoS attributes by extending the WS-Policy to represent QoS policies and applying ontological concepts to WS-policy to enable semantic matching. They also examined publishing QoS policies and suggested including QoS-based policies into the UDDI register expansion.

In [12], the authors addressed the problem of expressing the requester's preferences in a semantic context by extending the OWL-Q ontology to capture trade-off preferences expressed using QoSPref, a conditional lexicographic approach for QoS preference specification.

In summary, numerous approaches have been proposed for web service selection, each offering a distinct perspective. Some approaches are QoS-based, which use QoS as the sole criterion for selecting a web service, while others are ontology-based, which employ ontology as the means for selecting a web service. However, traditional syntactic search methods often produce either noisy or irrelevant results. If WSDL is annotated with semantic knowledge, UDDI can perform semantic search, leading to more complex web service discovery.

Researchers have taken various approaches to add semantic search capabilities to web service selection. Some researchers focus solely on describing the service semantically but fail to provide semantic searching capabilities in UDDI [3]. Other researchers add semantic searching to UDDI, but restrict the architecture to non-scalable, single point searching methodologies [4]. Some researchers also impose an imaginary world concept on service requesters and providers [5], which restricts web services to be more semantically flexible. However, common descriptive languages like WSDL lack the semantic richness required for automatic machine processing, and they require human intervention to interpret their meanings for discovery, composition, and invocation.

In this context, our approach is unique, as we combine semantic and syntactic selection to unify query comprehension and web service selection. Our approach utilizes ontology-based querying, collaborative filtering, and QoS-based selection to enable scalable searching with different service mechanisms at different endpoints. Consequently, we use different search algorithms at different endpoints. The steps involved in our approach are as follows: (1) Adding QoS settings to the domain of each service in the WSDL file. (2) Using ontology to analyze user requests and extract the requested service domain and QoS. (3) Implementing a collaborative filtering algorithm that measures the similarity of users on one side and the similarity of queries on the other side. (4) Conducting web service selection based on QoS.

3. Ontology Based Querying Approach for Web Service Recommendation

This work introduces a fresh method for optimizing search outcomes for diverse users seeking web services. Our method incorporates ontology to enhance the understanding of user queries, thereby improving comprehension of the domain pertaining to the requested service. Additionally, collaborative filtering and Quality of Service (QoS) metrics are employed in service selection. The proposed method is outlined in a general architecture (Figure 1), comprising two distinct phases. The ontology phase facilitates comprehension of both the domain of the requested service and the QoS criteria specified by the user, while the selection phase involves the discovery of web services based on query domain and assessment of query similarity to user history. In the subsequent layer, services are categorized according to the QoS specifications provided by the user.

