Jurnal
Teknologi

Full Paper

Semantic-based Discovery Framework for Web Services in Mobile Computing Environment

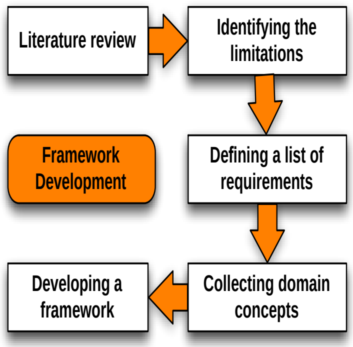
Nor Azizah Saadon*, Radziah Mohamad

Software Engineering Department, Faculty of Computing, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia

Article history
Received
2 February 2015
Received in revised form
8 October 2015
Accepted
12 October 2015

*Corresponding author
azizahsaadon@utm.my

Graphical abstract


Literature review
Identifying the limitations
Defining a list of requirements
Framework Development
Collecting domain concepts
Developing a framework

Abstract

As the number of Mobile Web Services (MWS) with different specifications has increased, the task of discovering the relevant ones becomes more challenging in a mobile environment. The challenges include mobile devices constraints and web service descriptions itself. The issues can be considered from the aspect of mobile device's performance, diversity of available MWS specifications and lack of enriched MWS descriptions. In order to address these issues, this paper presents an enhancement of semantic-based MWS discovery framework for discovering the most relevant MWS that takes into consideration of Non-Functional Properties (NFPs) in web service descriptions. In our work, the matchmaking algorithm with enhanced similarity measure is presented. Semantic MWS profiles are annotated semantically as a WSMO-Lite profile with a REST-based architecture. The experimental validation and statistical analysis demonstrates that the proposed framework can effectively discover relevant MWS according to the various user requirements. It can be concluded that using the proposed framework, there is a significant improvement in the effectiveness of semantic-based discovery for MWS in mobile computing environments.

Keywords: Mobile device, discovery, mobile computing, web service, semantic

Abstrak

Dengan pertambahan Servis Web Mudah-alih (MWS) yang mempunyai pelbagai spesifikasi hari ini, tugas mencari servis yang berkaitan menjadi semakin mencabar dalam persekitaran mudah alih. Cabaran ini termasuk kekangan alat mudah alih dan deskripsi servis web. Isu-isu ini boleh dipertimbangkan dari aspek prestasi alat mudah alih, kepelbagaian spesifikasi MWS yang ada dan kurangnya deskripsi MWS yang diperkaya. Dalam menangani isu-isu tersebut, artikel ini membentangkan penambahbaikan kepada kerangka penemuan MWS berasaskan semantik untuk mencari MWS yang paling sesuai mengambil kira Sifat-sifat Bukan Fungsi dalam deskripsi servis web. Dalam kajian ini, algoritma sepadan dengan penambahbaikan sukatan keserupaan dibentangkan. Profil semantik MWS dianotasi secara semantik menjadi WSMO-Lite dengan binaan berasaskan REST. Eksperimen pengesahan dan analisis statistik menunjukkan rangka kerja yang dicadangkan mampu mencari MWS yang berkaitan secara efektif berdasarkan keperluan pengguna yang pelbagai. Ianya dapat disimpulkan bahawa rangka kerja yang dicadangkan dapat meningkatkan keberkesanan penemuan MWS berasaskan semantik secara signifikan dalam persekitaran pengkomputeran mudah alih.

Katakunci: Alat mudah alih, penemuan, pengkomputeran mudah alih, servis web, semantik

1.0 INTRODUCTION

In recent years, the mobile computing field is becoming increasingly important due to the high demand of mobile technologies. Mobile computing involves mobile hardware, software and communication, which allow data communication through mobile devices and mobile software, without having to be connected to the physical link [1]. Mobile devices such as smartphones and tablets are now equipped with touch screen, embedded sensors, high resolution camera, better memory and processing capabilities as well as efficient operating system with significant improvements in both mobile hardware and software. High-end mobile devices allow users to access information in pervasive way, but are recognized as resource-constrained devices with limited capability.

According to the latest Mobile Factbook released by Portio Research [2], in the beginning of 2013, the worldwide mobile subscribers recorded nearly 6.5 billion subscribers and this number will increase 7.3 percent by the end of 2016. This report shows that the emergence of mobile devices and the improvement of the new paradigm of wireless technologies have directed to a significant demand for retrieving web services, which contributes towards Mobile Internet.

Current research trends mostly focus on the entire web service life cycle, including description, discovery, selection, composition and invocation [3]. Web service discovery in particular is the action of searching and matchmaking between available offered web services and service requester. Due to the increasing number of existing web services with similar functionalities and mobile device constraints, the users' ability in finding the most relevant web services that meet their requests has become even more difficult and hence, challenging. Furthermore, the existing web services discovery techniques are designed for static environments and lack the comprehensive mechanism for a mobile computing environment.

Mobile Web Services (MWS) discovery for mobile environment presents a significant challenge in the mobile domain. The proliferation of high-end mobile devices with different specifications, increasing number of similar web services, and lack of an enriched service description are the issues which have remained an open research questions [4]. Furthermore, mobile devices cannot support semantic web service discovery as it requires a heavyweight matchmaking process due to semantic reasoning. Thus, an efficient mechanism for MWS discovery is required to expand the ability of discovering web services over mobile devices with lightweight semantic web services.

Semantic web services such as Web Service Modeling Ontology (WSMO) [5] and Ontology Web Language for Service (OWL-S) [6] are top-down approach frameworks with recent improvement for web service technology. These approaches are hard to use with the current industrial developments where most of available web services use syntactic frameworks [7]. Effort in coming up with a bottom-up approach starts with simple extension of semantic description defined by WSDL-S [8]. Then, the effort grows by adding small enhancement on top of Web Service Description Language (WSDL) [9] which provides the ground for service modeling defined by semantic annotation for WSDL (SAWSDL) [10]. The recent effort known as WSMO-Lite [7] is a lightweight technology for annotating semantic web services. It uses SAWSDL annotation by introducing a service ontology with WSDL elements. The important of adding semantics to web services is that it improves automation of certain tasks and WSMO-Lite supports general tasks of discovery, negotiation, selection, composition and invocation of services.

WSMO-Lite is a lightweight technology that inherits its semantic design principles from WSMO. WSMO-LiteIt is an emerging technology produced by improving SAWSDL annotation with added ontologies and concrete semantic web service description. In WSMO-Lite conceptual model, SAWSDL is considered as the bridge between the non-semantic layer and semantic layer. There are three extension attributes where SAWSDL allows WSDL components to be annotated with semantic elements; namely modelReference, loweringSchemaMapping and liftingSchemaMapping.

WSMO-Lite seems to be the most competitive approach as this concept can use the available web services by adding semantic annotations. The semantic approach enhances the automation of discovery and selection for web services. Besides, it is a lightweight framework which is suitable for mobile computing environment and is chosen in this work because to its capability towards web service description.

