MASTER THESIS

A SYSTEMATIC MAPPING STUDY OF WEB SERVICES DISCOVERY MECHANISMS

EXAMINER(s): Professor, Ph.D Kari Smolander
M.Sc, Tommi Kähkönen

Supervisors: Professor, Ph.D Kari Smolander

Lappeenranta 20.11.2014
Mr Faisal Mohamed Awid
Itsehallintie 4A4
02600 Espoo
awidfaisal@yahoo.com
ABSTRACT

Author: Faisal Mohamed Awid
Name: A Systematic mapping study of web services discovery mechanisms
Year: 2014 Place: Lappeenranta

Master’s Thesis

Lappeenranta University of Technology

Keywords: SOA, SOC, WS, Web Service Discovery, Systematic Mapping, Service Discovery categorization.

Context: Web services have been gaining popularity due to the success of service oriented architecture and cloud computing. Web services offer tremendous opportunity for service developers to publish their services and applications over the boundaries of the organization or company. However, to fully exploit these opportunities it is necessary to find efficient discovery mechanism thus, Web services discovering mechanism has attracted a considerable attention in Semantic Web research, however, there have been no literature surveys that systematically map the present research result thus overall impact of these research efforts and level of maturity of their results are still unclear. This thesis aims at providing an overview of the current state of research into Web services discovering mechanism using systematic mapping. The work is based on the papers published 2004 to 2013, and attempts to elaborate various aspects of the analyzed literature including classifying them in terms of the architecture, frameworks and methods used for web services discovery mechanism.

Objective: The objective if this work is to summarize the current knowledge that is available as regards to Web service discovery mechanisms as well as to systematically identify and analyze the current published research works in order to identify different approaches presented.

Method: A systematic mapping study has been employed to assess the various Web Services discovery approaches presented in the literature. Systematic mapping studies are useful for categorizing and summarizing the level of maturity research area.

Results: The result indicates that there are numerous approaches that are consistently being researched and published in this field. In terms of where these researches are published, conferences are major contributing publishing arena as 48% of the selected papers were conference published papers illustrating the level of maturity of the research topic. Additionally selected 52 papers are categorized into two broad segments namely functional and non-functional based approaches taking into consideration architectural aspects and information retrieval approaches, semantic matching, syntactic matching, behavior based matching as well as QOS and other constraints.
Acknowledgements

I take this opportunity to thank my principal supervisor Professor Kari Smonlander for his unreserved support and continuous guidance as well as affording me to provide an extension study period when the project lasted longer than anticipated in the early stage of the work. This work has thought me a great deal of my personality in terms of resilience and patience that are needed to go through difficult times as times things seemed to be far and unattainable dream. Without Mr Kari’s valuable advices and encouragement, I could not have finished this work.

I would like also to thank my second supervisor Mr Tommi Kähkönen for his valuable suggestions and constructive feed backs during the research phase, also many thanks go to Department of Information Technology for their support and allowing me to continue my studies after my entitled study period expired.

Finally special thanks go to my family, friend and acquaintances for their help and support to get me to the point of finishing this work and accomplishing another milestone of my life.

When many work together for a goal,
Great things may be accomplished.
It is said a lion cub was killed
By a single colony of ants.
—Saskya Pandita
# Table of Contents

1 Introduction .................................................................................................................................................. 9  
1.1 Purpose ............................................................................................................................................... 9  
1.2 Motivation ........................................................................................................................................... 10  
1.3 Research Objectives .......................................................................................................................... 10  
1.4 Thesis Timeline ................................................................................................................................... 11  
1.5 Thesis organization .............................................................................................................................. 11  
2 Service-oriented Computing and Business Process .............................................................................. 12  
2.1 Service Oriented Computing and Service oriented Architecture ..................................................... 15  
2.2 Web Services (WS) ............................................................................................................................. 17  
2.2.1 Advantages Web Service .............................................................................................................. 19  
2.2.2 Disadvantages Web Services ....................................................................................................... 20  
2.3 Core Web Service Protocols ............................................................................................................ 21  
2.3.1 Extensible Mark-up Language (XML) ......................................................................................... 21  
2.3.2 Simple Object Access (SOAP) ..................................................................................................... 24  
2.3.3 Web Service Description Languages (WSDL) ........................................................................... 26  
2.4 Web Service Discovery Methods ...................................................................................................... 32  
2.4.1 Universal Description, Discovery, and Integration (UDDI) ......................................................... 32  
2.4.2 Search Engines ............................................................................................................................. 34  
2.4.3 Publication Site ............................................................................................................................ 35  
2.5 Criteria of Web Services Discovery .................................................................................................. 40  
3 Method .................................................................................................................................................... 42  
3.1 Study Design ....................................................................................................................................... 43  
3.2 Data Collection Instrument ............................................................................................................... 44  
3.4 Content Analysis ................................................................................................................................. 45  
3.4.1 General Overview ......................................................................................................................... 47  
3.4.2 Classification of the Studies ......................................................................................................... 49  
4 Results and Analysis .............................................................................................................................. 52  
4.1 Motivations and Contributions ......................................................................................................... 54  
4.2 Discussions ......................................................................................................................................... 57  
5 Conclusions ........................................................................................................................................... 62
List of Figures
Figure 2-1. Business Process Management Service Pattern [51] ................................................................. 13
Figure 2-2: Web Service Architecture in Enterprise[52] .................................................................................. 14
Figure 2-4: Web Services Architecture [9] ..................................................................................................... 16
Figure 2-5: Web Service Collaboration [16] .................................................................................................... 18
Figure 2-6: Web Service Stack [15] .............................................................................................................. 21
Figure 2-7: An Example of XML Schema ....................................................................................................... 23
Figure 2-8: SOAP Message Structure [14] .................................................................................................... 25
Figure 2-9: Web Services Description Languages[58] .................................................................................. 27
Figure 2-10: Service Ontology Model[49] ...................................................................................................... 30
Figure 2-11: UDDI Registry Elements[19] ..................................................................................................... 33
Figure 2-12: Web Services Provided By RemoteMethods[54] ...................................................................... 36
Figure 2-13: List of Available Web Services Provided by Xmethods[55] .......................................................... 37
Figure 2-14: List of Available Web Services Provided By ProgrammableWeb[56] ......................................... 38
Figure 3-1: Taken From [2] .......................................................................................................................... 43
Figure 3-2: Study Selection Process ................................................................................................................ 46
Figure 3-3: Publication Venue ........................................................................................................................ 47
Figure 3-4: Publication Distribution ................................................................................................................ 49
Figure 3-5: Categorization of Web Services Discovery Architectures[50] ....................................................... 50
Figure 3-6: Functional and Non-Functional Web Service Discovery Categorization[50] ............................... 50
Figure 4-1: Paper Types ................................................................................................................................. 54
Figure 4-2: Selected Papers Contributions ..................................................................................................... 57
List of Tables

Table 2-1: SOAP VS REST [17] ................................................................................................................... 26
Table 2-3: Comparison on UBRs and Search Engines [28] ........................................................................... 34
Table 2-4: Overview of Web Services Directories [28] .................................................................................. 38
Table 2-5: Comparison on Web Services Discovery Methods [36] ................................................................. 39
Table 3-1: Inclusion and Exclusion Criteria .................................................................................................... 45
Table 3-2: Selected Electronic Databases ....................................................................................................... 46
Table 3-3: Number of Papers Published in Data Sources ................................................................................ 48
Table 3-4: Categorizing Selected Papers ........................................................................................................ 51
Table 4-1: Paper Types [47] ......................................................................................................................... 53
Table 4-2: Paper Contributions ....................................................................................................................... 55
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WS</td>
<td>Web Services</td>
</tr>
<tr>
<td>SOA</td>
<td>Service Oriented Architecture</td>
</tr>
<tr>
<td>WSDM</td>
<td>Web Services Discovery Mechanism</td>
</tr>
<tr>
<td>ACM</td>
<td>Association for Computing Machinery</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>SOAP</td>
<td>Simple Object Access Protocol</td>
</tr>
<tr>
<td>WSDL</td>
<td>Web Service Description Language</td>
</tr>
<tr>
<td>UDDI</td>
<td>Universal Description, Discovery and Integration</td>
</tr>
<tr>
<td>SOC</td>
<td>Service Oriented Computing</td>
</tr>
<tr>
<td>WSMO</td>
<td>Web Service Modelling Ontology</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Mark-up Language</td>
</tr>
<tr>
<td>CORBA</td>
<td>Common Object Request Broker Architecture</td>
</tr>
<tr>
<td>JRMI</td>
<td>Java Remote Method Invocation</td>
</tr>
<tr>
<td>IT</td>
<td>Information Technology</td>
</tr>
<tr>
<td>DCOM</td>
<td>Distributed Component Object Model</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hyper Text Transfer Protocol</td>
</tr>
<tr>
<td>REST</td>
<td>Representational State Transfer</td>
</tr>
<tr>
<td>JSON</td>
<td>Java Script Object Notation</td>
</tr>
<tr>
<td>BPM</td>
<td>Business Process Management</td>
</tr>
<tr>
<td>W3C</td>
<td>World Wide Web Consortium</td>
</tr>
<tr>
<td>SGML</td>
<td>Standard Generalised Mark-up Language</td>
</tr>
<tr>
<td>DTD</td>
<td>Document Type Definition</td>
</tr>
<tr>
<td>OWL</td>
<td>Web Ontology Language</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Language</td>
</tr>
<tr>
<td>EbXML</td>
<td>Electronic Business Extensible Mark-up Language</td>
</tr>
<tr>
<td>CSV</td>
<td>Comma-separated Values</td>
</tr>
<tr>
<td>P2P</td>
<td>Peer to Peer</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>CoS</td>
<td>Cost of Service</td>
</tr>
<tr>
<td>B2B</td>
<td>Business to Business</td>
</tr>
<tr>
<td>RSS</td>
<td>Really Simple Syndication</td>
</tr>
<tr>
<td>HTML</td>
<td>Hyper text Mark-up Language</td>
</tr>
<tr>
<td>OWL-S</td>
<td>Web Ontology Language for Service</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------</td>
</tr>
<tr>
<td>VSM</td>
<td>Vector Space Model</td>
</tr>
<tr>
<td>LSA</td>
<td>Latent Semantic Analysis</td>
</tr>
</tbody>
</table>
1 Introduction

Service-oriented architecture (SOA) has become the standard paradigm for software component integration as it provides a framework within which enterprises expose functionalities in the form of loosely coupled services that can be integrated and consolidated in response to the user’s demand. These services are called Web Services as defined by the World Wide Web Consortium “A software system designed to support interoperable machine-to-machine interaction over network [1]. With the growing prevalence of Service Oriented Architecture (SOA), the number of Web Services has increased rapidly, as SOA emphasizes interoperability, decoupled components, and reuse. However due to large number of Web Services available it becomes time consuming process for end user to find appropriate and desired services that meet their demand. W3C has also defined Web service discovery as “the act of locating a machine-processable description of a Web service that may have been previously unknown and that meets certain functional criteria. The process involves matching a set of criteria with a set of Web service description with a goal of obtaining desired and appropriate Web service” [1]. Since Web service plays important role in locating Web service of interest in large pool of Web services, this entails the need for effective and efficient Web service discovery approaches and over the past several years, languages and frameworks have been proposed as an attempt to reduce Web service discoverability process. Due to this number of research work on this area has substantially increased. In order to be able to assess the present state and the maturity of research area it is important to study and examine the available research work, although there have been some effort to classify and categorize them, to my knowledge there has been no survey that follows methods of systematic mapping. Systematic mapping study enables you evaluate the existing research area and find any gaps that have been overlooked by the researchers.

1.1 Purpose

As research area matures and number of related papers increases, it is important to systematically identify, analyze and classify the state-of-the-art and provide an overview of the trends in this specific area. This thesis aims at providing systematic mapping study on the Web services discovery mechanism. The systematic mapping study is a means of categorizing and summarizing the existing information about a research area and it aims to minimize error and bias and in turn increases the quality of the work. It is also defined as a method to build a classification scheme and structure a software engineering field of interest. The analysis of results focuses on frequencies of publications for categories within the scheme. Thereby, the coverage of the research field can be determined. Different facets of the scheme can also be combined to answer more specific research questions [2].