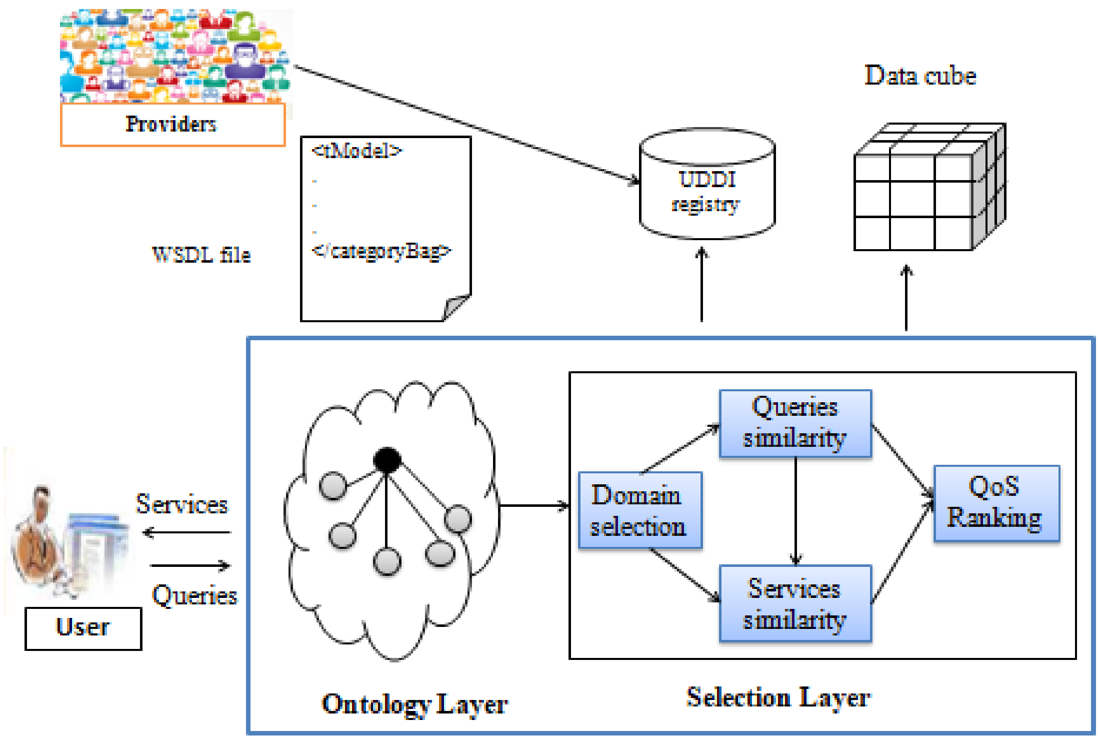

Figure1: General architecture of the approach.

3.1. Ontology-Based Querying

In our proposed approach for web service selection, ontology plays a pivotal role, characterized as "adomain conceptualization in terms of classes and properties that is formal, multilingual, consensual, and referenceable" [17]. As depicted in Figure 2, our query ontology is employed to grasp the domain of the requested query and the Quality of Service (QoS) provided by the user, facilitating the appropriate selection process. The initial step in ontology development involves describing its fundamental concepts, also referred to as classes, which constitute the backbone of our ontology. A concept delineates a set of distinct objects sharing common characteristics, with classes potentially serving as subclasses of one another, signifying that an individual of type A will also be categorized as a type B individual if class A is a subclass of B. Our ontology facilitates logic-based matching of requested and provided service functions based on inferred relationships, with the inputs and outputs of the ontology being:

- Input: User query
- Output: Domain and QoS request.

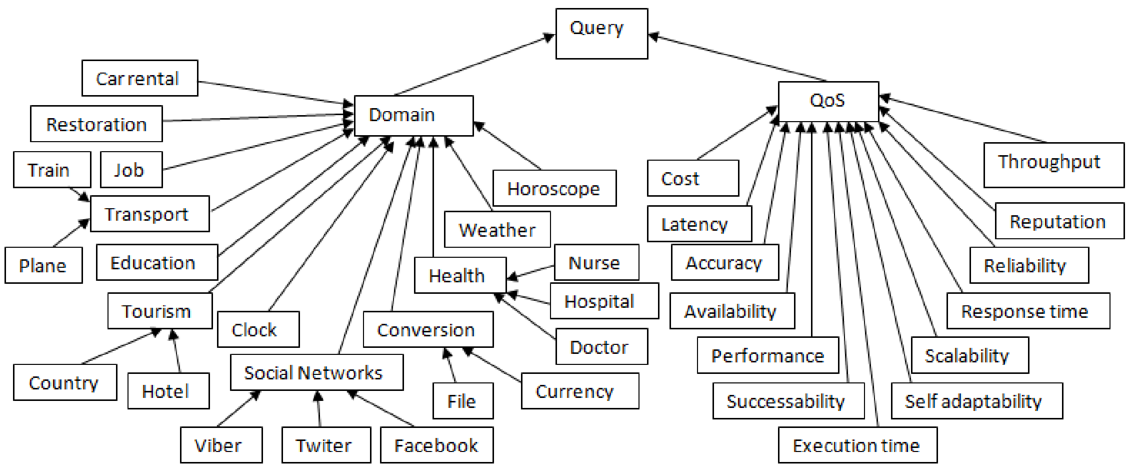

Figure2: Query ontology

3.2. Selection Phase

3.2.1. Domain Selection

In our proposed method, web service providers typically list their services on UDDI registries, detailing the domain and Quality of Service (QoS) parameters for each service. To enhance this process, we leverage the UDDI registry outlined in [14] and expand its capabilities to encompass domain and QoS parameter information. More specifically, we introduce a Quality-Information entity to delineate the QoS aspects of the provided service, which is integrated into the UDDI registry alongside the WSDL file containing QoS information and the web service domain.