The semantic-based approaches offer better discovery results by successfully retrieving the most relevant web services for service requester. MWS discovery is the process of locating service provider as a mobile provider that can match a service request specified by a mobile client [11]. As the number of available web services has been rising dramatically, semantic-based matchmaking is a promising approach to perform matching for discovering MWS. Hence, this paper proposes an enhancement of MWS discovery framework with semantic matchmaking approach in mobile computing environment. The framework aims at exploiting the limitation of the mobile devices.

The remainder of this paper is structured as follows. Section 2 presents the motivation behind this research. Section 3 discusses the background of related research. Section 4 provides the overview of MWS discovery mechanism. Section 5 describes the methodology of the proposed framework development. The proposed framework and its components' descriptions will be discussed in Section 6. Section 7 discusses the applicability study with some test cases. Section 8 presents the performance evaluation. Lastly, Section 9 concludes this paper and presents also the future work.

2.0 MOTIVATION

Mobile Web Services (MWS) discovery is a part of web service life cycle and it is an essential step in order to find web services dynamically performed by mobile devices. The purpose of MWS discovery is to locate a particular MWS based on service requester's requirement in order to discover the most appropriate MWS. However, the discovery of MWS is still a major challenge. The reasons for this include: (1) the diversity of mobile devices with different capabilities and constraints; (2) a variety of MWS available with different specifications; and (3) the lack of enriched service descriptions. Enriched information also refers to the enriched web service description, whereas the service is able to self-describe itself semantically with content-rich data. Ontologies provide semantically enriched profiles enable to find the most relevant mobile web services with respect to the user requirements and device profiles.

Currently, MWS discovery is based on functional requirements. However, as the number of similar functional web services increase, requirements requested solely by using functional properties will be insignificant. Furthermore, majority of the existing approaches rely on syntactic descriptions and disregard non-functional properties (NFPs). Thus, problem such as lack of enriched web service description in mobile computing affects the discovery process at the semantic level. In this sense, the discovery process often directs users to irrelevant result.

To solve this issue, enriched web service description through NFPs can help to improve the accuracy of the discovery process. NFPs can be exploited as the matching elements to express web services description semantically. NFPs such as Quality of Web Service (QoWS) and context information are becoming essential. When searching for a web service, despite of functional requirement, the service requester can specify QoWS and context elements in order to reduce the results returned to the requester's mobile device.

At the semantic level, the lightweight web services, WSMO-Lite is able to distinguish between the following four types of elements: (1) functional semantics; (2) non-functional semantics; (3) information model; and (4) behavioral semantics. Functional refers to functions provided to the requester and also describes service functionality. Non-functional defines details of the web service using an ontology which refers to the NFPs and policy. Information model defines the semantic input, output and fault messages. Behavioral semantics refer to the operation when the web service is invoked. WSMO-Lite seems to be the most competitive approach with semantic annotations. The semantic approach enhances the automation of discovery and selection for MWS discovery.

Web services discovery mechanisms are meant for the wired environment need to be enhanced for the mobile computing environment. Challenges unique to the mobile environment, such as rapidly changing connectivity, device capabilities, along with bandwidth and power constraints, all require special attention and still at its novice stage. Therefore, there is a need to develop a new MWS discovery framework that has the ability to match the service offered by the service provider and the service requested by the mobile requester with high accuracy.

3.0 RELATED WORKS

Many research have been made to overcome the drawback of MWS discovery. There are several initiatives trying to address MWS discovery through semantic-based matchmaking. However, most of them are still at an early stage and lack of comprehensive understanding as far as user context is concerned. The existing MWS discovery approaches in mobile computing environment for NFPs can be classified as context-aware discovery, QoWS-aware discovery and hybrid discovery, which includes both context and QoWS aspects.

3.1 Context-Aware Discovery

Context can be described as implicit information related to the service requester and service provider that can affect the usefulness of the returned results [12]. AIDAS [13] proposed a semantic-based matchmaking by exploiting user profile, device profile and service profile properties which relies on its own ontology. This middleware framework is a mobile user-centric semantic service discovery which is more focused on the service input and output.

MobiEureka [14] addressed the aspect of device capability with device-aware discovery mechanism. This work integrates device capability into MWS discovery by extending the current WSDL web service description with added device profile information. This framework only refers to the syntactic discovery and does not consider semantic web services.

Work in [15] presented semantic matching in mobile device by developing mTableu algorithm, a modified version of Pellet reasoner. It optimised the reasoning process to discover the most relevant web services in mobile devices. Kim and Lee [16] proposed a lightweight framework for mobile devices to host and migrate web services. It supports hosting and service migration on mobile devices. However, they used SOAP messages and the issue of overhead message still remains.

3.2 QoWS-Aware Discovery

Quality of Web Services (QoWS) refers to how well a service performs its behaviour to the customer. The research in [17] proposed MWS discovery and mediation framework, namely MWSMF (Mobile Web Service Mediation Framework) in mobile P2P network. MWSMF is addressed by the QoWS issues in terms of security and scalability. However, the research has a drawback, additional cost is required for compressing and decompressing the SOAP messages. The work in [18] proposed an agent-based framework with semantic-based web services discovery. However, the mobile device in this research only supports mobile client role.

Another approach in [19] proposed an extended service profile with user profile information. It allows QoWS attributes management by considering parameters such as response time, throughput, availability, interoperability, cost and accessibility. Research in [20] proposed evolutionary computing using Particle Swarm Optimization (PSO) technique for discovering web services. Their framework converts WSDL to OWL-S service profile and considers QoWS attributes. QoWS score are calculated by similarity matching and relevant services are ranked based on the QoWS score.

3.3 Hybrid Discovery

Hybrid concerns to both context and QoWS matching elements. However, taking both matching elements into consideration is a challenging issue because there is a trade-off between the query response time and peformance of relevant web services discovered. Work by [4] considers user preferences, device capabilties and QoWS for discovering relevant web services. Their approach is similar with MobiEureka approach by exploiting the Composite Capabilities / Preference Profile (CC/PP) specification for device capabilties and user preferences. In addition, they enriched it by proposing an ontology called Profile-Based Context-Aware Ontology (PBCO).

EASY [21] framework is an approach that introduces EASY-L (Language) which is extended from OWL-S, taking context and QoWS into account. Their approach matches purpose between service offer and request via a rate score aggregating the combination of functional and NFPs.

Recently, the research in [22] proposed a framework for MWS discovery. This work is similar to our approach, but they use the WSDL web service description with SOAP-based architecture for syntactic matching. In contrast, we focused on WSMO-Lite with REST-based [23] architecture for semantic matching.

To the best of our knowledge, all previous works for discovering web services in mobile computing environment are still at the early stage. In contrast to the work described above, the framework presented in this paper focuses on semantic-based matchmaking and ranking with NFPs provided from the service requester. Furthermore, the web service description is enriched with NFPs with its proposed context and QoWS ontologies.