The work is based on the papers published 2005 to 2013, and attempts to elaborate various aspects of the analyzed literature including classifying them in terms of the architecture, frameworks and methods used for web services discovery mechanism. This paper reports on the result of systematic mapping of approaches on Web services discovery mechanism that have been evaluated by collecting and surveying research literature. This is accomplished by employing
rigorous methods to identify, appraise and synthesize research by establishing effective research questions. The research will address the following questions:

- What are type of publications, publication year and the names of publication venue (Workshop, conference and journals)
- What type of research is being done on Web Services discovery mechanism and what are the main contributions of these researches?
- What are different approaches, models and frame works that are being explored in the present research regarding with web services discovery mechanism (WSDM)?
- What kind of future research directions have been reported in the present literature on Web service discovery mechanism (WSDM)?

Similarly, in addition to the above research questions content analyze will be used to extract relevant data from the collected data, exclusion and inclusion criteria will be utilized as means to classify the compiled papers. Subsequently the expected result will be reported accordingly, this in turn will benefit the researchers to understand current state of research maturity of the area in question.

1.2 Motivation

This work is motivated both the need to know the frameworks and tools designed for Web service discovery, and the opportunity to exploit the research work done in the past on this vast growing area of web service discovery as service discovery is the most important part of Web service lifecycle.

Due to the growing significance in Service Oriented Architecture and popularity of Web Services attracted the attention of researchers and practitioners; however there is lack of systematic mapping studies in the area. This work will be focusing on the current state and future direction of the topic in the research. This work will be beneficial for researchers aiming at exploring the current state of the art.

Finally the topic of the thesis is contemporary and new concept that has grasped the attention of most practitioners, developers and research community. It is opportunity for me to get acquainted with such contemporary issue and for my career prospect; it would provide an excellent knowledge base that would enhance my possibilities to get hired in the job market.

1.3 Research Objectives

During past decade, researchers and academic communities have been proposing different systems, methods and approaches for this fast evolving research area of Web service discovery systems. The objective of this work is to analyse collected articles and papers in order to evaluate the topic from various dimensions: general trends of the
published work such as sources, types and time of publications as well as researched topics and methods used. All in all following steps will be carried out to attain research objectives:

- Investigate research literature systematically on the topic of Web Services discovery mechanism using search strings.
- A systematic mapping study will be conducting using the data collection instrument, when the data is collected a content analysis will be applied by creating coding schemes. The result of analyzed information will be presented.
- Based on obtained collected work papers, level of maturity of the research area will be assessed.
- Provision of the current state of the research looking into different Web Services discovery mechanism presented. Identifying works being done about WSDM as well as pointing out any gaps that might be present.

To gather articles an iterative process will be used, collecting research studies systematically from popular journals including ACM digital library, IEEE Explorer, Science Direct, Springer link, Elsevier and Emerald, Google scholar and Citeseer.

1.4 Thesis Timeline

Since I have already collected most of the relevant papers published about the topic, the classification, mapping stages, and content analysis followed by the discussion of the result will be next phase. So the time line of the work will be:

- Data collection April 2014 to August 2014.
- Coding and Analysis August 2014 to October 2014
- Reporting October 2014 to November 2014.

1.5 Thesis organization

The remainder of this thesis is structured as follows: Initially background information about the SOA and its technologies is presented to give the reader general overview of the area. This section provides a link between the methodology used in this work and the work that has been done in the area of Web services discovery. Chapter 2 explains the details in regards to the systematic review methodology used in the study. This chapter outlines the structure behind the planning stage of the review, execution stage of the review and details the data collection instrument used to collect data. Chapter 3 presents the results, as well as the analysis behind the study. It also describes the overview of the studies, evaluation methods used, quality of the evaluation, and validation details. In addition, it presents the findings of Web Services discovery mechanism, as well as the purpose, and then offers a future outlook. Chapter 4 provides an in-depth discussion on the results, while Chapter 5 outlines the limitations of the study. Lastly Chapter 6 concludes and summary overview of the paper.
2 Service-oriented Computing and Business Process

The usage of Internet has been drifted from a sole repository for publishing contents such as texts and images to a popular tool that revolutionized the way we run our daily business. As the Internet increasingly becoming faster and more reliable business community and enterprises wanted to take advantage of emerging technologies and use it as a platform to reach their business partners and customers.

Service oriented computing is a contemporary IT field that covers issues like service principles and service technologies to business oriented aspects including service consulting and delivery or providing service solutions and management [3]. The above mentioned definition conforms to the earlier explanation given by Gartner in 1996 “Style of multitier computing that helps organizations share logic and data among multiple applications and usage of modes [4]. Since 1996 various publications have been trying to modify the definition of the term, some definition has been focusing on technical aspect of the SOA while others emphasized business aspects of the paradigm. This is due to the fact that SOC’s researchers came from different fields including information science, business science, and computer science. However, semantic web service is our utmost interest, and we will be using terms like service and web service interchangeably.

Business process has always played a significant role in information systems. Business process can be defined as a collection of related activities that its ultimate goal is produce a service or product that attracts more and more service and product users, it is also defined as “systematic approach to making an organization's workflow more effective, more efficient and more capable of adapting to an ever-changing environment”. Due to this Business Process Management (BPM) strive for optimizing the business process to link to client's needs in order to generate as much profit as possible.

Bearing in mind the aforementioned facts, the BPM recently accepted as pattern diagram that aligns business process to IT driven redesign initiatives. This pattern has been adopted to reduce ineffective process and increase productivity as well as enhancing service delivery. The figure2-1 below illustrates interaction and activities between people and systems where IT plays major role in bridging Business process and enhancing enterprise overall revenue target as well as consumer satisfaction.
Figure 2-1. Business Process Management Service Pattern [51].

With the competitive globalized economy, companies strive for business integration by globalizing their business process. This made service oriented computing as an emerging discipline that enables to link the gap between business process and IT services. SOC architectural approach allows companies to access external services published by other service providers, on the other hand it similarly enables to offer their services to other organizations and companies. SOC utilizes service oriented architectural approach (SOA) to integrate business application through collaborating web services using standardized interfaces and protocols. Hence WS represent basic fundamental blocks responsible for creating and executing business process residing over geographically dispersed area. Figures 2-2 and 2-3, clearly depict how enterprises use WS as an emerging technology of integrating applications across companies’ boundaries to achieve goals of business to business (B2B).
Figure 2-2: Web Service Architecture in Enterprise[52].
2.1 Service Oriented Computing and Service oriented Architecture

As mention earlier service oriented computing is a computing paradigm that refers to a set of concept, design principles and methods that utilizes service as fundamental elements for developing application. Services are cross platform software components that allow rapid, low-cost composition that locates over dispersed geographical area. Service enable organisation to expose their core products and services over the Internet using a set of standard XML based languages and protocols [5][6].

To attain its goals service oriented computing (SOC) utilizes service oriented architecture paradigm (SOA) in which software applications are constructed based on independent component services with standard interfaces. Gartner [7][8] has predicted that “by the year 2010, 80% of all applications software will be driven by products based on Service oriented architecture (SOA)”.

In SOA paradigm, services are deployed in order to organize and implement IT architectures and eventually realize Business/IT alignment. The success of SOA paradigm can be linked to its ability to overcome the main challenges facing IT executives including problems arising from integrating heterogeneity legacy systems i.e. the systems from multiple vendors and different partners and suppliers, as well as its capability to cope with the global e-commerce pace of change as it provides mechanism to adopt ever changing market trends and needs to satisfy the demand of the end-consumers.
The advantages that can be gained by using of SOA can be summarized as follows:

- Due to its components reusability and flexibility, the SOA provides cost effective, reliable services to customers.
- It enables more business operations via Internet and thus increases business opportunities.
- It eliminates frustrations with IT through flexible solutions and shorter lead time to deployment.
- It justifies IT investments more clearly through the closer association of IT to business services.
- It provides to business executives with a clear understanding of what IT does and its value.

In a brief service oriented architecture model can be summarized the interaction between service provider, service broker and service consumer. This interaction involves publishing service, finding it, binding and execution it. This architectural paradigm can be used to integrate multitude applications on various technologies and platforms. This in turn enables different enterprises to mix and match different services to perform business transactions. The following figure 4 clearly illustrates the interaction between different roles and entities in the SOA architectural model.

![Figure 2-3: Web Services Architecture [9]](image-url)
A **Service provider** establishes number of services, service in here represents software component that has service interface, service provider places it all its services in a service registry to make it accessible and available for end users. The registry is normally kept in service broker, where services can be published and requested.

A **service consumer** requests services, if the service is available and known in advance, the service is directly allocated, however, if the service is not already known, service broker initiates invocation process that involves searching the service, finding it, and executing it. As can be seen from above diagram, discoverability of a service plays vital role in this architecture, thus, the description of a web service as well as its presentation in a service registry is primary interest. However due to the proliferation of web service, it is becoming increasingly difficult to locate the required Web Services.

In the following sections, detailed background information about Web Services and related technologies are presented including defining web services, as well as its building block protocols such as WSDL, XML and SOAP.

### 2.2 Web Services (WS)

Enterprise application integration has become a vital issue and researchers of e-commerce development have presented a various solutions to exploit the business opportunities provided by the Internet. Some of the early technology proposed to deal with enterprise application integration included CORBA, DCOM, JAVARMI [10]. Although to some extent these technologies provided some solutions to prevailing problems at the time, however, there was a common problem stemming from lack of interoperability since software components and applications written in specific programming language require a common communication platform that enables to integrate different applications residing on a various dispersed locations[12]. Lack of effective interoperability has considerably reduced flexibility and in turn created complexity by increasing the cost of enterprise application integration. To come over the interoperability obstacles Web Services emerged as best solution for enterprise application integration.

According to W3C web service is defined as “Software system identified by a Uniform resource identifier, whose public interfaces are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the web service in a manner prescribed by its definition; using XML based messages conveyed Internet protocols” [1].

The Web services are loosely coupled software components that are published using XML based Web Service Description Language [13], Web Services are able to communicate each other via Simple Object Access [14], can be located and invoked by using registry system called Universal Description Discovery and Integration [11]. All these technologies are Internet open standard protocols designed for discovering web services. The Web service is a remotely accessible application that resides on multiple geographically dispersed servers that intended to provide to attain specific functionalities.

There are two types of Web services, namely simple or informational which basically provides contents to end users through a process based on "request/response" between the client and the server, occasionally the server side might
pass the request to the backend database to fetch or update information. This type of WS can in turn be divided into three categories based on nature of the functionality it provides. These three subcategories are pure content service, simple trading service and information syndication services. The other type of WS is composite service that involves aggregating several services from various service providers to accomplish a complex business process, the latter WS can be further divided into groups based on how they were grouped, these two subcategories are complex services that is composed by programmatic WS, whiles the other category is complex service compose by interactive WS.

In general there are certain features that all WS have in common including:

- **WS are self-contained meaning** on the client side additional software is not required merely a browser supporting XML and HTTP with script language would be enough to access WS, on the other hand on the server side Web Server and engine is needed to process user or client request.
- **WS are self-describing implying** that either side "Client" or "Server" needs to know apart from the message content containing request and response. In other words, WS is loosely coupled application integration.
- **Web services can be published, located, and invoked** across the Web.
- **Web services are language independent and interoperable,** irrespective of the platform, programming language or operating system used service user and service providers can interact with each other using communication channels.
- **Web services are compassable indicating** Web Services are aggregated to accomplish complex tasks.
- **Web Service are dynamic implying** using different technology including WSDL and UDDI, WS can be discovered and invoked dynamically.

The following figure demonstrates the collaboration between service provider and service consumer based on SOA model previously discussed.

![Figure 2-4: Web Service Collaboration](image)

[16]
The ultimate goal of Web Services architecture is to facilitate a platform that service provider can publish their services and similarly enable service end users to locate that service. The major components of building block of Web Service architecture is presented by [9].

The above figure elaborates three major components that represent core of WS as follows:

**Service Provider:** the service developers and providers include enterprise; companies, organizations as well as individual software developers e.g. open source software. This group publishes their service information through the discovery system or registry to attract service users.

**Service Requester:** Similarly this group represents companies, organizations, or individuals who require the functionalities of the Web services. This group consumes the service by retrieving service provider’s information from a global registry. Once the required services are located, the requesters can directly invoke the services according the information provided by service publisher.

A list of Web service that matches the service requester’s requirement will be outputted by the discovery system

**Global service registry/Service broker:** This is central registry where all Web Service providers publish their Web Services to be accessed by service end users.

**Binding:** The last and the final part of the process is binding that takes place at run time, after Web Service requester selects from the list of Web Service providers, the one that meet its requirements. Then communication between the provider and requester occurs through SOAP, which is a XML based protocol designed for Web Service communication.