Central to our approach is the Selection-Service, which plays a pivotal role. It receives the query domain and accesses the UDDI registry to identify web services sharing the same domain as the query (Q). Subsequently, the identified list of web services is forwarded to the Collaborative Filtering algorithm. In the subsequent section, we introduce an algorithm designed to accept the domain of the requested service as input, facilitating web service selection based on the domain of query (Q), and subsequently transmitting the selected list to the Collaborative Filtering algorithm:

Selection-Service algorithm:
Inputs: Domain.
Outputs: List L.

```
Begin
   List = {}
DomainQuery = getDomain(Q)
   For each Web_service in UDDI do
DomainService = getDomain(Web_service)
      If (DomainQuery == DomainService) then
         List = List ∪ {Web_service}
      End If
   End For
   If (List ≠ {}) then
      send List to Calculate-Service
   Else
      make a call to Web service providers
   End If
End
```

3.2.2. Collaborative filtering

The aim of this phase is to establish the connections between historical queries, web services, and users. This phase consists of three main steps: (1) Creating a matrix that links web services and queries, (2) Assessing the similarity between queries, and (3) Evaluating the similarity between users.

Step 1: Matrix Creation between Web Services and Queries

In this step, we intend to establish a matrix illustrating the relationship between web services and queries. Each entry in this matrix represents how often a web service is utilized by a specific query. To achieve this, for a given set of queries Q_j and a set of web services Wi, we generate a matrix M, where rows represent queries and columns represent web services. The value in each cell (I_{ij}) indicates the frequency with which web service W_i is invoked by query Q_j.

Step 2: Query Similarity Calculation

Here, we analyze all queries that a user has interacted with and calculate their similarity to a target query using the Pearson Correlation Coefficient. This formula computes the similarity between two queries based on the Quality of Service (QoS) values provided by common users. The result is a similarity matrix of queries.

Step 3: Service Similarity Calculation

Like Step 2, this step involves evaluating the similarity between pairs of web services. Using the Pearson Correlation Coefficient, we analyze the correlation between the QoS values provided by users for different services. This process enables us to construct a similarity matrix that provides insights into the similarities and connections among various web services.

Algorithm: Collaborative Filtering

1. **Input**: The algorithm takes as input a databaseT containing historical interactions between users, queries, and web services. This database provides the necessary information to compute similarities between queries and ultimately recommend web services.

2. **Output**: The output of the algorithm is a recommended web service based on the preferences and behaviors of users, inferred from similarities between queries.

Begin

3. For each pair of queries (i, j) inT:
   - Calculate similarity Sim (i, j) using Pearson Correlation Coefficient: For each pair of queries in the database, we compute their similarity using the Pearson Correlation Coefficient. This coefficient measures the linear correlation between two sets of data, in this case, the Quality of Service (QoS) values provided by users for each query. By comparing these QoS values, we determine how similar or dissimilar the queries are.
   - Store Sim (i, j) in matrixT1: After calculating the similarity between two queries, we store the resulting similarity value in matrixT_1. This matrix keeps track of all computed similarities between query pairs, providing a comprehensive overview of the relationships between queries.
4. Find the maximum similaritySimMax in matrixT_1:
   - Once we have computed the similarities between all pairs of queries, we identify the maximum similarity value,SimMax, within the matrixT_1. This maximum similarity value represents the highest degree of similarity observed between any pair of queries in the database.

End

By executing this algorithm, we leverage collaborative filtering techniques to recommend web services based on similarities between queries. The recommended web service corresponds to the pair of queries with the highest similarity, as indicated by the maximum similarity valueSimMax. This recommendation process helps personalize the user experience by suggesting web services that align closely with the preferences and behaviors of users, inferred from their interactions with similar queries.

3.2.3 . QoS Ranking

In this phase, we implement Quality of Service (QoS) to rank and prioritize web services. Among similar services, we select the one with the highest QoS rating. Our method revolves around user-specified QoS parameters, where users define their preferred QoS for service selection. Various parameters contribute to this process:

- **Accessibility:** This indicates the capability of a web service to fulfill a query.
- **Accuracy:** Represents the error rate produced by the service within a specified timeframe.
- **Availability:** Measures the time between a user's request for service access and the actual service access.
- **Cost:** Reflects the service charge levied by providers within the same group.
- **Execution time:** Indicates the time taken to execute a user request.
- **Latency:** Represents the time difference between the response time and the request time for invoking a web service.
- **Performance**: Assessed through throughput and latency, where lower latency and higher throughput signify better performance.
- **Reliability:** Measured by the ratio of successfully responded requests to total requests within a specific period.