4.0 MOBILE WEB SERVICE DISCOVERY MECHANISM

Basically, MWS discovery mechanism is the process of user searching for the relevant MWS and invoking it using a mobile device. Figure 1 illustrates the generic architecture of MWS discovery. In this figure, the mobile broker receives web service descriptions that will be published later by a mobile service provider. Then, based on the published web service descriptions in the mobile broker, the services are discovered by the mobile client. The main characteristics of MWS architecture vary from the traditional web services, especially in its mobility. Furthermore, in mobile environment, a mobile device can be equal participant as it can be a mobile client, a mobile broker, and even a mobile provider [16]. Meanwhile, as for the traditional web service architecture, the descriptions are published into a centralized repository, namely Universal Description, Discovery, and Integration (UDDI).

However, there are challenges in performing web service discovery in mobile computing with limited capability compared to the traditional web service architecture. The traditional web services do not meet the important criteria of mobile computing environment, and hence, it is not straightforward to host web service on mobile device. In addition, there are issues in mobile computing environment regarding the constraints of web services which require additional considerations, such as device capabilities, mobility, and security.

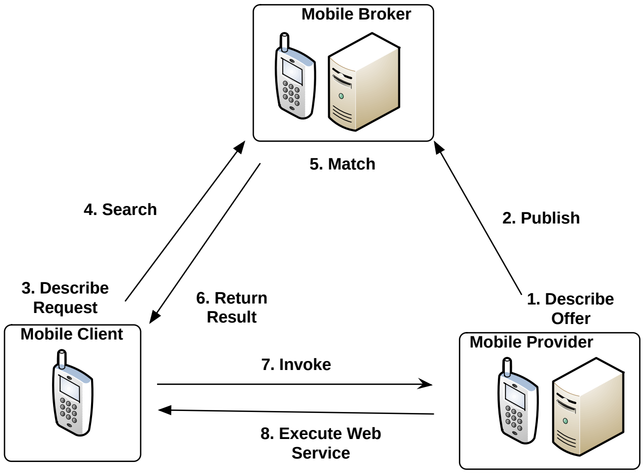


Figure 1 A generic MWS architecture

5.0 FRAMEWORK DEVELOPMENT METHOD

The method to develop the proposed framework consisted of five stages. In the first stage of collecting the domain concepts, several studies [14], [16]–[18], [22] were selected from the literature to collect the discovery requirements and their related concepts. Care had been given to represent almost all the basic web service discovery concepts in the framework, as described in [22].

In the second stage, mobile computing constraints for the web service discovery were considered. In addition, the semantic modelling for NFPs in mobile computing was also considered. Then, limitations for web service discovery in a mobile domain were identified, namely: device capabilities, reasoner capabilities, network connectivity, network access, mobility, context information, quality and security. Web services for mobile computing have to be designed within the limits of the available device constraints.

Therefore, in order to deploy mobile web services, these limitations should be taken into consideration. The following are the details for each limitation:

i. **Device capabilities**: With diverse types of mobile devices having different platforms and features, the service discovery mechanism should take into consideration the compatibility of mobile device capabilities with web services discovered.

ii. **Reasoner capabilities**: This refers to semantic reasoning in which the important part of the semantic matchmaking process is currently only suitable for high-end servers. With limited mobile device capabilities, semantic web service discovery could be a heavyweight matchmaking process in mobile devices. Thus, a highly optimized semantic reasoner is needed for the mobile computing environment.

iii. **Network connectivity**: A combination of intermittent connectivity and mobility of mobile users in the wireless network have affected the stability of the networks. Thus, web service discovery mechanisms must take into consideration that the discovered web services are active and usable.

iv. **Network access**: In wireless networks, mobile devices are always changing their points of network and IP addresses, resulting to services being provided to them to become inaccessible and hence invalid. Thus, an adaptive web service discovery mechanism which can provide alternative access to the same service is required. It should also be able to adapt to the environmental changes.

v. **Mobility**: Since mobile users are always on the move from one location to another, the changes of context may change rapidly and unpredictably. An environment where the service operates is impacted by the changes of location and context information as the mobile device moves. Thus, context-aware web service discovery is highly required.

vi. **Context information**: Mobile devices are associated with device profiles and context information. As mobile device is portable and operates with changes in environment, a web service discovery mechanism must incorporate the device's current profile and context information.

vii. **Quality**: Since mobile device capabilities are limited (such as small screen size display), only important aspects of user interface can be displayed. Thus, user friendly interface and quality-aware web service discoveries with user preferences are highly required to enable the retrieval of most relevant web services.

viii. **Security**: Mobile devices with wireless network communication medium for message exchanges may not be fully secured. Mobile users need to restrict certain information whenever their mobile devices become mobile hosts. Thus, authenticated users and secured access control are highly required in developing discovery mechanisms for mobile web services.

The third stage was to define a list of requirements that the framework should fulfil. The requirements are as follow:

- Service discovery mechanisms should ensure that the discovered services are available (active) and compatible with the device capabilities.
- Service discovery should incorporate context information of mobile devices for ranking and selecting MWS.
- The discovery of semantic web services is heavyweight matchmaking. Therefore, efficient discovery mechanism is required as the resource-intensive task is offloaded to resource-rich environment.
- The current service descriptions are not suitable for mobile devices because of the expressiveness of the service description. Therefore, lightweight service descriptions are required for mobile computing.

In the fourth stage, the important concepts of web service discovery were collected from traditional web service and mobile computing domains, and additional concepts of non-functional properties were introduced in the discovery process. Finally, the framework was developed in the construction stage, inspired from the basic architecture of MWS discovery for mobile computing, as shown in Figure 1.

6.0 MOBILE WEB SERVICE DISCOVERY FRAMEWORK

Based on the constraints and limitations of the existing MWS discovery mechanisms in smartphones, a semantic-based framework was proposed for MWS discovery for mobile computing environments. Figure 2 illustrates the important framework components to establish effective MWS discovery in mobile computing environments. The framework comprises of three main parts, namely mobile client, mobile provider, and mobile broker. The mobile client encompasses of components on the user request side or known as the service requester. The mobile provider part contains the components on service provider side and the mobile broker contains the components that run the process of discovery on the cloud. There are seven essential components in the framework: user query interface, query processor, service reader and parser, service classification, service matchmaking, service ranking, and *context manager*.

6.1 User Query Interface

User Query Interface is a mobile user interface, which enables mobile client as a user to submit MWS requests that express users' goal. The users' goal is described in a natural language (not in a formal service description language) as a keyword to represent service request. This is due to the limitation of input capabilities in mobile devices. This component listens for Hypertext Transfer Protocol (HTTP) or Extensible Messaging and Presence Protocol (XMPP) connection for REST-based web services and parses the service request further in *Query Processor* component to extract keywords and meaningful information.

6.2 Query Processor

Query Processor performs processing operations using Natural Language Processing (NLP) [24] techniques on the service request. Service request by mobile client must be processed to extract keywords into meaningful words from a user query written in plain text. These keywords are used to match the web service description in the repository and to perform keyword-based matching process. In our work, we used the WSMO-Lite web service description as a semantic web service and the design followed the RESTful philosophy [25].