As all communications are taking place through XML, due to this Web services are not actually tied to a particular operating system or programming language, which enables Web services to communicate each other irrespective of platforms used by different systems.

Although WS emerged as de facto standard technology for intrinsic inter-systems interaction across the Internet, however as any technology WS has some drawbacks, in the following sections comparison of advantages and disadvantages of WS is presented.

### 2.2.1 Advantages Web Service

When WS technologies are implemented effectively and used appropriately, it provides plenty of business opportunities, the following points summarizes importance of WS and its capabilities:

- It provides faster and cost-effective solutions by using sophisticated software development on the core business while it uses non-core business programming at client side.
- Aggregates legacy system to modern IT system by using WS to integrate old system thus WS ensures continuity of legacy systems usability.
WS enables enterprises to integrate their business process with service or product users at lower cost, as WS allow business community to share process without sharing technology. This enables even companies with less budget to compete with the giant companies.

Interoperability referring to the fact that WS are meant to be invoked and interacted over the Internet irrespective of platform, programming languages or operating system. This makes WS virtually platform-independent.

Reusability and usability are both Web Services' strengths that once deployed the WS can be used for different systems and different applications.

Now that we have discussed some of the key features that enable WS to dynamically transform the internet from a file and information sharing platform to a full-application remote-execution environment, let us turn our attention to some shortcomings of WS that needed to be addressed in order to exploit inherent advantages and solution that WS provides in terms of application integration and architectural approach.

### 2.2.2 Disadvantages Web Services

As we have already stated earlier no technology that provides absolute solution to every problem, instead a new technology must be seen as both an opportunity and a challenge, it could be recommendable and appropriate to certain organization while it might not suitable to another organization for some reasons. Due to this WS also have some problems that needed to be considered including:

- **WS discovery and its binding process involves registry center where service is being published by the service providers.** That entails the contents of UDDI should be trusted, currently only private UDDI provides trusted contents.
- **Standards and technologies used to implement business process and transactions are still under development in order to fully exploited opportunities offered by WS and SOC technologies.**
- **Problems related to standard core protocols such as SOAP and statelessness of HTTP and HTTPS.** For instance if disconnection or crush occurs during request and response process between client and the server. To initiate another should be started from scratch as it does not record any information about the client. This makes HTTP unreliable protocol.

The following section we will look at the core protocols that the success of Web Service technology entirely depends on, these standard protocols include data formatting language XML, Web Service Description language (WSDL) used to describe Web interface, Simple object access protocol (SOAP) for web service data communications, as well as Universal Description Discovery integration (UDDI).
2.3 Core Web Service Protocols

A number of XML based standard protocols have been used to realize the success of Web Services. As can be seen from figure below a set of standard protocols that are used to implement, develop, invoke and publish Web Services [15]. The protocol stuck includes Extensible markup Language (XML), Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), and Universal Description, Discovery and Integration (UDDI).

![Web Service Stack](image)

**Figure 2-5: Web Service Stack [15].**

XML forms the base of the standard protocols that have established to implement Web Service lifecycles and service client communication; SOAP provides a communication line by encoding messages between service provider and service requester. WSDL provides description language designed to describe Web Service’s invocation details including the name of the service, its operation as well as input and output data types. Similarly UDDI provides an interface to register and publish Web Services. The standard protocols mentioned here will be presented in the following sections in detail.

2.3.1 Extensible Mark-up Language (XML)

To have access to a Web service the input and output data have to be represented in some data format, interoperability between heterogeneous systems has traditionally been very difficult to accommodate due to the different programming language, platform and hardware architecture. Extensible mark-up language XML is a general purpose markup language designed for representing and transferring information between heterogeneous systems.

XML is an extensible mark-up language, that is part of the standard generalized mark-up language (SGML), XML is used for creating mark-up documents and describing data to make easy different application residing on different location to share and exchange information smoothly.
XML tags are not predetermined in advance; this means users can define their own tags. XML is a self-descriptive language designed to carry data. Using XML, contents can be defined separately from its formatting, which in turn enables the content to be used in different application and different presentation environments.

XML has a set of procedures that enables users to define their own tags; these are called Document-Type-Definition (DTD), however, it has some drawbacks including limited set of data type, it does not support date format, the flexibility and its capability to define its own tags makes difficult to manipulate programmatically as it does not conform to XML syntaxes [16].

Due to this limitations, and the need to address these problems, World Wide Web consortium (W3C) has proposed another standard to define XML documents, this new standard is known as "XML Schema" and it provides many advantageous over DTD including:

- Key mechanism that is similar to foreign keys used by relational database.
- Its tags is defined as XML documents, enabling to be easily programmatically accessible.
- Strong typing for elements and attributes, as well as standardized way to represent null values.