- **Response time:** Indicates the time taken to respond to a user request.
- **Scalability:** Reflects the ability of a web service to handle varying request volumes.
- **Successability:** Measures the provider's ability to deliver successful results in response to user requests.
- **Throughput:** Represents the maximum number of transactions a web service can complete within a given time unit.
- **Reputation:** Derived from user feedback, calculated as the difference between positive and negative comments divided by the maximum number of completed transactions within a unit of time.
- **Self-adaptability**: Indicates the system's capability to dynamically adjust its behavior in response to internal and external events.

While some parameters like Accessibility, Cost, Execution time, and Self-adaptability are provided directly by service providers, others can be calculated by users using available tools.

The Calculate-Service algorithm, receiving a list of services from the Selection-Service, computes the QoS of each web service based on user-defined criteria and selects the most suitable one. For instance, if the user prioritizes efficiency, the algorithm selects the web service with the highest efficiency, calculated via QPerformance.

The selectionalgorithm:

1. Receive a list of web services (L) and the specified QoS parameter.
2. Iteratethrougheachweb service's QoS in the list.
3. Compare the QoS of each service.
4. Select the web service with the highest QoS.
5. Return the selectedweb service.

This algorithm ensuresthat the chosenweb service alignswith the user's defined QoS preferences.

4. EXPERIMENTS AND PERFORMANCE EVALUATION

In order to evaluate the efficacy of our proposal, we implemented the approach utilizing Netbeans 8.1, Tomcat 6.2, and Protégé, which is a software tool that enables the editing and visualization of ontologies (http://protege.stanford.edu). Through this implementation, we conducted a series of evaluations that measured the performance of various scenarios and enabled us to compare the resulting outcomes.

Our proposed application provides a user interface and supports the web service selection process outlined in the previous sections. Additionally, it facilitates web service retrieval in order to capture user interaction. The purpose of this section is to elucidate how this can be achieved when a query is submitted using our application. We demonstrate how the returned results are tailored to suit each individual user. According to our method, the principal functions of this system can be segmented into three parts: (1) The ontology phase, which is aimed at extracting the domain and QoS for web service selection, (2) The selection phase, which is comprised of two steps, namely, Hybrid collaborative filtering and (3) the QoS classification. When a new user submits a request to search for the best web service, our system provides a list of recommendations generated from the multidimensional database.

To test the effectiveness of our approach, we opted to utilize the "Tourism domain" as depicted in Figure 5. Users typically search for services such as hotel reservations, payment processing, and touristic place searches, among others. The evaluation process was applied to a collection of 62 services, of which 41 services belong to our retrieval domain of "Tourism". The remaining services are associated with other domains.

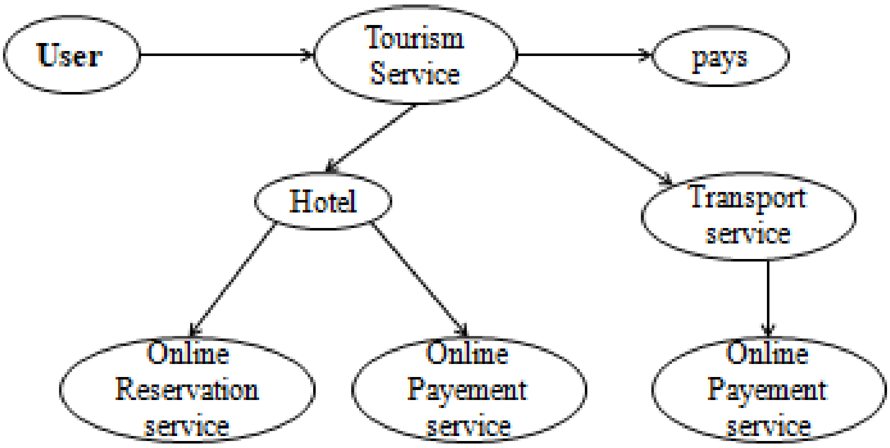

Figure5: Tourism domain description.

To assess the efficacy of the results generated by our approach, we compared the performance of our proposed methodology (Ontology-based query, Collaborative Filtering, and QoS - OCFQ) to that of the collaborative filtering method with QoS (CFQ) using the precision of classification method. The experimental simulation results, as shown in Figure 6, demonstrate that our approach can achieve a success rate of 88%, which is significantly superior to the traditional collaborative filtering method.