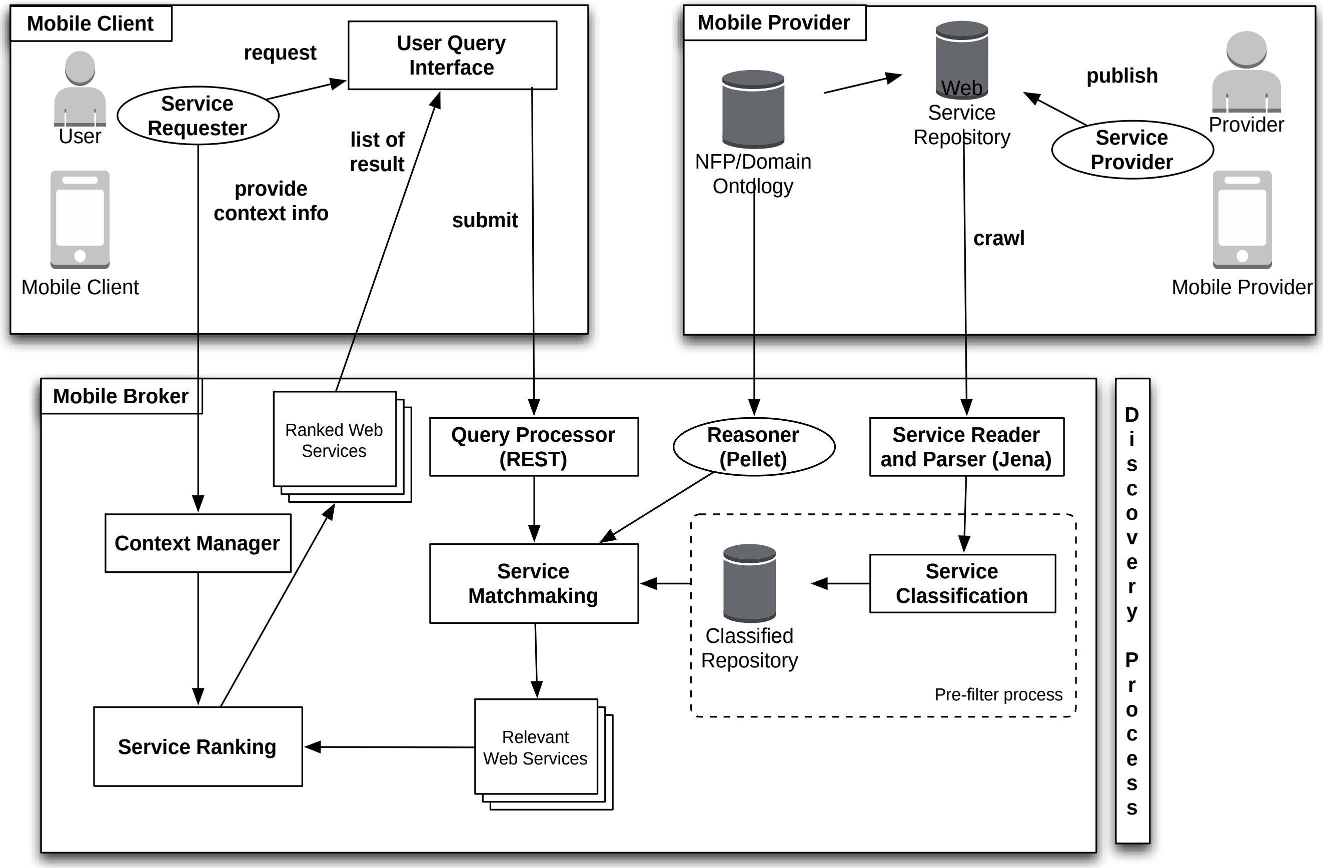


Figure 2 An overview of mobile web service discovery framework

Query Processor includes several steps of NLP techniques [24] for operations processing. Firstly, word segmentation known as tokenization is performed to split a string of written text of service request into its component words. Tokenization is a process of breaking concatenated text up into words, phrases or useful meaning of words called tokens. Then, the list of tokens becomes input for further processing, such as the parsing process. The parsing process is based on white spaces to produce a vector of tokens. Secondly, in order to find useful meaning of words, Part-of-Speech (POS) tagging is performed to tag tokens as nouns, verbs or adjectives. We used Stanford Part-of-Speech Tagger to process the tokens. Thirdly, the transformation of all tokens from the uppercase into lowercase characters is performed. Fourthly, word stemming is performed, as all tokens are transformed to their root words using Porter stemmer. Fifthly, Word Sense Disambiguation (WSD) establishes the tokens from previous process by extracting relevant sense. Many words represent multiple meanings. For example, bank can be used to represent financial institute or river shore. Therefore, WSD helps in finding the correct context. The variant of the SSI algorithm [26] used for finding the senses is described as follows:

$$selectedSense(word) = \arg \max_{s_j \in sense(word)} \sum_{sc_i \in I} sim(s_j, sc_i) \quad \text{(Eq. 1)}$$

6.3 Service Reader and Parser

There are several different web service descriptions that exist in semantic and non-semantic contexts. Each language has its own web service description reader and is able to parse a description by extracting the relevant elements from the description. In our approach, we used the WSMO-Lite web service description which is a semantic description that used the REST-based architecture.

Service Reader component extracts element from the semantic descriptions and ontologies. Based on an example of the WSMO-Lite web service, the service reader must be able to extract the elements, such as service names, input, output, and non-functional properties.

Service Parser component parses the web service description before extracting words from the description. Different parsers are available for different languages, such as the parser for WSMO [27] and WSML [28] is WSMO4J, while Sesame and Jena can be used as a parser for OWL-S [6], hREST [29], and WSMO-Lite [7].

6.4 Service Classification

The *Service Classification* component is used to categorize similar functionalities of web service to related categories based on the web service descriptions. Several steps in the classification process are similar to the process of *Query Processor* component, which include several steps of NLP techniques for processing the operations. However, this process refers to the web service descriptions in order to process the NLP technique. The ontology for categorization is presented during pre-filter process in order to filter the web services based on similar category.

6.5 Service Matchmaking

A basic step towards semantic discovery in mobile environments is established in the *Service Matchmaking* component, whereby the Degree-of-Match (DoM) between the concepts are calculated in a defined ontology. A multi-level matching approach is used to measure the DoM between two ontological concepts. Functional properties, such as input and output, are used for the matching process. In addition, non-functional properties are also used for the next level matching. DoM for functional properties and non-functional properties are aggregated with their weight for total DoM in order to list out the appropriate web services that match the service request.

The multi-level matchmaking algorithm in this work involves five steps. The first three steps involve functional properties that include input and output properties, NFPs of context properties, and NFPs of QoWS properties. Next, the Degree of Match (DoM) calculation is presented for all properties. Finally, web services with high value results are listed based on mobile users' requirements. Such approach exploits as much information as possible with the available functional and non-functional web service information.

Mobile web service (MWS) is defined as two-tuple that consists of functional and non-functional properties, where MWS = {F, NF}. Functional properties are the input and output properties of MWS. NFPs web service is defined as two-tuple where NFPs = {C, Q}, whereby C represents context properties and Q represents QoWS properties. Figure 3 shows the matchmaking algorithm deployed in our work with aggregation of all DoM.