The following figures illustrates clearly, how xml-file typically looks like and its data can formulated and presented including basic syntax of the data, elements and attributes that a XML document normally contains.

```xml
<?xml version="1.0"?>
<books>
  <book>
    <author>Carson</author>
    <price format="dollar">31.95</price>
    <pubdate>05/01/2001</pubdate>
  </book>
  <pubinfo>
    <publisher>MSPress</publisher>
    <state>WA</state>
  </pubinfo>
</books>
```

```xml
<xsd:element name="book" maxOccurs="unbounded">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="author" type="xsd:string"/>
      <xsd:element name="price" type="xsd:decimal"/>
      <xsd:element name="pubdate" type="xsd:date"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```
Figure 2-6: An Example of XML Schema

As normally the situation is every technology has some advantages over the previous technologies with some shortcomings, in other words there is no perfect solution, and XML is not different, hence the following we will review and compare issues that makes XML widely accepted standard.

Advantages and Disadvantages of XML

The key features of XML can be summarized in the following points:

- XML provides a standard way of transferring data in a format that can be processed, and exchanged across different operating system, and software applications regardless of programming language used.
- As XML code is very simple and human readable without any programming experience.
- XML provides a standard and common format that widely accepted across the industry and can be developed upon.
- XML tags can be extended and structured in a such way that it meets the organizational needs. XML lets us to separate the data representation from presentation.
- To search data written in XML is easy and efficient. For search engines such as Google, Yahoo, Bing and AltaVista are able to simply parse the description-bearing tags instead of muddling in the data.
- XML documents normally have hierarchical structured as a tree, communication between elements and the root is well defined and structured.

On the other hand, XML has some shortcomings that hamper its usability as standard message format. These limitations are listed below as follows:

- Lack of adequate processing applications
- XML documents are normally large in size; in terms of system performance large documents would limit system performance.
- Inefficient data storage, as it was not initially designed as a database, its algorithm is not as efficient as other database storages.
- Tree structure problems.

In a nutshell, XML generally provides a mechanism to share and exchange data over the Internet, regardless of the applications, platforms and programming languages used by the communicating peers.
2.3.2 Simple Object Access (SOAP)

Due to the nature of heterogeneous environment and application integration problems mentioned earlier, a run time independent communication protocol that is capable of transferring data between Web Service provider and Web Service consumer is paramount. Simple Object Access (SOAP) is intended to help Web Service developers create Web Service and connect heterogeneous applications over the Internet. It provides an open and universal communication bus for application integration, regardless of the programming language, operating system or Object model a certain application might use.

Simple object access protocol (SOAP) is a protocol that enables web services to exchange information over the Internet using HTTP and HTTPS. This means that SOAP provides a communication channel for different applications residing on different servers in a distributed environment. Bluntly put, SOAP is Protocol for accessing web services. Additionally, as can be seen clearly from the figure 2-8, SOAP message contains three main features that facilitate efficient way of information exchange including:

a) An envelope that describes the content of the message and identifies the XML document, as well as how to process the message.

b) A set of encoding rules for describing application-defined data types, sometimes called header information.

c) A convention for representing procedure remote calls, these represent the body element of the message as well as fault elements describing errors and status information.
SOAP is a part of the pool of standardized protocols designed for SOA to provide an open, extensible way for applications to communicate employing XML-based messages over the web, regardless of the type of operating system, object model or for that matter a programming language used.

In terms of historical background, the SOAP has gone through different development stages. Initially SOAP1.1 was developed in 2000, this particular specification comprised of processing model, HTTP binding as well as pattern for remote procedure calls and encoding. The subsequent specifications SOAP1.2 in 2007 has improved substantially the deficiency of the previous specification including adding new fault code, new remote control procedure, as well as additional semantic proposed for different versions of SOAP used.

Although we have discussed SOAP as the sole communication protocol for accessing Web service, however, there is an alternative protocol that also suits for transferring messages, namely Representational state transfer (REST).

REST is a resource-oriented protocol that provides a lighter weight alternative that relies on a simple Unified Resource Locator (URL), for data representation. REST does not use XML instead it can be represented as command separated value (CSV), JavaScript Object Notation (JSON) and Really Simple Syndication (RSS). This means XML parser is not required instead the output can be formatted using language of the application. REST is a new comer compared to SOAP; however, it has already gained wide support within the software development community because it’s easy use, fast
and closer to other Web technologies. Both SOAP and REST are standard protocols used to implement Web service, however, to choose between them one needs to examine the features of each protocol and its suitability for certain application.

The following table 2-1 compares SOAP and REST in terms of their usability based on Web service. Protocols support for criteria such as Client/Server, URI, Caching, Transport layer support; Methods of information and security are compared thoroughly. This comparison is carried out to present which frameworks suitable for different context.

According to [17] work REST provides better solution and out performs SOAP based Web service particularly in terms of mobile environment context as it provides loosely coupled, supports caching and all data types as well as uniform interface and expandable X-links.

### Table 2-1: SOAP VS REST [17]

<table>
<thead>
<tr>
<th>Criteria</th>
<th>SOAP Based Web service</th>
<th>REST Based Web Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server/Client</td>
<td>Tightly Coupled</td>
<td>Loosely Coupled</td>
</tr>
<tr>
<td>URI</td>
<td>One URI representing the service endpoint.</td>
<td>URI for each resource</td>
</tr>
<tr>
<td>Transport Layer Support</td>
<td>ALL</td>
<td>Only HTTP</td>
</tr>
<tr>
<td>Caching</td>
<td>Not supported</td>
<td>Support</td>
</tr>
<tr>
<td>Interface</td>
<td>Non-uniform Interface (WSDL)</td>
<td>Uniform Interface</td>
</tr>
<tr>
<td>Context Aware</td>
<td>Client context aware of WS behavior</td>
<td>Implicit Web Service behavior</td>
</tr>
<tr>
<td>Data Types</td>
<td>Binary requires attachment parsing</td>
<td>Supports all data types directly</td>
</tr>
<tr>
<td>Method Information</td>
<td>Body Entity HTTP</td>
<td>HTTP Method</td>
</tr>
<tr>
<td>Data Information</td>
<td>Body Entity HTTP</td>
<td>HTTP URI</td>
</tr>
<tr>
<td>Describing Web Service</td>
<td>WSDL</td>
<td>WADL</td>
</tr>
<tr>
<td>Expandability</td>
<td>Not Expandable (No hyperlinks)</td>
<td>Expandable without creating new WS using (x-links).</td>
</tr>
<tr>
<td>Standard Used</td>
<td>SOAP specific standards (WSDL, UDDI, WS security)</td>
<td>Web standards (URI, HTTP methods, XML, MIME Types)</td>
</tr>
<tr>
<td>Security/Confidentiality</td>
<td>WS- security standard specification</td>
<td>HTTP Security</td>
</tr>
</tbody>
</table>

### 2.3.3 Web Service Description Languages (WSDL)

Wide variety of languages have been developed in describing Web Services, these languages provide different models and forms to describe Web Services’ characteristics and properties as well as its functionalities. Some rely on plain text
to describe Web Services while others use complex semantic description to express Web Service behavioral properties. The following figure clearly indicates the diversity of these languages. The figure emphasises the significant increasing number of description language evolving from plain HTML to Ontology Web languages. Initially WSDL is discussed in detail as it is a part of Web Service stack protocol and closely coupled with UDDI registry, however, later part of this section other languages will be explained.

The Web services description language is an XML based language that provides a mechanism for describing the essence of the web services. WSDL invokes web services and enables them to exchange data and information. WSDL consist of sequences of messages that define the functionality of the web services, as well as protocol binding and discovering the service locality.

WSDL defines the services as collection of network or endpoint ports, WSDL employs the following elements that are defined in the WSDL document as listed below [18].

- **Types** - define the structure of data in the XML Message.
- **Message** – provides an abstract definition of data that is being transferred in the XML message.
- **Operation** – provides an abstraction description of what actions a specific service can perform for instance operation might define what the input and output of part of an operation.
- **Port type** – contains collection of all operation provided by the service providers.
- **Binding** – a unique protocol that defines data format or structure of the message described in the service as well as operation or endpoint.
- **Service** - a group of related ports or endpoints. A numbered of related ports or end points.
WSDL provides an efficient mechanism that allows Web Service providers to present their software as services that can be published and accessed online by the end users.

WSDL has been modified and updated many times, as previous version had some drawbacks, for instance WSDL1.1 lacked features like reusability interface, provides only four types message exchange patterns, lack of simplicity in defining the operation, as well as inability to capture non-factional requirements and its limited fault description, all these shortcomings paved the way to WSDL2 [21] which provides substantial difference to its predecessor. In nutshell WSDL provides a mechanism to describe web services and its operations as well as information relating to how to interact web services.

Various technologies introduced in this section including (WSDL, UDDI, and SOAP) were designed to operate over the web, they are mainly software components that rely on syntactic description of the web services functionality and properties in XML, however, this method cannot help the users to discovery expected services and has low precision and recall. Additionally in terms of web service discoverability, there has been a shortcoming due to lack of machine-interoperability information regarding web services’ functional and non-functional aspects. To address this issue many researchers and practitioners have been invested lot of effort in producing Semantic web services that is capable of scalable, cost-effective infrastructure for electronic transaction in business and public administration. As result a number of semantic web ontologies have been proposed to add automation and dynamics to current web service technologies, thus significantly reducing efforts required to integrate applications. These ontologies include OWL, OWL-S, WSDL-S and WSMO. In the following sections, we will discuss the above mentioned technologies in detail.

**Web Service Description Language with Semantic (WSDL-S)**

As we mention earlier different standards for creating semantic web services have been introduced. The previously discussed WSDL lacked semantic expressivity that needed to describe web service requirements and capabilities). WSDL-S is a light weight application that enables Web Service developers to annotate semantics to Web services. WSDL-S approach is called light due to the following facts [19]:

- It provides a mechanism that enables us to add semantics to Web services as extension of WSDL.
- It enables us to combine semantic and non-semantic descriptions of Web services.
- Additional semantic descriptions improves software reusability and discoverability, it also facilitates composition of Web services by integrating legacy applications to the business process.

WSDL-S extends WSDL to be able to utilize advanced features of OWL, in other words it combines OWL capabilities to WSDL to provide a meaningful WSDL services. However, WSDL-S provides far greater advantages compared to OWL in many aspects. These advantages can be summarized as follows:

It equips the developers with a capability to describe both semantics and operational level of the Web services in WSDL.
- Due to the external semantic domain models in WSDL-S, WSDL-S provides a flexibility that enables Web service developers to utilize language of their choice to annotate their Web services.

- Utilizing WSDL specification, it is easy to update as incremental approach already existing tools.

On the other hand, OWL-S and WSDL have common similarities on certain issues for instance both languages semantic annotation is based on OWL. Additionally both languages semantic representations are attached to the input and output of WSDL operations, although each does in a distinct manner, furthermore both languages emphasis the significance of precondition and expressing category within given context [20].

In general OWL-S offers better representation in terms of outcome and effect whiles WSDL outweighs in terms of its flexibility of various ontologies for annotation that can be utilized to add semantic to Web services.

### Ontology web Languages (OWL) and (OWL-S)

The Web ontology language (OWL) is a language described by a set of documents, designed to achieve various purpose. OWL is intended to process the content of information instead of just presenting information to humans. It enables greater machine interpretability of the Web content than that provided by XML based contents.

OWL is divided into three sublanguages namely: OWL-Lite, OWL-DL, and OWL-Full. Each of These three sublanguages is meant to be used by specific communities of implementers and developers of users. The following points will shed a light each language:

- **OWL-Lite** is designed for mainly users needing a classification hierarchy and simple constraints.
- **OWL-DL** is intended for users who want the maximum expressiveness while retaining computational completeness decidability. OWL-DL includes all OWL language constructs, however, there are certain limitations involved when applying it.
- **OWL-Full** is intended to be used by users demanding maximum expressiveness syntactic flexibility of RDF with not computational guarantees.

Generally OWL documents can be easily transferred and exchanged between different computers regardless of operating systems, applications and programming languages used.

OWL-S is OWL-based web service ontology; it provides web service developers set of mark-up-language that enables them to describe the web services in unambiguous, and computer –interpretable way. Unlike older technologies e.g. WSDL that lacked the ability to fulfilled major tasks of automating, composition and interoperation, The OWL- S is defined to enable web services to be automated and that involves web services discovery, execution, composition and interoperation. In other words OWL-S’ main objective is automating the advertising, discovering, invocation, composition and monitoring of Web services using network agents or simple clients [3].

As the following figure clearly illustrates OWL-S defines ontology structure that involves:
- The service profile used for advertising, locating Web services, it elaborates service’s functionality, tasks that can be accomplished by employing it as well as its limitation and shortcomings, it also elucidates basic requirements that needs to be met by service end user.

- The process model used for describing the services operation, this involves how web services operates to attain the desired outcome.

- The service grounding used for a protocol level demonstrating how can be accessed the Web service, features used here include message formats, port numbers, mechanism to exchange to input and output data.

![Service Ontology Model](image)

**Figure 2-9: Service Ontology Model[49].**

In Summary OWL-S has been proposed to explicitly annotate semantic information with WSDL. OWL-S supports the specification of the composite process and enables flexible, robust invocation and interoperation between service clients and providers.

**Web Service Model Ontology**

Web Service Model Ontology is another approach proposed as an enabling framework aiming at a total or partial automation of service discovery. It is also formal language used to describe relevant aspect of Web service to enable web services discoverability [20] the framework is designed for providing a conceptual model and language for the semantic markup of all relevant aspects of general services, which are accessible through a Web service interface.

WSMO encompasses four main components assumed to be necessary to support Web service requirements and capabilities, these core elements include:

- Ontologies – allows all the terms and terminology employed by WSMO other components. Ontologies are described in WSMO at a meta level.

- Web service description – illustrates and describes functional behaviour of an actual Web service.

- Goals – represent the client and end users objectives and desires.
Mediators – finally this element deals with interoperability problems among various WSMO components. WSMO’s Web Service specification elucidates Web service’s functional behaviour as well as its capabilities and interfaces. The interface description includes two closely related concept namely choreography and orchestration. Each one of these notions describes different tasks in WSMO frame, for instance the former contains information required to interact the Web Service, while the latter provides the necessary details for the execution of all the component services and may be a proprietary item for the provider that are not accessible to others.