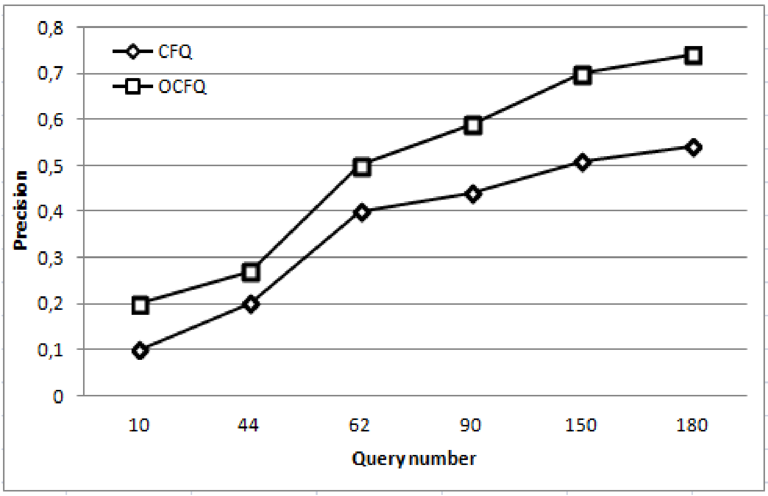

Figure 6: Comparison of the approach's performance.

Upon analyzing Figure 7 and Table 1, it becomes apparent that identical requests were submitted to our system by ten different users (U0, …, U10). It is noteworthy that, despite the requests being identical, the results generated were distinct depending on the user's history, prior experience, and feedback pertaining to the invoked services. Similar user histories were also considered. As the number of users and services increases, it is expected that the results will improve as there will be a greater number of opinions to refine the discovery process.

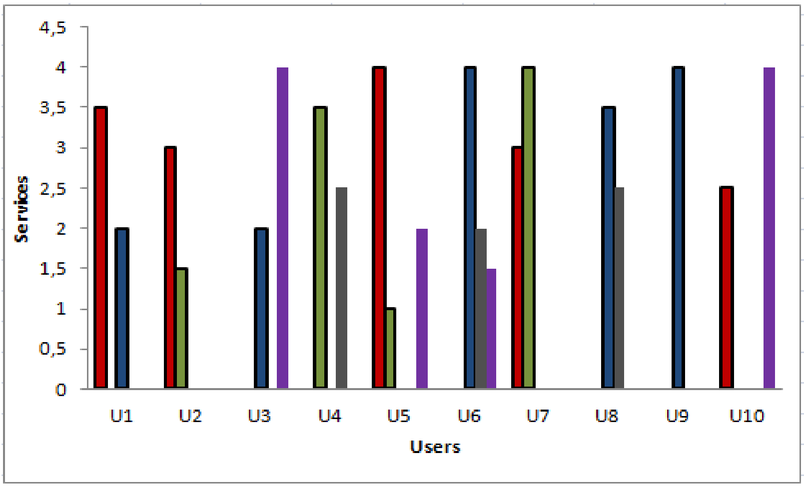

Figure 7:Selection of the different services according to different users.

	S1	S2	S3	S4	S5
U1	3,5		2		
U2	3	1,5			
U3			2		4
U4		2,5		3,5	
U5	4	1			2
U6			4	2	1,5
U7	3	4			
U8			3,5	2,5	
U9					
U10	2,5				4

Table 1:Selection web services according to different users' history.

5. CONCLUSION AND FUTURE WORKS

This paper presents an innovative approach for helping users select the most suitable web service based on their needs. The proposed method employs an ontology that connects users to the system and analyses their queries, while also including Quality of Service (QoS) data and the service domain in the WSDL file. By utilizing both semantic and syntactic selection, the method offers high precision and stability in operation, as confirmed by experimental data. Collaborative filtering is used in the selection step to calculate similarity and improve accuracy. We plan to improve the ontology in future research by adding more domains, terms related to each class, or by using different languages for the same class. Scalability concerns will also be addressed.

6. DATA AVAILABILITY STATEMENT

Data availability: Not applicable. This paper does not involve the collection or analysis of data. All information presented herein is based on literature review and theoretical analysis.

7. ETHICAL APPROVAL

This article does not contain any studies with human participants or animals performed by any of the authors.

8. DECLARATION OF INTEREST STATEMENT

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome. We confirm that the manuscript has been read and approved by all named authors and that there are no other people who satisfied the criteria for authorship but are not listed.