As depicted in Figure 3, the input of the matchmaking algorithm is services requester marked as **R** and services offer of WSMO-Lite web service description, marked as **S**. The output is the list of the ordered services offered, which returns results to the service requester.

Firstly, the services performed a pre-filter process to categorize services with similar categories into the same domain. This process is to enhance the web service discovery by providing a user requester with refined options for the next matchmaking process before selecting a web service. Secondly, each service offer is matched with service requester in a pair matching of input, output, NFPs-QoWS and NFPs-context as `PairList` accordingly.

Next, for each couple in `PairList`, the function `AggregateWeight()` where the degree of match (DoM) is used to calculate weight and score for each pair. The function `AggregateWeight()` implements the rules of semantic-based matching. Finally, the algorithm returns an ordered list of matching services by using the function `Sort()`, which is restored in the list of `WeightedOrderedList`. The details of this matchmaking process can be referred to our previous work [30].

Algorithm 1 : Matchmaking Algorithm

```
Input:
R - Service Requester
O - Set of Services Offer
Output: List of ordered services offer

SameCategoryList = FilterByCategory(R,O)
WeightOrderedList = Null
for each service offer s ∊ O
  for each operation op ∊ s.getOperation()
     PairList = Null
     PairList[Inputs] = pair (R [inputs], O[inputs])
     PairList[Outputs] = pair (R [outputs], O[outputs])
     PairList (QoWS) = pair (R [QoWS], O [QoWS])
     PairList (Context) = pair(R [context], O [context])
     Return PairList
  end for
  for all Couples in PairList
    Weight = AggregateWeight (Couples)
    If Weight != Null
    (WeightedOrderedList =
    Sort(Weight, WeightedOrderedList)
  end for
Return WeightedOrderedList
end for
```

Figure 3 Matchmaking algorithm

As depicted in Figure 4, for each couple concept R (from Requester) and O (from Service Offer), semantic relationships of concept is used where the similarity degree of match is assigned from Exact (weight=6) to Fail (weight=0). The logic-based strategy of rating semantics considers seven grades of similarity measures to improve the matchmaking process which is inspired from work in [31].

The semantic matching relationship that are considered in this discovery approach are :

- **Exact-match** : The service requester is equivalent to the service offer , $R = O$.
- **RSubClassO** : The concept described by the requester is a direct subclass of the concept defined by the service offer. In this case the service offer is a more general concept than the requester.
- **OSubClassR** : The concept described by the service offer is a direct subclass of the concept defined by the requester. In this case the service requester is a more general concept than the service provider.
- **Plugin-match** : The service requester is a subset of the service offer, $R \subset O$. In other words, the concept described by the service requester is found within the subtree of the concepts described by service offer.
- **Subsume-match**: The service offer is a subset of the service requester, $R \supset O$. In other words, the concept described by the service offer is found within the subtree of the concepts described by service requester.
- **RSiblingO** : The service requester concept and service offer concept are from the same father concept which is still in the same tree.
- **Fail** : None of the elements, neither requester nor service offer is similar, $R \neq O$.

Algorithm 2 : Logic-based Algorithm

```
DegreeofMatch = 0
if Exact then weight +=6
else if RSubClassO then weight +=5
else if OSubClassR then weight +=4
else if Plugin then weight +=3
else if Subsume then weight +=2
else if RSiblingO then weight +=1
else if Fail then weight +=0
Return (weight)
```

Figure 4 Logic-based algorithm

6.6 Context Manager

The *Context Manager* component is to process the context information extracted from the mobile client to be used in the *Service Ranking* component for further process. Context information consists of device profile and environment context, which are used for ranking purpose. The details of the context properties model can be found in our previous work [30]. The compatibility of the discovered web services is established by the device profile information in the MWS discovery process. With this information, the discovered web services can be used properly with the current device status. Furthermore, the environment context information is also important as the mobile client may be on the move and the environment context may change frequently.

6.7 Service Ranking

The functionality of *Service Ranking* is to rank the discovered web services with the extra context information provided: device profile and environment context. The inputs of relevant web services and context information are used in the *Service Ranking* component to calculate the context information. Then, a list of ranked web services will be returned to the service requester for a selection process.

During the ranking process, the features from the mobile client device for device profile and environment context are collected. Table 1 shows the context profile with the ID number to represent each feature. For each discovered relevant MWS, whereby [MWS = $mws_1,mws_2,mws_3,\dots,mws_i$)] and context, [C = $C_1,C_2,C_3,\dots,C_j$)], a ranking calculation determines which mws_i is more precise based on context information.

Table 1 Context Profile

ID	Context feature
1	Operating system
2	Browser
3	Screen resolution
4	Memory
5	Audio
6	Video
7	Longitude
8	Latitude
9	Bandwidth data

With context features information (same as the ID from Table 1), the following context matrix (equation 2), whereby each row represents an MWS (mws_i) while each column represents each context features, is constructed.

$$S_O = \begin{bmatrix} mws_{1,1} & mws_{1,2} & mws_{1,3} & mws_{1,j} \\ mws_{2,1} & . & . & . \\ mws_{3,1} & . & . & . \\ mws_{4,1} & . & . & mws_{i,j} \end{bmatrix}$$

(Eq. 2)

The rank of each MWS (mws_i) is then represented by the following formula in equation 3. The function *f* computes relationship between the service request and the relevant web service descriptions, and f(x,y) gives a value in [0…1] to define how x value is related to y.

$$RankC = \sum_{i=1} w_i * f(mws_{ij}, C_j)$$

(Eq. 3)

where,
w_i = weight for each properties in *R* and *O*

7.0 APPLICABILITY STUDY

7.1 Experimental Scenario

The importance of NFPs in mobile computing's web service discovery is illustrated in our case study. In order to test the applicability of the proposed algorithm, we have defined a query representing the user's requirement. Using the medical ontology as depicted in Figure 5, the semantic relations between the parameters of MWS for functional properties are presented.

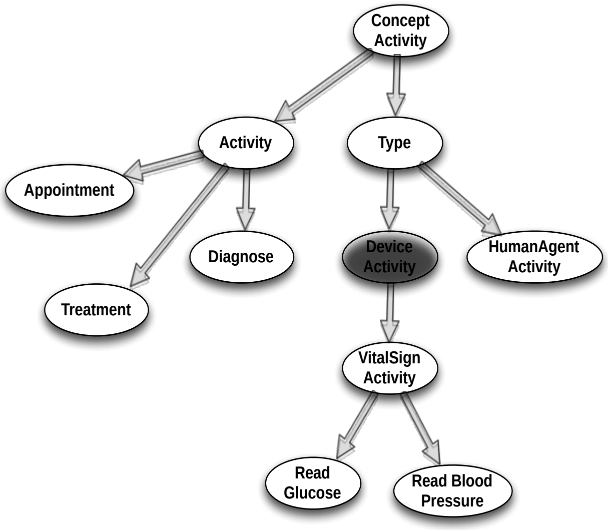


Figure 5 Part of medical ontology with test concepts

The requirement is defined as SET $_R$ = {input, output, cost, location} where input and output are the set of operation properties while cost and location are the set of NFPs. Cost represents NFPs-QoWS and location represents NFPs-context. Therefore, user's request is defined as:

SET $_R$ = {Date, DeviceActivity, Free, Johor}.