Despite the facts that OWL-S and WSMO are two main approaches that deal with the service composition in the Semantic Web Service, however WSMO significantly differs from other ontology languages like OWL-S. For instance OWL-S approach does not separate user request from Web service provider advertisement as the profile ontology is used to accomplish both task, on contrast WSMO makes separation between the two by using a Goal to specify the user request and Web service description to define what service does and its functionality as well as its capability. Another important difference between OWL-S and WSMO is that the OWL-S defines the externally visible behaviour of the Web service, while WASMO also models how one service utilizes another service in order to accomplish specific tasks. The following table2-2 summarises the core difference between various semantic languages including OWL-S, WSMO, and WSMO-Lite, the main difference that can be deduced from the table2-2 include syntax used, different parts each language comprised, and different kind of reasoning and logic employed to describe behaviour of Web Service.

**Table 2-2: Core Difference Between OWL,OWL-S and WSMO [57]**

<table>
<thead>
<tr>
<th>Language</th>
<th>OWL-S</th>
<th>WSMO</th>
<th>WSMO-Lite</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>OWL</td>
<td>WSML</td>
<td>RDF/XML</td>
</tr>
<tr>
<td>Reasoning</td>
<td>Logic language</td>
<td>Logic and rule language</td>
<td>Logic language</td>
</tr>
</tbody>
</table>

In a nutshell, WSMO Web service is the most significant semantic Web Service frame model available today that is designed to attain selection, composition, mediation, execution, monitoring, etc. involved in both intra- and inter-enterprise integration of Web Services.
2.4 Web Service Discovery Methods

Web service discovery can be defined as a method of finding and searching Web services that are suitable and specific to requirements of user’s development, which can be challenging and time-consuming [23]. As already indicated Web service discovery involves three steps firstly publishing Web service in order to be discovered, secondly finding Web service description and retrieving it in order to invoke, and thirdly to invoke and bind it [25].

Due to the increasing numbers of Web services available online, Service Oriented Architecture (SOA) has gained mass appeal throughout IT industry. In SOA the interaction between Web Services takes place dynamically, and SOA allows the Web services to communicate and interoperate among different application and systems. However, as number of Web service increased rapidly, the problem arise from discovering and selecting appropriate web service because of massive number of Web Service available over the Internet, also another obstacle that obstructed finding appropriate Web Service is lack of appropriate search mechanism as most of the search mechanism is based on syntax rather than semantics, additionally failure of existing search mechanism to involve non-functional parameters such as Quality of Service “QoS” and Cost of Service “CoS”.

There are various methods that can be used to discover Web services including Universal description discovery and integration (UDDI), using popular search engines, publication sites and registries [26][28]. The above mentioned methods differ in the ways to retrieve information and in terms of its functionality. Following are some description of each method.

2.4.1 Universal Description, Discovery, and Integration (UDDI)

UDDI is XML based and part of web service pool of protocols used for registering and publishing as well discovering web services. UDDI not only provides public web services across the Internet, but it provides as well a platform that enables the enterprises to exchange their web services [19].

UDDI comprises of three major components, namely white, yellow and green pages, each of these pages cover different information for instance business name, contact information, and text that describes the nature of the web service is defined in the white pages, whilst yellow pages describe information relating to industry classification using standard taxonomies, similarly green pages provides technical aspect of the web services. In terms of the data structure within UDDI, it consists of four construction entities including Business entity, Business Service structure, a binding Template structure, and a Model structure. The following figure clearly demonstrates how these four structures are connected and linked [19].
The above figure classifies different element that UDDI provides, for instance business Entity provides information relating to web service provider or service publisher, while the business Service describes information about a particular web service including technical services, the binding Template gives technical information needed to bind and interact with the target web service; whereas tModels is used to represent interfaces.

Although UDDI provides essential platform for registering the web services, there are shortcomings and problems that needs to be addressed, one of the problems is its scalability as number of web services increases significantly, the possibility of locating a certain web service efficiently and effectively decreases dramatically. Additionally with the rapid increase of privately registered web service on distributed environment creates bottle-neck thus degrading the performance.

Another limitation is lack of availability of largest public registry which was closed June 2006, thus makes difficult to search Web service [27]. Similarly another drawback of UDDI registry is its inability to represent QOS qualities hence unable to rank Web services of similar functionality but different QOS.

Due to these limitations, in 2007 IBM announced the demise of UDDI by stating that UDDI had reached its limit and does not meet SOA prevailing realities and recommended that is a high time to design a new registry that addresses the building block enterprise approach of SOA, thus abandoning global registry[31].

Many researchers conclude that UDDI registry does not offer a preferred Web service discovery method that would satisfy user’s desired and expected results, thus concluded that search engines might preferably be alternative to UDDI registry [26][28]. There is other registry that is not as popular as UDDI registry but provides an alternative to UDDI.

For instance ebXML which stands for “Electronic business XML” with Web service Inspection Language “WSIL” which provides centralised Web service registry that stores more data than broader than UDDI. It aims at providing more global e-business infrastructure focusing on the goal of open system, where companies of all size can participate to conduct a business with low entrance barriers.
ebXML offers two tightly coupled components namely the registry and the repository. The registry provides information such as service interfacing, reference system implementation and information model, while the repository provides physical back end store. ebXML also suffers from lack of granted quality registered services, More-over any semantic ontologies will have to be externally utilised.

2.4.2 Search Engines

General purpose search engines have been used to provide methods to discover Web services; these search engines include Google, Alex, Baidu, Bing and Yahoo. Search engines are capable of handling large amount of information and thus able to cope with the ever-increasing number Web service [30]. Web services are located and discovered using Web service’s WSDL file types [27].

General purpose search engines for Web service discovery involves three separate business entities, namely service provider, search engine provider and consumer. Discovering a particular Web service depends on how the Web service is published, indexed and queried [32]. Despite its promise to offer a better alternative of accessing Web service than conventional UDDI registry, the general purpose searching engines also have some shortcomings that limit their performance; these drawbacks include [29]:

- WSDL weakness to describe Web service on searching engine, lack semantic expressivity.
- Inability to include non-functional properties such QoS and CoS.
- Search engines do not present information such as binding information, ports and endpoints.
- Web services are treated as any other web page, which in turn make the relevant Web service information dispersed throughout result list.

To show further the weakness of search engines [28][29] have made a comparison between search engines and UDDI business registry, checking their respond to many features including their support of web service specific information such as business information, tModels, ability to store WSDL document, search and caching capabilities.

**Table 2-2: Comparison on UBRs and Search Engines [28].**

<table>
<thead>
<tr>
<th>FEATURES</th>
<th>UBRs</th>
<th>SEARCH ENGINES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains Business Information</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>Uses tModels</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>Is publishing (listing) voluntary?</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Any service-like structure?</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>Stores WSDL Documents</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>Any update interval</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>Any support for range-based searching?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Any Support for caching?</td>
<td>No</td>
<td>Possibly</td>
</tr>
<tr>
<td>Search capabilities</td>
<td>Limited</td>
<td>Keyword matching</td>
</tr>
<tr>
<td>Any Web service subscription business model</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>Can Handle versioning?</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Validates, governs, or secures Web services?</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Any support for web service specific measurements?</td>
<td>NO</td>
<td>NO</td>
</tr>
</tbody>
</table>

The above comparison table clearly illustrates the limited search capabilities that both search engine and UBRs share in common; it also demonstrates that search engines do not provide support for web service’s specific information; on the other hand it shows that UBRs provide far better specific information related to Web service, this is due its initial design principal. From this comparison one can conclude that UBRs and search engines utilizes different techniques to discover Web services, to improve their performance in discovering Web services it might be recommended to combine different techniques used. Additionally annotating WSDL files with semantic information can increase efficiency of discovering relevant Web services without changing internally search engines design principal.

2.4.3 Publication Site

Publication sites are web sites that contain registries for Web service discovery; occasionally these web sites are referred as service registries, portals or directories. For example Web service portals such as XMethods, RemoteMethods, BindingPoint, WebServiceList are typically Web service publication site that allows the use of keyword queries to find Web services with matching names and description [28]. Searching is done using Hypertext Mark-up language (HTML). Additionally, it allows displaying QOS properties of the Web service.

The recently developed centralized domain specific Web service registries in life science have increased the popularity of Web service publication sites [33].

The following figure depicts large number of Web services provided by RemoteMethods, which provides source of reliable Web services, where Web services are categorized in terms of services, each category has a number of Web services grouped according to their functionalities. Web service user can select, browse and search to locate desirable Web services. Number of Web services in each category is also displayed, when user selects a specific Web service, information relating to that particular Web service is listed including price, description, hitting, review and ratings.
Similarly the following figure is another example of XMethods which also provides a flat listing of Web services from individuals and organizations. User can browse XMethods interface list of Web services sorted by the time submitted, addition to this friendly Web service interface, it offers pragmatic interface to the registry, but it does not provide searchable interface which makes evaluating and comparing Web services functional and non-functional features very difficult [34].

The XMethods interface tabulates the publisher, style, Web service name, description and implementation of Web service. XMethods support for services have XMethods as listed owner, but it does not support the services published by other entities, it also recently setup a processes to ensure that WSDL files for each service stays accessible and technically valid.

Both XSince XMethods and RemoteMethods take a decentralized approach, transactions occurs at the final stage between user and service provider; this means both web sites do not provide purchasing and transaction functionalities. On the other hand, both platforms enable end users to invoke the Web services, confirm its functionalities and response time.
Figure 2-12: List of Available Web Services Provided by Xmethods[55].

To further publication site, another example of is presented in the following figure is ProgrammableWeb, which is most popular of intermediary publication site [35], it provides a friendly application program interface (API) with searching capability, allows user to browse the Web service by category, and it lists the Web service tabulating by name, description, category and last updated. The ProgrammableWeb tracks the latest update of API news, offers the API developers and providers possibility to add a new applications. Although the interface offers rich and different methods to assist the user, however, once listed the Web services, user may need additional assistance to select appropriate Web service.
As the number of Web services increased, so the number of publication sites, the following table gives an overview of the public Web services back in 2008. Although the data in the table 2-4 is somewhat outdated, however, the readers can draw a conclusion the exponential increase of Web services and corresponding publication sites. Although Both UBRs and PindingPoint registries were discontinued, however, number of both public and private web service registries mushroomed.

Table 2-3: Overview of Web Services Directories[28].

<table>
<thead>
<tr>
<th>NAME AND LINK</th>
<th>OWNER</th>
<th>ACTIVE</th>
<th>STATUS (August 2008)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Universal business registry (UBR)</td>
<td>IBM, Microsoft, NTTcom, SAP</td>
<td>2002-2006</td>
<td>Discontinued in 2006, with more than 50,000 entries. (The objective of UDDI was reached)</td>
</tr>
<tr>
<td>(inactive)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PindingPoint</td>
<td>Acclaim IT solutions Ltd</td>
<td>N/A -2006</td>
<td>Discontinued in 2006 with more than 400 services (&quot;Market too slow to adopt Web services)</td>
</tr>
<tr>
<td>(Inactive)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eSigma.com</td>
<td>eSigma</td>
<td>2003-today</td>
<td>295 Web services</td>
</tr>
<tr>
<td><a href="http://www.e-sigma.com">www.e-sigma.com</a></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
To conclude our discussion about Web service registries publication sites, there are issues and limitations related to these web services registry including lack of standard registry as UDDI, this in turn lowers the trustworthiness of these sites, as result of this drawback both Woogle and Salcentral closed their portals. According to [27] 60% of Web registries are valid web sites while rest is inaccessible. Most of publication sites display a list of Web services to filter and find a suitable Web service is time-consuming process.

The following table 2-5 compares the three Web services discovery methods discussed in the previous section, evaluating properties such functional Web service properties, non-functional Web services properties and graphical user interface used by each method, as well as classification capabilities and ranking facilities.

**Table 2-4: Comparison on Web Services Discovery Methods [36].**

<table>
<thead>
<tr>
<th><strong>Criterion</strong></th>
<th><strong>Support</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RemoteMethods</strong></td>
<td>InfoGenius</td>
</tr>
<tr>
<td><strong><a href="http://www.remotemethods.com">www.remotemethods.com</a></strong></td>
<td>Started as Web development directory; 358 Web services.</td>
</tr>
<tr>
<td><strong>StrikeIron</strong></td>
<td>StrikeIron</td>
</tr>
<tr>
<td><strong>WebServiceList</strong></td>
<td>IT Netix,Inc</td>
</tr>
<tr>
<td><strong><a href="http://www.webservicelist.com">www.webservicelist.com</a></strong></td>
<td>N/A –today</td>
</tr>
<tr>
<td><strong>WebserviceX.NET</strong></td>
<td>Generic Objects Technologies Ltd</td>
</tr>
<tr>
<td></td>
<td>411 Web services</td>
</tr>
<tr>
<td><strong>XMethods</strong></td>
<td>XMethod</td>
</tr>
<tr>
<td><strong><a href="http://www.xmethods.com">www.xmethods.com</a></strong></td>
<td>N/A –today</td>
</tr>
<tr>
<td></td>
<td>503 Web services</td>
</tr>
</tbody>
</table>

The following table 2-5 compares the three Web services discovery methods discussed in the previous section, evaluating properties such functional Web service properties, non-functional Web services properties and graphical user interface used by each method, as well as classification capabilities and ranking facilities.

**Table 2-4: Comparison on Web Services Discovery Methods [36].**

<table>
<thead>
<tr>
<th><strong>Criterion</strong></th>
<th><strong>Support</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Functional Web Service Properties</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Non-Functional Web Service Properties</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>An Effective Graphical User Interface</strong></td>
<td>Textual List, Focus on Business Information</td>
</tr>
<tr>
<td><strong>Classification Facility</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Search and/or Browsing Facility</strong></td>
<td>Limited</td>
</tr>
<tr>
<td><strong>Ranking Facility</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>Result List Sort Facility</strong></td>
<td>No</td>
</tr>
</tbody>
</table>
As can be seen from the table 2-5 publication sites meet the criteria listed on left column, all the three Web service discovery support for functional properties of Web services, on the other hand search engines provide somewhat limited capabilities of functional Web service properties. In contrast to the other Web Service discovery method, the most publication sites provide support for non-functional properties such as QOS, throughput, and reliability. From the table 2-5 can be deduced that all the three Web service discovery methods in the table have not addressed a potential problem with graphical user interface, according to [37][38] current research regarding Web service discovery component such as search and ranking has not been adequately addressed the issue of graphical user interface and presentation of Web services. Additionally UDDI registry does not support ranking and result list sort facility, this due to the fact that UDDI focuses on functional aspect of the Web Service and overlooks the non-functional aspect of the Web Services making difficult for the user choose between Web Services of similar functionalities.