We further confirm that the order of authors listed in the manuscript has been approved by all of us. We confirm that we have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property. In so doing we confirm that we have followed the regulations of our institutions concerning intellectual property.

We further confirm that any aspect of the work covered in this manuscript that has involved either experimental animals or human patients has been conducted with the ethical approval of all relevant bodies and that such approvals are acknowledged within the manuscript.

We understand that the Corresponding Author is the sole contact for the Editorial process (including Editorial Manager and direct communications with the office).

He is responsible for communicating with the other authors about progress, submissions of revisions and final approval of proofs. We confirm that we have provided a current, correct email address which is accessible by the Corresponding Author.

9. REFERENCES

[1] Afaf Mousa, Jamal Bentahar, An Efficient QoS-aware Web Services Selection using Social Spider Algorithm, The 13th International Conference on Mobile Systems and Pervasive Computing, Vol 94, pp 176 – 182, 2016.

[2] Rajeev Pratap Singh, K.K.Pattanaik, An Approach to Composite QoS Parameter based Web Service Selection, The 4th International Conference on Ambient Systems, Networks and Technologies, vol 19, pp 470 – 477, 2013.

[3] Tao,Y. and Kwei, J.L. Service selection algorithms for Web services with end-to-end QoS constraints, ACM Transactions on the Web (TWEB), Volume 1 Issue 1, May 2007.

[4] Pradnya, K. and Rashmi, P. QoS based web service discovery using oo concepts, International Journal of Advanced Technology & Engineering Research (IJATER), Vol. 2, November 2012.

[5] Wei-Li Lin, Chi-Chun Lo, Kuo-Ming Chao, Muhammad Younas, Consumer-centric QoS-aware selection of web services, Journal of Computer and System Sciences, Vol 74, pp 211–231, 2008.

[6] M.Suchithra, M.Ramakrishnan, Efficient Discovery and Ranking of Web Services Using Non-Functional QoS Requirements for Smart Grid Applications, SMART GRID Technologies, Vol 21, pp 82 – 87, 2015.

[7] Sana Ben Abdallah Ben Lamine, Hajer BaazaouiZghal, Michael Mrissa, ChirineGhediraGuegan, An ontology-based approach for personalized RESTful Web service discovery, International Conference on Knowledge Based and Intelligent Information and Engineering Systems, Vol 112, pp 2127–2136 2017.

[8] Uma Pavan Kumar Kethavarapu, S.Saraswathi, Concept Based Dynamic Ontology Creation for Job Recommendation System, International Conference on Computational Modeling and Security, vol 85, pp 915 – 921, 2016.

[9] Vuong Xuan Tran, Hidekazu Tsuji, Ryosuke Masuda, A new QoS ontology and its QoS-based ranking algorithm for Web services, Simulation Modelling Practice and Theory Vol 17 pp 1378–1398, 2009.

[10] Kyriakos Kritikos and Dimitris Plexousakis, OWL-Q for Semantic QoS-based Web Service Description and Discovery.

[11] Sodki CHAARI, Youakim BADR, Frédérique BIENNIER, Enhancing Web Service Selection by QoS-Based Ontology and WS-Policy.
[12] Raluca Iordache, Florica Moldoveanu, QoS-Aware Web Service Semantic Selection Based on Preferences, 24th DAAAM International Symposium on Intelligent Manufacturing and Automation, vol 69, pp 1152 – 1161, 2014.
[13] Faîçal, F.and Jalel, A. Real Time Self Adaptable Web Services to the Context, Case Study and Performance Evaluation, International Journal of Web Applications (IJWA), Vol. 7, No. 1, pp 1-9, 2015.
[14] Shuping, R. A. Model for Web Services Discovery With QoS, CSIRO Mathematical and Information SciencesGPO Box 664, Canberra, ACT 2601, Australia, Volume 4 Issue 1, 2003.
[15] Summary of Quality Model for Web Services: https://www.oasisopen.org/committees/download.php/15444/Comparsion WSQMFE.doc.
[16] Vaneet S, Mukesh K. Web Service Discovery Research: A Study of Existing Approaches. International Journal on Recent Trends in Engineering and Technology, 2011, 5(1).
[17] G. Pierra, Context representation in domain ontologies and its use for semantic integration of data, J. Data Semantics 10 (2008) 174-211.

AUTHORS

Dr. Hashim is an assistant professor in the department of Information Systems, Bisha University - KSA. His research interests include databases, big data, ontologies, information systems, image encryption.