In order to fulfill user's requirement, these set of properties are evaluated with the proposed matching algorithm. As shown in Table 2, there are several web services, namely MWS 1 to MWS 10 that offer similar functionalities as requested by user requester. There are four elements used in this study which are input, output, NFPs-QoWS and NFPs-context. For example, MWS 1 offers date as input properties and device activity as an output properties. While for NFPs-QoWS, MWS 1 offers free cost and NFPs-context offers location in Johor. All these properties are matched with user's request as SET $_R$ using semantic matchmatching algorithm and the Degree of Match (DoM) between the service requester and service provider is calculated.

Table 3 shows the results of matching properties whereby each of the matched properties is given a score from Algorithm 2 as shown in Figure 4. The total score obtained from the function `AggregateWeight()` is the aggregation from all DoM (input, output, NFPs-context and NFPs-QoWS).

For example in MWS 1, service offer (O) for input properties is Time.owl#Date and it is considered as 'Exact-match' with user request (R) Time.owl#Date. The score result is 6 based on the rating semantic (Exact-match) used in this research.

As shown in Table 3, total score with and without NFPs are presented in Table 3. It is discovered that MWS 1 and MWS 3 give equal value and the highest degree of matching without considering NFPs where the total score is 12 for both. If the user invokes the service MWS 1, then the user needs to pay for the cost of the service unexpectedly even if the user requested for a free service. The invoked web service will be charged because there is no mechanism to predefine the NFPs compatibility in the first place.

For this applicability study, it can be seen that the input properties for all web services are same in order to evaluate the differences, particularly in output properties and NFPs. For matchmaking purposes of this case study, the output properties between service requester and service offered should refer to the test concept of medical ontology as in Figure 5.

Table 2 The list of mobile web services offered

			NFPs - QoWS	NFPs - Context
Web service	**Input**	**Output**	**Cost**	**Location**
MWS 1	Time.owl#Date	Activity.owl#DeviceActivity	Cost.owl#charge	Location.owl#Johor
MWS 2	Time.owl#Date	Activity.owl#TypeActivity	Cost.owl#free	Location.owl#Johor
MWS 3	Time.owl#Date	Activity.owl#DeviceActivity	Cost.owl#free	Location.owl#Johor
MWS 4	Time.owl#Date	Activity.owl#HumanAgent	Cost.owl#free	Location.owl#Johor
MWS 5	Time.owl#Date	Activity.owl#VitalSign	Cost.owl#free	Location.owl#Johor
MWS 6	Time.owl#Date	Activity.owl#ReadGlucose	Cost.owl#free	Location.owl#Johor
MWS 7	Time.owl#Date	Activity.owl#Appointment	Cost.owl#free	Location.owl#Johor
MWS 8	Time.owl#Date	Activity.owl#TypeActivity	Cost.owl#charge	Location.owl#Melaka
MWS 9	Time.owl#Date	Activity.owl#Treatment	Cost.owl#charge	Location.owl#Johor
MWS 10	Time.owl#Date	Activity.owl#VitalSign	Cost.owl#charge	Location.owl#Johor

Table 3 Degree of Match (DoM) results between available mobile web services and user request (Req1)

			NFPs – QoWS	**NFPs – Context**	**Total Score**	
User Request	**Input**	**Output**	**Cost**	**Location**		
Req1	Time.owl#Date	Activity.owl #DeviceActivity	Cost.owl#free	Location.owl #Johor		
Available Web service	**Input**	**Output**	**Cost**	**Location**	**Without NFPs**	**With NFPs**
MWS 1	6	6	0	5	12	17
MWS 2	6	5	5	5	11	21
MWS 3	6	6	5	5	12	22
MWS 4	6	1	5	5	7	17
MWS 5	6	4	5	5	10	20
MWS 6	6	2	5	5	8	18
MWS 7	6	0	5	5	6	16
MWS 8	6	5	0	0	11	11
MWS 9	6	0	0	5	6	11
MWS 10	6	4	0	5	10	15

7.2 Result Analysis

For standard basic semantic relations, the similarity degrees of exact-match and direct subclass have been assigned with the same weight for these measures. However, we use extended semantic relations inspired from the work in [31] whereas these similarity degrees can be differentiated with the related concept between service requester and service offer and have been assigned with different weight as in Figure 4.

For example, with the standard basic semantic relations, **MWS 2** and **MWS 5** have the same weight for output properties if the request output concept is '**DeviceActivity**'. However, by referring to the concept taxonomy in Figure 5 and by using extended semantic relations, **MWS 2** and **MWS 5** can be differentiated as Exact-match and RSubClassO respectively. It is concluded that the differentiation between exact-match and direct subclass resulted in web services more appropriate to the request.

In addition, using sibling relationship (RSiblingO), '**HumanAgent**' and '**Appointment**' concepts can be differentiated if the request output concept is '**DeviceActivity**'. As shown in Figure 5, '**HumanAgent**' and '**DeviceActivity**' are from same father with sibling concept. However, '**Appointment**' and '**DeviceActivity**' are from the different branch tree without any close relationship between them. Therefore, '**HumanAgent**' obtained a score of 1 while '**Appointment**' obtained no score in this matching process even they are from the same level. It is concluded that the differentiation with sibling relationship resulted in more appropriate web services as been requested.

Furthermore, as shown in Table 3, it is discovered that **MWS 1** and **MWS 3** give the highest degree of matching without considering NFPs where the total score is 12 for both. Both MWS (1 & 3) only differ on NFP-QoWS for cost properties. If the user invokes the service MWS 1 instead of MWS 3, then the user needs to pay for the cost of the service as they are no different between them. The invoked web service will be charged accordingly because there is no mechanism to predefine the NFPs as the matching elements in the matchmaking process.

With the consideration of NFPs in context and QoWS, this study discovers that MWS 3 is the most suitable web service for user requester with the highest score, 42. MWS 3 can satisfy user's request for functional properties (input and output), considering NFPs-context for location properties and NFPs-QoWS for cost properties.

From the discussion above, the importance of NFPs, especially in mobile computing with limitation of device capabilities is highlighted. The degree of matching with ranked results of web services give mobile users accurate information in order to select the most appropriate web service based on users' requirement and device capabilities.

8.0 PERFORMANCE EVALUATION

This section describes the evaluation strategies and evaluation analysis of the semantic discovery framework in mobile computing environment. The evaluation results are analyzed to show the effectiveness of semantic-based discovery for mobile web services framework. The experiment is designed with respect to the plan recommended by Wohlin *et al*. [32] for experimental evaluation in software engineering. Firstly, the experiment design is performed to analyze the proposed framework. Secondly, the hypothesis of this research is defined. Thirdly, the evaluation criteria is defined in order to validate the hypothesis. Fourthly, the statistical analysis is performed and the results are discussed.