2.5 Criteria of Web Services Discovery

With large number of Web service available on Internet, the task of finding desirable and suitable Web Service from pool of Web service becomes a daunting task, this due to the following issues [39]:

- Variety and large number of Web Service
- Most searching is based on a simple keyword search and syntax instead of semantic ontologies.
- Lack of standard WSDL format that Web Service description complies to.
- The UDDI registry contains static information that only updated when registry is initially stored or manually updated in other words lack UDDI dynamic information update.

With above mentioned obstacles and issues, Web service discovery has several criteria to be satisfied in order to be effective and efficient method. Depending on what contexts, applications and environment as well as Web service discovery method is used the criteria to be adhered may show a discrepancy [40]. However, there are general key criteria to be considered when designing Web service discovery, these includes:

- In order method to ease integration and enhance seamless interoperation, it should be standard-compliant.
- Functional Web service properties should be expressed in such way that user grasps easily the capabilities of the Web Service, additional textual description of Web service functionality may be added [26].
- Non-Functional Web service properties, in addition to functional Web service, non-functional aspect of the Web service should be addressed to enable the user decide which Web service satisfies and meet the required functionality with reasonable level of QoS [29]. The non-functional aspect of Web service discovery includes QOS parameters such as performance, reliability, integrity, accessibility, availability, interoperability and security.
- Scalability is very important criteria as both number of available Web services and potential users may increase rapidly, the discovery system should be to accommodate both the number of Web service and users as well as subcomponents and subsystems [40].
Effective graphical user interface, which enables user to search and browse the available Web service, additionally it is important that Web service discovery should possess a mechanism to classify and filter the available Web services to assist the end user select the preferred Web service [41].

Discovery mechanism should be capable of coping faults and network changes, without considerable function losses.

Web service discovery should be able to accommodate different platforms, applications, data formats as well as different ontologies used by semantic Web services.

These above-listed points are highly considered to be most important criteria to support the user in web service discovery. It still needs to be identified on what to base each criterion in order to be effectively used in web service discovery. For example, to provide effective web service discovery based on non-functional properties it would be necessary to determine the desired non-functional property combination to aid in the web service discovery process.
3 Method

As research area matures and the number of available published work increase sharply, there is a need to systematically synthesize available research evidence. The well-established evidence-based disciplines such as medicine and education have already adopted research method called systematic review, following the suit many computing discipline including software engineering have also recently started to utilize this research method [42]. As result of the systematic review research method, another new method has been emerging that primarily focusing on classification schemes of specific topic of interests. The latter method is called systematic mapping studies [2]. Mapping studies are typically used to give a general overview of a studied area and is normally done before implementing literature review. Although several surveys and comparisons concerning Web service discovery have been reported, for instance in survey performed by [44] is one of the earliest survey reported in this research area, although extensive comparison and evaluation were presented, however, their work mainly focused on architectural aspects and data models facilitating discovery while overlooking the importance of semantic web service discovery. In contrast [43] presented a survey of web service discovery systems which mainly focuses on methods of using semantic matching, although they introduced a taxonomy that classifies different approaches of Web service discovery and made comparison of advantages and disadvantages of each discovery method, however, it was not inclusive as much attention has not been given to semantic matching techniques that use other than logic based reasoning. On the other hand [45] presented another survey where they define a set of criteria to classify the existing Web service discovery approaches coupled with different algorithms used by each method, although undoubtedly one of the most comprehensive survey in this area, however, its shortcomings can be attributed to the negligence of most important criteria such as multi stage matching, support of UDDI matching and support of different ontologies matching criteria. Another comprehensive and extensive survey and review has been carried out by [46], in this survey eight well defined criteria have been proposed to compare different approaches’ strength and weakness, additionally, service discovery evolution has been discussed thoroughly, despite commendable effort made by these researchers, this survey overlooks architecture aspect of Web service discovery and non-functional Web services discovery aspect such as QOS parameters are not adequately dealt. Generally, the majority of the published studies are informal literature surveys with no defined research questions, no search process, and no defined systematic data extraction.

This thesis main objective is to make classification of papers published in the area of Web Service discovery mechanism following systematic mapping study specific rules and guidelines indicated in the paper [2]. It is a repeatable process for extracting and interpreting available literature related to the research objectives. This approach begins with identification of research objectives and developing research protocol followed by search strategy with well established inclusion and exclusion criteria. The following section study design of this work will be elaborated further.
3.1 Study Design

As mentioned earlier, systematic mapping study research method will be utilized in this work as it provides means of evaluating and interpreting the literary papers. Additionally owing to its structure the method aims to present a fair evaluation of the research ensuring that is trustworthy, rigorous and auditable[48]. This achieved by using predefined search strategy that is designed to be wholesome and complete. According to [2], there is a process of conducting search, initially one must define research question by narrowing the scope of the area researched, and subsequently search is conducted, followed by filtering papers to exclude irrelevant work. Then classification stage will be next and finally data extraction and mapping process are to be carried out to conclude as can be seen from figure3-1.

**Process - steps**

The following points summarize some of the features that are unique to systematic mapping study [2]:

- Well defined research questions and clear methods to perform the mapping study.
- Well defined search strategy that enable researchers to accumulate the relevant literature.
- The search strategy that can be documented so that anyone interested can access to check its rigorousness and completeness of the work.
- It requires explicit inclusion and exclusion criteria to exam the potential primary study.
- It often gives a visual summary, the map, of its results. It requires less effort while providing a more coarse-grained overview.
- It allows content analysis to code, extract and validate the literature.

Moreover, for this particular work systematic mapping is chosen to:

- Provide overview of the existing research already performed on Web services discovery mechanism.
- Analyze and identify existing gaps in the current research in order to establish future work direction.
- Attempt to generalize findings across collected research papers and provide constructive criticism.
- To identify the quantity, type of research and results available within it, as well as mapping the frequencies of the publication to see the trend and the direction of the research area. Additionally publications forums will be mapped to make it easier for the researchers interested in the topic.

---

**Figure 3-1: Taken From [2].**
3.2 Data Collection Instrument

Once the objective of the research has been determined, in this section we will introduce review protocol regarding the collection of data. This relates to the questionnaire used to gather the data relating to the Web Services Discovery Mechanism “WSDM”. These questions are formulated in order to mine and extract relevant information regarding the trends and data pertaining to the research papers written about the topic in the research. Following is list of questions and reasoning behind gathering data under each survey question:

Q1. Where the research has been published, the year of publication and type of the publication venue (Journal, Conference, Workshop, other)?
- This question has been raised in order to determine venue where the work has been published and to restrict the time of publication.

Q2. What is category of the paper? (Survey paper, Evaluation Research, Solution proposal, Position paper, Experience paper, other/Suggested Category)?
- This is to classify the published papers in terms of its category that it belongs.

Q3. What is the major contribution of the paper (Process, Methodology, Models, Tool, and Area Overview)?
- This is to determine the main contribution of the paper in terms of particular part of software engineering discipline.

Q4. What is the motivation of the research and what research gaps have been reported by author/authors?
- To explore the reason behind the authors effort and endeavors in publishing the research.

Q5. What are the main findings and conclusion of the paper?
- To explore the evaluation and the findings of the work

3.3 Data Source and Search Strategy

The primary task of systematic mapping is data collection or data gathering. In order to gather most relevant papers a search strategy has to be in place to search scientific databases. According to [48] a proper method to create a search string is to structure them in terms of population, intervention, comparison and outcome. To ensure that all the relevant works being searched the following search strings are employed: “Web Services Discovery mechanism”

All the papers were listed, bibliographic details will be entered into a spreadsheet as the survey documents is completed for a paper. The above listed search string will be used to perform pre-review search to explore and find most relevant papers. The following well-known scholarly databases will be searched: ACM Digital library, IEEE explorer, Springer link and Elsevier.

Similarly, Proceedings of Web Services conferences will be investigated as well as reports on international conferences regarding with Semantic web and generally Service Oriented Architecture conference to check any relevant information regarding the research questions.
In order to classify the papers exclusion and inclusion criterion will be utilized, the goal of defining selection criteria is to find and include all relevant published literature. The selected papers are published between 2005 up to 2013. Only studies published in English language are selected. Papers on panel discussions, tutorial summaries, presented slides, prefaces and editorials are discarded. In case, the different version of same paper was published, the comprehensive version is to be included and discarded papers discussing only some parts of comprehensive version.

**Table 3-1: Inclusion and Exclusion Criteria.**

<table>
<thead>
<tr>
<th>Inclusion Criteria</th>
<th>Exclusion Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Research papers published in journals (IEEEExplore, ACM digital, Science Direct and Springerlink), Conferences and workshops.</td>
<td>Panel discussions, presented slides, prefaces, tutorials and book reviews.</td>
</tr>
<tr>
<td>Material published only English language.</td>
<td>Paper is too short or in the form of abstract or published in languages other than English.</td>
</tr>
</tbody>
</table>

All the papers will be stored in repository and can be validated; when the papers will be classified according to their categories the coding schema will be introduced depending on the contribution of the paper in terms of methodology used and observing the abstract and conclusion of each paper to classify according to the papers motivation and contribution to research area in question.

**3.4 Content Analysis**

Based on research questions the following string “Web service discovery mechanism” was utilized to search databases. To get all the relevant papers the search string was created and formulated as follows: ("Web service" Or "Web services discovery mechanism"). Searching systematically through academic and state of the art publication databases including relevant digital libraries such as ACM portal digital, Elsevier, IEEE Xplore digital library resulted in 61 studies. Searches were conducted only scientific databases accessible online as can be seen from the table 3-2, books and other printed sources were not specifically explored as these digital libraries include all high impact factor conferences and journal publications, hence, there was no need to look into other electronic source of information, only three very important articles has been selected from other journals such as IGI global and IJC.
Table 3-2: Selected Electronic Databases.

<table>
<thead>
<tr>
<th>Electronic Database</th>
<th>URI</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE</td>
<td><a href="http://ieeexplore.ieee.org/Xplore">http://ieeexplore.ieee.org/Xplore</a></td>
</tr>
<tr>
<td>ACM</td>
<td><a href="http://dl.acm.org">http://dl.acm.org</a></td>
</tr>
<tr>
<td>Springer</td>
<td><a href="http://www.springerlink.com">http://www.springerlink.com</a></td>
</tr>
<tr>
<td>ScienceDirect</td>
<td><a href="http://www.sciencedirect.com">http://www.sciencedirect.com</a></td>
</tr>
<tr>
<td>Elsevier</td>
<td><a href="http://www.elsevier.com/">http://www.elsevier.com/</a></td>
</tr>
<tr>
<td>IGI</td>
<td><a href="http://www.igi-global.com/">http://www.igi-global.com/</a></td>
</tr>
<tr>
<td>IJC</td>
<td><a href="http://www.ijcsmc.com/">http://www.ijcsmc.com/</a></td>
</tr>
</tbody>
</table>

Once list of papers was obtained, the classification phase has been carried out. The first stage all the relevant papers has been gathered, redundant and duplicate studies has been eliminated, this led to the collection of 61 papers, then reading the abstracts and the introduction of the gathered articles as well as employing the exclusion criteria the number of papers were reduced to 45papers. Subsequently in order the searching to be systematic and make sure any important information not to be missed, reference of most of the studies were examined, although this method was not as thorough as the previous stages, however, this led the inclusion of further 7papers. In the stage four supervisor will be consulted to recommend of there was some literature that have been overlooked before classification phase will be concluded.

The following figure demonstrates the four different stages data selection process passed through before reaching final number of papers based on this systematic study work. This selection is based on the researchers [2] where keywords are extracted from the abstract and introduction part of the papers as well as the conclusion to sort out the papers into different categories. However, initially demographic disperse of the papers selected will be presented followed by graphical demonstration of publication venue.

Figure 3-2: Study Selection Process.
3.4.1 General Overview

As can be seen from the following figure, papers surveyed were conference proceedings publications with 48% of all papers collected in this study and 39% journal articles, while 10% of the publications were workshops and only 4% belongs to other category such as books and other sources. It is important to note that most of the papers were scholarly database such as IEEE, ACM, Elsevier and Springerlink as adequate number of papers were found in these databases.

As can be seen from the publication venue figure below, almost half of papers published were conference publications indicating the importance of discovery in the Web service lifecycle and great interest shown by both academy and practitioners in this fundamental area of research in distributing computing. It has been discussed and debated in how to develop and provide different solution to the prevailing problems related to Web service discovery. Moreover, close to 40% of the papers were journals, this clearly shows that the Web service discovery is very active and development topic as well as maturity of researched area and its relevancy.

![Publication Venue](image)

**Figure 3-3: Publication Venue.**

Table 3-3 shows number of paper of papers published by each electronic data source, IEEE is leading by publishing 22 papers out of 52, which is 42% of the selected papers, this clearly demonstrates IEEE being leading-edge conference proceeding publisher every year organized by academia and industry worldwide; while ScienceDirect is second largest publishing 14 papers out, equivalent to almost 27% of the selected papers, while rest combined together produces 31% of the selected published papers. ACM and IEEE were gathered most of the conference proceeding process and workshops while others were article published in journals. There are also few other journals that published very relevant articles such IGI and IJC which categorized as others in the publication venue figure.
Table 3-3: Number of Papers Published in Data Sources.

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Number of papers Published</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE</td>
<td>22</td>
</tr>
<tr>
<td>ACM</td>
<td>8</td>
</tr>
<tr>