8.1 Experiment Design

Significant statistical test is conducted to the mobile web service discovery framework to validate the framework. Considering that there is no comprehensive framework for semantic discovery which is comparable to the framework, therefore the efficiency of the provided framework has been compared with the case that Non-Functional Properties (NFPs) does not act as discovery elements for semantic discovery. For this purpose, two scenarios have been defined as follows:

- Scenario 1: The semantic discovery of mobile web services without NFPs.
- Scenario 2: The semantic discovery of mobile web services with NFPs.

8.2 Hypothesis

The hypothesis is defined with respect to the contribution of this research in producing a semantic discovery framework for mobile computing environment. The testing of the hypothesis in this research is based on the effectiveness metric. The null hypothesis is formed as follows:

- $H_{0\ \text{discovery quality}}$: there is no significant difference between the discovery quality of scenario with NFPs and the scenario without NFPs.

If the null hypothesis can be rejected with high confidence, the corresponding alternative hypothesis is supported. The alternative hypothesis which is corresponding to the defined null hypothesis is listed below:

- $H_{1\ \text{discovery quality}}$: the discovery quality of scenario with NFPs has significant difference from the discovery quality of the scenario without NFPs.

8.3 Evaluation Criteria

Performance measure is used to evaluate the effectiveness of the proposed semantic discovery for mobile web service discovery. The evaluation criteria in this research are as below:

- Discovery quality: Ratio of successful retrieved web services to total number of web services. In other words, the percent of successful web services retrieved is considered as the discovery quality. Discovery quality is calculated as follows:

  $$Q = N_{\text{retrieved}} / N_{\text{total}}$$

  With $N_{\text{retrieved}}$ is the number of successful retrieved web services, N_{total} is the total number of web services.

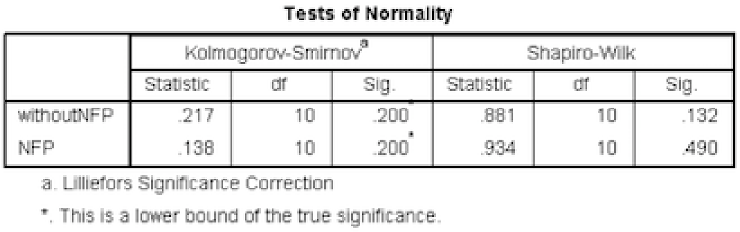
8.4 Statistical Analysis

In order to determine the difference of the proposed framework is significant or not, statistical analysis of a paired t-test experiment is used in this research. The said test is a parametric test for a paired comparison design to compare two samples mean. Before conducting the test, it is important to ensure that the distribution of data meet the assumptions of parametric testing. If the data is normally distributed, the parametric test is used, otherwise the non-parametric test is used.

This test used Shapiro-Wilk test to check for data normality. Figure 6 shows the significant p-value of Shapiro-Wilk test for both frameworks, with and without NFPs are 0.490 and 0.132 respectively. Since the p-value for both frameworks are greater than 0.05, then the data is normally distributed and parametric test is used.

In Figure 7, statistically significant result of the paired t-test indicates that the null hypothesis should be rejected as presented in the Sig. (2-tailed) column. The p-value for this result is 0.000 which is lower than 0.05. The statistical test on the discovery quality results shows the rejection of the corresponding null hypothesis and consequently acceptance of the alternative hypothesis. Thus, there is a significant difference for discovery quality of framework with NFPs compared to without it.

Based on the data in Figure 7, by analyzing the results and the means value, it can be concluded that the major reason behind the rejection of the null hypothesis is the consideration of semantic-based discovery with NFPs elements in the web services description.



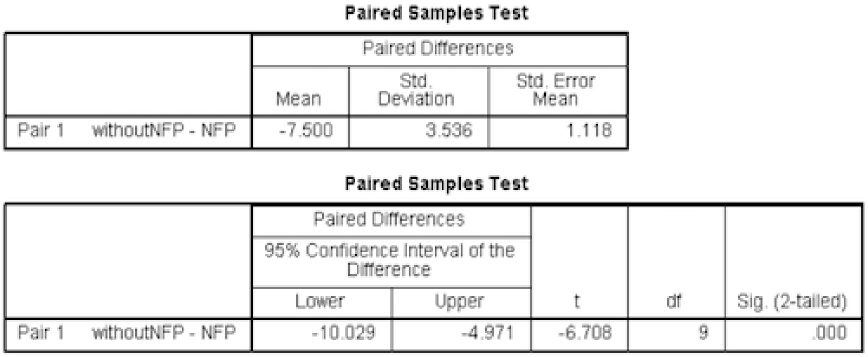
Tests of Normality

	Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
withoutNFP	.217	10	.200*	.881	10	.132
NFP	.138	10	.200*	.934	10	.490

a. Lilliefors Significance Correction

*. This is a lower bound of the true significance.

Figure 6 Result of normality test to check data distribution



Paired Samples Test

		Paired Differences		
		Mean	Std. Deviation	Std. Error Mean
Pair 1	withoutNFP - NFP	-7.500	3.536	1.118

Paired Samples Test

		Paired Differences				
		95% Confidence Interval of the Difference				
		Lower	Upper	t	df	Sig. (2-tailed)
Pair 1	withoutNFP - NFP	-10.029	-4.971	-6.708	9	.000

Figure 7 Result of paired t-test

9.0 CONCLUSION AND FUTURE WORK

Mobile Web Services (MWS) discovery is an important part in the web service life cycle and the process of MWS discovery poses new challenge due to the limitations of mobile devices. Mobile devices have become the most significant interface for mobile computing environment because of their mobility and accessibility. The increasing number of mobile clients accessing information anywhere and anytime from mobile devices has increased the growing demand for MWS discovery in mobile computing.

This paper has presented an enhancement of semantic-based MWS discovery framework for discovering the most relevant MWS that takes into consideration Non-Functional Properties (NFPs) in web service descriptions. In our work, the matchmaking algorithm with enhanced similarity measure is presented. Semantic MWS profiles are annotated semantically as WSMO-Lite profile with a REST-based architecture.

The applicability study for this framework is presented to demonstrate the importance of NFPs' web service discovery in mobile computing. In addition, the experimental validation and statistical analysis demonstrate that the proposed framework can effectively discover relevant MWS according to the various user requirements. It is concluded that using the proposed framework, there is a significant improvement in the effectiveness of semantic-based discovery for MWS in mobile computing environment.

Our ongoing research extends the prototype framework to achieve better accuracy and precision in web service discovering with the enhancement of QoWS properties and context properties as NFPs preferences. Accordingly, a prototype has been developed on an android-based platform to evaluate the applicability of the framework.

Nowadays, cloud computing has been widely recognized as an effective reuse paradigm and the computing infrastructure for the next generation. As future work, to improve the framework by taking advantage of the cloud is our next plan and strategy. Thus, the process of discovering heavyweight matchmaking such as reasoning, classification, matchmaking and ranking will be applied onto the cloud.