<td>Springer</td>
<td>5</td>
</tr>
<tr>
<td>ScienceDirect</td>
<td>14</td>
</tr>
<tr>
<td>IGI global</td>
<td>2</td>
</tr>
<tr>
<td>IJC</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 13 illustrates the number of publication in each year, in other words publication distribution over the years that selected papers were written. The rapid growth of the research indicates the massive interest, effort and money has been invested in the area, the figure above can be deduced the trend of continual increment that Web service discovery research has gone through. For instance EU spent about 70 million Euros in semantic web service discovery research, which run from 2002 to 2006, this gives a good impression of the importance being currently put on this field of Web service discovery research [24]. In the years (2005, 2006 and 2007) combined together the number of selected papers are 10 papers, while (2008, 2009, and 2010) combined together the selected papers are almost 30 papers, this clearly shows massive increase in the number of the publication of this period, this could be due to the fact that as more and more enterprises as well as organizations adopted SOC, number of publications addressing underlying issues of SOA increased. However, there are only 14 papers selected in years 2011 to 2013 this could be attributed to the level of maturity already achieved, and other application architectures such as cloud and grid architecture as alternative to SOA that might divert the attention of researchers. On the other hand the less number of publication selected in the years of 2011 to 2013 does not show the decline and lack of interest in the research area, in contrast the number has increased but become more sophisticated in nature dealing with diversity application area that not necessarily providing solution to the prevailing Web service discovery problems, thus some of these literatures have been skipped.

The figure below gives you overall picture and the distribution of paper throughout the years, this indicates that at the beginning of the millennium the research area started with rapid growth reaching at the peak after a decade and still continues to be one of dominant and fundamental research area.
3.4.2 Classification of the Studies

During the data classification phase the collected paper were grouped into categories according to their contents and research focus. The selected papers are grouped together in different categories of themes considering if they are focusing on a common set of problems and proposing solution Web service discovery mechanism. These categories were identified after analyzing data extracted from papers to address research objectives. As number of papers selected addresses different issues, categories that contain large number of papers are further sub divided into sub categories. More over articles that covers wide ranges of subjects are included into multiple corresponding categories or subcategories. The selected papers were initially classified into two broad categories namely Web service discovery based on architecture category and a category based on Web service discovery method. As can be seen figure3-5 which is turn the architecture based category is subdivided into centralized and decentralized, subsequently the centralized category is sub divided into portal architecture based and UDDI architecture based, and latter one can be sub divided hybrid based architecture and broker based architecture. Similarly the decentralized architecture is sub divided into two sub categories namely Agent base architecture and Peer to peer (P2P) based architecture.

On the other hand the other main category, Web service discovery methods based category is sub divided into two categories, functional and non functional based categories, in which each category is subdivided into multiple categories, for instance functional based categories is further classified into many two main groups namely: Syntactic matching and Semantic matching, the former one can be still classified into Key word matching and interface based keyword matching, while latter is subdivided into ontology based matching, context information based method matching, functional semantics based matching and information retrieval method based matching. On the other hand non-functional based method is classified into QOS based matching, usability based matching, providers and consumers
expectation based matching. The following figures clearly depict the above mentioned classifications, the first one illustrates the architecture based categories and its classification, while the second one is a illustrative of Web service discovery based on functional and non-functional categorization.

Figure 3-5: Categorization of Web Services Discovery Architectures[50]

The following figure 3-6 depicts categories based on functional and non-functional Web Service discovery techniques, illustrate different branches and groups that can be divided into Web service discovery versus architecture categorization.

Figure 3-6: Functional and Non-Functional Web Service Discovery Categorization[50].
The above categorization will be based on our findings, observing how different discovery approaches have been utilized and corresponding number articles in each section, which in turn describes level of research maturity in each category.

The following table 3-4 provides list of above illustrated categories and corresponding number of papers based upon research motivation and goals.

**Table 3-4: Categorizing Selected Papers.**

<table>
<thead>
<tr>
<th>Categories</th>
<th>Number of Cases</th>
<th>Percentage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture</td>
<td>12</td>
<td>20%</td>
<td>Papers dealing with architecture aspect of Web service discovery mechanism.</td>
</tr>
<tr>
<td>Semantic Frameworks (Functioonal Based WSDM)</td>
<td>22</td>
<td>37%</td>
<td>Papers proposing a unified method of easing web service discovery process based on functional details of advertised services matching with requested functional description.</td>
</tr>
<tr>
<td>Surveys</td>
<td>10</td>
<td>17%</td>
<td>Papers providing reviews and survey of current available methods and approaches for Web service discovery mechanism.</td>
</tr>
<tr>
<td>Non Functional Based WSDM</td>
<td>15</td>
<td>25%</td>
<td>Papers providing solution toward easing Web service discovery mechanism focusing on non functional aspects of Web service discovery such QOS, usability, level of user agreement etc.</td>
</tr>
<tr>
<td>Total</td>
<td>59</td>
<td>100%</td>
<td></td>
</tr>
</tbody>
</table>

As can be seen from the above table 3-4, the numbers of cases exceed the number of selected papers, this is due to fact that some of the papers cover multiple topics, as result some papers might belong to different categories. Semantic frameworks 37%, covering all functional based Web Service Discovery Mechanisms listed in figure 15, while 20% of selected papers cover architecture aspects of Web service discovery mechanisms, 17% of the selected papers cover reviews and surveys covering current research in the field of Web service discovery mechanisms, while rest of the papers 25% covers non functional aspects of Web service discovery mechanisms including Quality of Service, usability and other parameters.
4 Results and Analysis

Prior to presenting the finding from the analyses, during the coding phase detailed information has been collected from the selected papers examining the research approaches. In addition to categorization of the selected paper, general research facet has been observed to classify the research approaches used in the selected papers. In order to sort out the different research approach, Wieringa’s classification approaches has been utilized, where research approaches are classified into six categories namely Validation research where the authors defined as “Techniques investigated are novel and have not yet been implemented in practice techniques”, another approach facet is evaluation research which is defined “Techniques are implemented in practice and an evaluation of the technique is conducted, that is shown how the technique is implemented in practice (Solution implementation) and what are consequences of the implementations in terms of benefits and drawbacks (Implementation evaluation) including problems industry. According to authors third approach is Solution proposal which is defined “A solution for a problem is proposed, the solution could either be novel or a significant of extension of existing technique. The potential benefits and applicability of the solution is shown by a example or good line of argument”, the fourth approach is Philosophical papers which is defined as authors termed as” Papers that sketch a new way of looking at existing things by structuring the field in form of a taxonomy or conceptual framework. The fifth research approach can be Opinion papers which are defined as “Papers that express a personal opinion of somebody whether a certain technique is good or bad or how things should be done. They don’t rely on related work and research methodologies. Finally the last and sixth approach is termed as Experience papers which are defined “Papers explain on what or how something has been done in practice. It has to be the personal experience of the researcher” [47].

Using above listed research approach it is easy to interpret and use for classification without evaluating each paper in detail, these research approaches can be further classified as two main categories empirical and non-empirical research approaches. According to [2] the empirical approach includes evaluation and validation research approaches where non-empirical research approach solution proposal, philosophical papers, opinion papers and experience papers. Additionally authors mentioned to classify research approach evaluation research can be excluded if no industry cooperation or real world project is mentioned. Validation research is easy to pinpoint by checking whether the paper states hypotheses, uses summary statistics (eg, figures like scatter diagrams or histograms) and describes the main components of an experimental setup. Using above research approach classification resulted in the following table that describes the classification of selected papers according to the authors. Frameworks and survey papers are combined as philosophical papers, where unified frameworks supporting and improving techniques alleviating the process of discovery mechanism are compiled as framework providing set of guidelines and rules. These papers include both architectural aspect of the Web service discovery focusing papers as well as semantic and non-semantic discovery approaches focusing papers. Papers providing novel solution or significantly extending existing techniques in Web service discovery are categorized as solution papers which represent a huge bulk of selected papers. Opinion and experience papers also are two categories where authors of those papers state their position in regarding with techniques
of discovering Web service mechanism, this category belongs case studies, developed tools for discovering Web services and specific systems designed to address the issue of accelerating the process of finding and discovering the desired Web Services that are dispersed over Internet.

Table 4-1: Paper Types [47].

<table>
<thead>
<tr>
<th>Paper Type</th>
<th>Number of Cases</th>
<th>Percentage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Philosophical papers</td>
<td>20 papers</td>
<td>38%</td>
<td>This category encompasses all the Web services discovery frameworks and Survey papers. Framework papers suggest a set of rules and guidelines that can be structured approaches of Web service discovery mechanism, while survey papers present survey about existing approaches and methods for Web service discovery mechanism.</td>
</tr>
<tr>
<td>Validation Research</td>
<td>8 papers</td>
<td>13.5%</td>
<td>These papers investigate novel techniques that can be used to enhance and improve discovery process of Web services. Techniques could be implemented in experiments or written as prototypes.</td>
</tr>
<tr>
<td>Evaluation Research</td>
<td>3 papers</td>
<td>5.8%</td>
<td>These papers analyses and evaluates approaches, methodologies that are implemented in practice, it also shows the benefits and drawbacks of the implemented techniques.</td>
</tr>
<tr>
<td>Solution Proposal</td>
<td>15 papers</td>
<td>29%</td>
<td>These papers identify problems relating to Web service discovery and attempts to find a solution by providing a novel solution or extension of existing solution.</td>
</tr>
<tr>
<td>Position Paper</td>
<td>6 papers</td>
<td>11.5%</td>
<td>These papers state personal opinion of somebody on an identified existing problem. It gives initial idea for solution; the solution is neither implemented nor evaluated.</td>
</tr>
<tr>
<td>Experience Paper</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Most of the selected papers fall in the philosophical paper category; this shows a sharp increase of the number scholarly articles in the recent years and level of maturity of the research area. Due to this academic researchers attempt to classify the existing literature by structuring them in the form of a taxonomy or conceptual framework. The major part of the selected papers were solution papers, this in turn clearly illustrates that different academic and research folks to a great extent trying to find possible method to ease the process of discovery and composing Web service discovery mechanism.
The following figure 4-1, presents the number of selected papers in each category and corresponding percentage, where the blue column or pillar representing the number of papers where red column presents percentage of each category.

![Figure 4-1: Paper Types.](image)

4.1 Motivations and Contributions

In the previous section, paper types were classified according to [47], however, in order to comprehend the driving factors of the research, the contribution of the papers from the main study were identified and once again categorized through five categories depending on whether the paper contributed to a process, methodology, modeling, tool and an area of overview.

The following table 4-2 gives broad and detailed description of the above mentioned categories according to .
Table 4-2: Paper Contributions.

<table>
<thead>
<tr>
<th>Contribution</th>
<th>Number of Cases</th>
<th>Percentage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Process</strong></td>
<td>11</td>
<td>21%</td>
<td>These papers generally propose an algorithm to solve discovery mechanism, especially dealing with the selection of functional and non-functional aspect of Web service discovery mechanism. These might involve also algorithms designed to deal with architectural problems.</td>
</tr>
<tr>
<td><strong>Methodology</strong></td>
<td>10</td>
<td>19%</td>
<td>In this methodology section, papers proposing rules and guidelines of Web service discovery mechanism fall in this category. Methodology describes rules and guidelines of how things should be performed. For instance methods of Web service discovery mechanism using ontology of semantic Web services are discussed in these papers.</td>
</tr>
<tr>
<td><strong>Modeling</strong></td>
<td>16</td>
<td>30%</td>
<td>Papers fall in this category, provide a solution to the Web service discovery process, it also could be papers proposing models of Web service discovery frameworks, papers proposing a novel or significant extension of earlier proposal or solution.</td>
</tr>
<tr>
<td><strong>Tool</strong></td>
<td>5</td>
<td>9.6%</td>
<td>Papers belonging into this category are papers providing software tooling support for Web service discovery, papers discussing search engines, portals etc.</td>
</tr>
<tr>
<td><strong>Area of Overview</strong></td>
<td>10</td>
<td>19%</td>
<td>In this category papers that compare literature regarding such survey, review and comparative study belongs to this category. Some of the papers listed in this category provide an overview of existing approaches, some of them give greater and extensive analyze of characteristics of different approaches proposed for Web service discovery mechanism.</td>
</tr>
</tbody>
</table>

As can be seen from the table 4-2, the distribution of the papers in terms of their contribution, on the process category about 21% of the selected papers falls into this category. Papers [1, 9, 10, 19, 22, 24, 29, 36, 38, and 41] brought forward process related solutions that provides an algorithm that facilitates Web service discovery. These papers focus on a specific area related to the Web service discovery process such as ranking Web services according to QOS parameters. It mainly approaches that presented decisions to be made when selecting appropriate Web service. Additionally the
common aspects within these papers are, authors of these papers emphasized the importance of future work to be done on the topic to include more elements.

On the other hand methodology classified category encompasses papers[2,15,28,30,35,37,39,49,50,52] which represent 19% of the selected papers. These papers provide numerous approaches tackling Web service discovery mechanism including automating discovery process, methods to enhance Web service discovery utilizing different techniques such as reuse service, quality of service, user data handling as well as set of rules and guidelines that facilitate of Web service discovery mechanism.

Papers were categorized under modelings which represent 30% of the selected papers, provide frameworks for modeling Web service discovery mechanism including multitude semantic web service discovery mechanism these papers include [1,5,11,13,23,14,17,24,25,26,40,48]. Most of these papers present a combined solution to Web service discovery problems taking into account dimensions such architectural, functional and non-functional aspects of Web service discovery problems.

In the tool category, the papers that discusses the tools that supports discovering Web services fall in this category, Although the number of these papers are only five papers and represent only around 10% of the papers, still provide significant and important applications that enables users to find or access Web services. These papers encompasses of [4, 7, 8, 12 and 27]. Finally the overall category represents 19% of the selected papers; these papers include all survey and other review literatures about Web service discovery. These papers are exhaustive review summarising the results, observations and finding of the researchers in the domain of Web service discovery. These papers that fall in this category include [3, 32,42,43,44,46,51 and 52]. Generally these papers put forward comparative analysis regarding different methods and approaches presented in the literature. To demonstrate graphically the contribution of the paper, the below figure clearly illustrates paper distribution in different categories, where first column represent the number of paper in each section with the percentage of each category corresponds to over the whole selected papers. Modelling being the main category shows that researchers have been attempting to define the problem of web discovery issues, effects and causes to come up a solution, the provided solution could be a novel algorithm or architectural paradigm that eases finding appropriate Web services. This illustrates researchers’ attempt to model different solutions already proposed to find a novel or significant extension to earlier proposed approach or methods.