Acknowledgement

We would like to thank Universiti Teknologi Malaysia for sponsoring the research through the RUG grant with vote number 05H83 and providing the facilities and support for the research. In addition, we would like to extend our gratitude to the lab members in the EReTSEL Lab of Faculty of Computing, Universiti Teknologi Malaysia for their invaluable ideas and support throughout this study.

References

[1] Deepak, G., & Pradeep, B. 2012. Challenging Issues and Limitations of Mobile Computing. *Int. J. Computer Technology & Application.* 3(1): 177-181.

[2] Portio Research Mobile Factbook 2013. 2013. Retrieved from http://www.portioresearch.com/media/3986/Portio Research Mobile Factbook 2013.pdf (Accessed on 1st August 2015).

[3] Yu, Q., Liu, X., Bouguettaya, A., & Medjahed, B. 2006. Deploying and Managing Web Services: Issues, Solutions, and Directions. *The VLDB Journal.* 17: 537-572.

[4] Niazi, R., & Mahmoud, Q. H. 2009. An Ontology-based Framework for Discovering Mobile Services. In *2009 Seventh Annual Communication Networks and Services Research Conference IEEE.* 178-184.

[5] Roman, D., Keller, U., Lausen, H., de Bruijn, J., Lara, R., Stollberg, M., Fensel, D. 2005. Web Service Modeling Ontology. *Applied Ontology.* 1(1): 77-106.

[6] Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Sycara, K. (n.d.). OWL-S: Semantic Markup for Web Services. Retrieved from http://www.w3.org/Submission/OWL-S/ (Accessed on 1st August 2015).

[7] Kopecký, J., & Vitvar, T. 2008. WSMO-Lite: Lowering the Semantic Web Services Barrier with Modular and Light-Weight Annotations. *2008 IEEE International Conference on Semantic Computing*. 238-244.

[8] Akkiraju, R., Farrell, J., J.Miller, Nagarajan, M., Schmidt, M., Sheth, A., & Verma, K. 2005. Web Service Semantics - WSDL-S. Retrieved from http://lsdis.cs.uga.edu/projects/METEOR-S/WSDL (Accessed on 1st August 2015).

[9] Christensen, E., Curbera, F., Meredith, G., Weerawarana, S. 2001. WSDL 1.1. Retrieved from http://www.w3.org/TR/wsdl (Accessed on 1 August 2015).

[10] Kopecký, J., Vitvar, T., Bournez, C., & Farrell, J. 2007. SAWSDL: Semantic Annotations for WSDL and XML Schema. *IEEE Internet Computing*. 60-67.

[11] Srirama, S., Jarke, M., & Prinz, W. 2010. Mobile Web Service Discovery in Peer to Peer Networks. arXiv Preprint arXiv:1007.3631.

[12] Doulkeridis, C. 2008. CASD: Management of a Context-aware Service Directory. *Pervasive and Mobile Computing.* 4(5): 737-754.

[13] Toninelli, A., Corradi, A., & Montanari, R. 2008. Semantic-Based Discovery to Support Mobile Context-aware Service Access. *Computer Communications.* 31(5): 935-949.

[14] Al-Masri, E., & Mahmoud, Q. H. 2009. MobiEureka: An Approach for Enhancing the Discovery of Mobile Web Services. *Personal and Ubiquitous Computing*. 14(7): 609-620.

[15] Steller, L. A., & Honours, B. 2010. Light-Weight and Adaptive Reasoning for Mobile Web Services by Caulfield School of Information Technology Monash University. Interface.

[16] Kim, Y. S., & Lee, K. H. 2009. A Lightweight Framework for Mobile Web Services. *Computer Science-Research and Development.* 24(4): 199-209.

[17] Srirama, S., Vainikko, E., Šor, V., & Jarke, M. 2010. Scalable Mobile Web Services Mediation Framework. In *2010 Fifth International Conference on Internet and Web Applications and Services IEEE.* 315-320.

[18] Adacal, M., & Bener, A. B. 2006. Mobile Web Services: A New Agent-based Framework. *IEEE Internet Computing.* 10(3): 58-65.

[19] Al-Masri, E., & Mahmoud, Q. H. 2009. MobiEureka: An Approach for Enhancing the Discovery of Mobile Web Services. *Personal and Ubiquitous Computing.* 14(7): 609-620.

[20] Bhuvaneswari, A., & Karpagam, G. R. 2010. Reengineering Semantic Web Service Composition in a Mobile Environment. In *2010 International Conference on Recent*

Trends in Information, Telecommunication and Computing IEEE. 227-230.

[21] Mokhtar, S., Preuveneers, D., Georgantas, N., Issarny, V., & Berbers, Y. 2007. EASY: Efficient semAntic Service Discovery in Pervasive Computing Environments with Qos and Context Support. *Journal of Systems and Software*. 81: 785-808.

[22] Elgazzar, K., Hassanein, H. S., & Martin, P. 2013. DaaS: Cloud-based Mobile Web Service Discovery. *Pervasive and Mobile Computing.*

[23] Wilde, E., & Pautasso, C. 2011. REST: From Research to Practice. (E. Wilde & C. Pautasso, Eds.). New York, NY: Springer New York.

[24] Perea-Ortega, José M., García-Cumbreras, M. A., & Ureña-López, L. A. 2013. Applying NLP Techniques for Query Reformulation to Information Retrieval with Geographical References. Emerging Trends in Knowledge Discovery and Data Mining, Springer Berlin Heidelberg. 57-69.

[25] Pautasso, C., & Leymann, F. 2008. RESTful Web Services vs. "Big" Web Services: Making the Right Architectural Decision Categories and Subject Descriptors. *Technology*. 805-814.

[26] Navigli, R., & Velardi, P. 2005. Structural Semantic Interconnections: A Knowledge-based Approach to Word Sense Disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 27(7): 1075-86.

[27] Roman, D., & König-ries, B. 2006. D2v1. 3. Web Service Modeling Ontology (WSMO). Elements.

[28] Bruijn, J. D., Lausen, H., Polleres, A., & Fensel, D. 2006. The Web Service Modeling Language WSML: An Overview. The Semantic Web.

[29] Kopecký, J., Gomadam, K., & Vitvar, T. 2008. hRESTS: An HTML Microformat for Describing RESTful Web Services. *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*. 619-625.

[30] Saadon, N. A., & Mohamad, R. 2014. WSMO-lite Based Web Service Discovery Algorithm for Mobile Environment. *International Journal of Advances in Soft Computing and Its Applications*. 5(SPECIALISSUE.3): 75-90.

[31] Samper Zapater, J. J., Llidó Escrivá, D. M., Soriano García, F. R., & Martínez Durá, J. J. 2015. Semantic Web Service Discovery System for Road Traffic Information Services. Expert Systems with Applications. 42(8): 3833-3842.

[32] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. 2012. Experimentation in Software Engineering*. Advances in Computers* (Vol. 44). Berlin, Heidelberg: Springer Berlin Heidelberg.