As mentioned in section 3.4.2 the selected papers were classified into two main categories namely functional and non-functional based methods, where each category has many subcategories. In this section the data extracted in those categories will be discussed thoroughly.

Prior to semantic web the Web service discovery used UDDI service registry which only supports keyword and category (taxonomy) based discovery mechanism, which is in turn too static in nature and returns a huge amount of irrelevant Web services as well as lack of scalability and incompatibility problems as the number of WS increases rapidly. Due to these obstacles semantic web has recently attracted lots of attention from both academic and industrial researchers. This trend can be observed in this work as papers [29, 30, 35, 37, 38, 39, 40, and 49] can be seen. All these papers have in common is each provides a framework that tackles problem related to semantic web service discovery. These frameworks can be classified in several broad categories in terms its contribution for instance [35] introduces a solution that describes a framework for ontology-based flexible discovery of Semantic Web services, the proposed...
approach relies on user-supplied, context-specific mapping from an user ontology to relevant domain ontology used to specify Web services. While [37] presents a framework that uses service’s reuse, quality, evolution and user’s data handling as well as scheme of discovery using user’s annotating information. On the other hand the paper [38] introduces web service discovery based on offer discovery that uses WSMO-Lite a new lightweight semantic annotation framework. Focusing on service offers provided by service providers, this framework is able to rank the returned Web services and select the appropriate one using offered parameters that relevant to the goal of the user. Similarly papers [29, 30 and 49] present an OWL based enhanced framework aiming at improving semantic web service discovery, for instance [29] introduces to a concept of dividing the semantic web into different domain using semantic web service router designed to help reduce the set of service description information, which service request information has to be matched. This approach still uses centralized registry but semantic web service router employed is used across the distributed network to reduce the set of service description of published semantic web services. However, paper [30] introduces three layer pervasive semantic service matching algorithm employing OWL-S. This method matches services according to category of service, input/output parameters and non-functional user required goals such as QOS. Additionally paper [49] proposes a framework relies on a lightweight and minimal description model allowing the service to describe a set of functional and non-functional properties. This approach divides matchmaking process into two fold, initially using deductive relaxation of discovery queries based on semantic service description to select as many services possible that meets user’s request, and then non functional properties are incorporated into the web service discovery mechanism to generate partially ordered list of services that meet user’s functional and non functional requirements.

All these papers focused on reducing service description information by dividing the domain and matching against a service using described capabilities.

As the number of researches in the field of semantic web services increased, some of researchers turn their attention to the automating service discovery process without human intervention. Two points have emphasized throughout the entire papers which are in order to become a successful semantic web services requires an automatic and scalable discovery mechanism, in addition to that, the distributed nature of semantic web services has to be considered when designing a semantic web services discovery mechanism. In the light of this, papers [11, 19, 20, 23, 25, 27 and 40] are proposing a new model to automate semantic web service discovery, for instance all these papers attempt to provide rich and machine-understandable representation of service properties, capabilities, and behavior as well as reasoning mechanism to support automation activities. Authors [19 and 20] introduce a novel approach to automate the discovery of Web services, the former utilizing DL–based ontologies and proposing a hyper graph based algorithm to effectively compute best covers of a given request. While latter [20] employs Web Service Modeling eXecution Environment (WSMX) with peer to peer (P2P) based discovery architecture to provide an automotive semantic discovery mechanism.

Some of the researchers even took further step by taking into consideration user context such as environment, location and preference to formulate an automotive semantic web service discovery mechanism. For illustration papers [23, 25
and 27] present an approach that takes into consideration user’s context, particularly paper [23] introduces to a framework by which both user’s and provider’s context are explored to help understand the real need of a user. An association rule for context modeling has been proposed from which ranking service is invoked to compare web services candidates with user requirements to select best suitable potential web service. Authors [11] present Context-aware service directory focusing on user mobility in the case of mobile web services, where both service requestor and provider might both be mobile. This approach is built upon traditional service directory where an algorithm is designed for construction, search, update and merge of service directories.

On the other hand, lots of efforts have been invested to provide best web service discovery mechanism considering web services infrastructure and architectural aspect of the system. Some of the researchers suggested peer to peer (P2P) based web service discovery mechanism instead of centralized UDDI registry which poses information stress problems such as performance bottlenecks and fault tolerance. In this context, papers [4, 14, 15, 16, 17, 18, 26 and 43] present an architecture discovery depending on what environment is implemented, basically semantic web services are distributed in nature thus P2P are recommended since it provides a scalable, self-organizing alternative to centralized approach. Moreover, others have promoted the idea of having an agent as a middleware, web crawler or search engine. Authors [14, 15, 16, 17 and 43] opted to structured P2P with clustering web services into communities based on functional properties, and then several query packets are passed among these clustered communities to identify their non-functional properties. In contrast to that authors [4, 18 and 26] preferred to use web service crawler which explores the web service accessible over the Internet and uses also WordNet Lexical database to gather synonyms of the functional semantic enriched request query. In [26] authors present a service broker which facilitates an effective Web service discovery and automatic publishing. The broker utilizes strength of functional semantics, keyword matching, structural syntactic matching and semantic. Commonly the agent based architectures make use of software agents for intelligent and quality driven web service discovery by assisting the discovery mechanism in processing the requests and finding Web service satisfying the desired quality.

Some academic and industrial research community have also explored various techniques and methods to ease and mitigate the process of Web service discovery, these methods include information retrieval approaches, incorporating non functional properties such QOS, performance and trustworthiness of the web service (Hansen et al, 2007). To this end papers [5, 7, 12, 24, 40 and 41] have emphasized the importance to take into account non-functional properties such as QOS characteristics.

For instance authors in paper [5] proposed Web service discovery model in which both the functional and non-functional requirements are taken into account. A mechanism is presented that carries quality information into either WSDL description or in a database of web service selection intermediary (Broker-like).Moreover it retrieves these quality characteristics from the database or WSDL description and process them, this in turn enhance the effectiveness and the quality of web service selection process and elevate the possibility of delivering desired WS. Additionally, papers [7, 12, 24, 40 and 40] each paper presents an approach to include QOS parameters to WS selection for instance paper [24] the authors proposed two phase web service discovery mechanism based on Quality of Service “QOS” and
collaborating filtering, which discovers and recommends the needed web service effectively for users in the distributed environment. With collaborative filtering the proposed approach is able to solve the problems of Web service incorrect QOS information. Similarly in the paper [41] a remarkable framework for QOS based web service contract has been proposed. This framework is for the automating of Web service contract specification and establishment by introducing a novel algorithm that enables to rank web service with equivalent functionalities on the basis of its ability to fulfill the service requestor requirements; in addition it provides an algorithm for the configuration of negotiable part of the QOS Service-Level-Agreement. All these works clearly illustrate the importance of nonfunctional properties play in exploring Web services for the requester which fulfill his expectations on nonfunctional properties. Due to this in literature Web services are discovered based on the level of QOS offered by the Web services. QOS parameters include reputation used to discover trustworthiness of Web services; another very important parameter also is usability which provides predicted value meaning how the advertisement is similar to the request, in formality. Thus, usability based discovery provides appropriate Web service to the users when there are no exactly matched Web service available for the service requestor.

In literature it has been observed that a number of fuzzy based discovery mechanism have been proposed, for instance papers [8, 13 and 28] present fuzzy based discovery approaches, in paper[8] authors mentioned a fuzzy web service discovery used to construct and represent Business Process; it helps in relaxing the matching criteria of Web services and allows service consumers to specify business requirements in a more fuzzy way and hence increases the possibility of finding required Web services that could construct Business Process. Likewise paper [13] proposes a moderated fuzzy Web services discovery method that models subjective and fuzzy opinions to assist service consumers and providers to reach consensus. This process is an iterative process such that further fuzzy opinion and preference can be added to improve precision of web service discovery. Another attention grabbing fuzzy discovery approach is introduced in paper [28] where authors attempting to close the gap between Web services using different ontologies, thus proposed framework termed FuzMod that applies fuzzy logic to the discovery of web services that use the same ontology as well as different ontologies.

Other researchers have devoted their effort to the Web service discovery mechanism based on information retrieval methods using Web crawler or search engines to locate WSDL documents and useful information spread over the Internet. Such discovery mechanism retrieve functional descriptions from WSDL documents using information retrieval based methods and adopt matchmaking mechanisms based on Latent Semantic Analysis (LSA) or Vector Space Model (VSM). In the studied literature, papers [1, 2, 9, 10 and12] authors attempted to put forward solutions to discovering process using classical information retrieval approaches. For instance papers [1, 2 and 10] propose a novel IR style mechanism for discovering and ranking web service automatically, given textual description of desired services.

All in all the selected papers clearly indicate the though the field of Web Service discovery is rather new, even newer than Web services themselves, much work has been devoted lately. Different solutions and proposals have been introduced ranging from considering network nodes as peers sharing Web services to ontology based semantic web
discovery as well as using classical traditional information retrieval approaches coupled with functional and non-functional properties to enhance quality of discovered Web services.

Regarding the research questions listed in the section 1.1, observing findings of the systematic mapping work clearly indicates that there are numerous approaches that are consistently being researched and published in this field. In terms of where these researches are published, conferences are major contributing publishing arena as 48% of the selected papers were conference published papers. As can be seen from Figure3-4, the number of published papers on this field will proportionally grow as the interest of Web service discovery mechanism is growing and number of academic and industrial researchers is even attempting to automate the entire process of finding and selection appropriate desired Web services. Additionally it has been observed during systematic mapping process that the majority of the papers were categorized under philosophical papers as can be seen in Table 4-2, this clearly shows the maturity level of the research area as this categories encompasses frameworks and surveys that attempts to combine previous related works done on the research area with the motivation to create efficient Web Service discovery mechanism. Moreover, the motivating forces behind these approaches were combining functional and non-functional properties of WS in the process of finding suitable and appropriate WS as well as the need for automation and for on-demand adaptation during run-time and within an environment that is constantly changing. In addition to the above mentioned facts, lot of surveys were part of selected papers, which in turn point to the level of maturity of the research area as 17% of the selected papers were Surveys some of the survey reviews were devoted to the literature of QoS based Web service discovery, while others focused on context awareness literature, further more architecture aspect of Web Service discovery have been explored.

All in all authors concluded predominantly with indicating the usefulness of their innovative solutions and approaches, however, the emerging pattern from the conclusion section of the most of the selected papers was that additional work should be done in addition to their presented research works. This included better ways to explore how to further automate the process of finding, selecting and composition of Web Services, while others suggested further extensive experimental study to measure effectiveness of their approaches for instance paper [52] a case study will be carried out encompassing larger number of context scenarios to better evaluate the benefit of their approach. Others suggested qualitative evaluation of autonomic maturity will be needed particularly research in autonomic service management, business-driven automated composition and dynamic connectivity capabilities are required. Although different studies focus on different dimension of Web service discovery but all the authors have in common is importance of effective Web service descriptions and well defined service request including intended functionality in natural form containing all necessary participating domain objects and object qualities along with action to be performed on them. A combat structure is also required to represent operations of all advertised Web services to keep Web service information in memory for the speedy discovery process. The solution which caters to the above requirements will enhance the effectiveness of Web service discovery.
5 Conclusions

Web services are essential for business to business communication and Internet users at large, it resolves all problems of service consumers through interoperability. Web service discovery mechanism plays very important role in the lifecycle of web service composition, selection and discovery and needs to be optimized and upgraded. In this thesis literature on the field of Web service discovery has been studied, examined and analyzed through systematic mapping study. The main motivation for the work reported in this paper was to investigate the state of the art Web services discovery mechanisms and systematically map the literature in order to determine challenges associated with it and how these issues are addressed by researchers. Through search, selection and exclusion approach 52 papers were selected; data extraction strategy has been utilized. Many approaches differ in the way web service matching is carried out. Some approaches are considering concept of semantic web, while some other focus on information retrieval methods. Some approaches suggest enhancement in web service request based on metadata about web services generated by feedback of other users. Some approaches suggest additional tools in traditional framework of web service discovery. Minimizing total search area using clustering techniques is also suggested. Survey shows that considering QoS parameters while selecting is important because, number of available web services providing same kind of functionality is very large. As web service discovery requiring manual interference may take more time, solutions for automatic discovery are drawing more attention.
Reference


URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5386918&isnumber=5386912


BIBLIOGRAPHIC DETAILS OF THE 52 ASSESSED PAPERS


[24] Lin et al. ” A Trustworthy QoS-based Mechanism for Web Service Discovery Based on Collaborative Filtering” Ubiquitous and Future Networks (ICUFN), 2013 Fifth International Conference on 2-5 July 2013, Page(s): 71 – 76.


[38] Pilioura et al. ” Unified Publication and Discovery of Semantic Web Services” Journal ACM Transactions on the Web (TWEB), Volume 3 Issue 3, June 2009, Article No.11.
[45] Ngan et al. ” Semantic Web service discovery: state-of-the-art and research challenges” Personal and Ubiquitous Computing ISSN: 1617-4909 (Print) 1617-4917 (Online).


