CHAPTER 3
GOALS AND RESEARCH METHODOLOGY

3.1 INTRODUCTION

“Webservices based Systems promise to bring significant business value out of existing IT assets through increased operational efficiencies, optimized business processes, and the ability to adapt and change quickly,” (Utah CIO Stephen Fletcher 2008). Platform and language independent and flexible access to information is indeed a complex task and resource intensive too. Web Service based Systems simplify this through standard protocols which treat all platforms equally. Through such systems, it is possible to offer data Services to a wide variety of business partners and the requests can originate from anywhere. The terms, Enterprise Architecture (EA), Services Oriented Enterprise (SOE), Service-Oriented Architecture (SOA) and Service Oriented Computing (SOC) are being exposed in connection with the Web Service based Systems to an ever wider and more influential audience.

Web Service based Systems promise to be a significant innovation that will provide the ability to pick and choose business and technology services, and will allow the trade out of Services based on organizational re-design, new strategic intent, legislative requirements, or business process modifications (USA National Association of State Chief Information Officers (NASCIO) 2011).
From its inception Web Service based Systems have been a lightning rod for dissension among enterprise architects, solution architects and application architects. Enterprise architects view such systems as a business initiative that should be guiding how information technology assets receive investment and how they relate to the business’ goals and mission. Solution architects view such systems as a means to deliver solutions faster using the tenets of loose-coupling and finer-grained Services, which enable faster construction. Finally, application architects see Web Service based Systems as an infrastructure to deliver applications based on Service interfaces (JP Morgenthal 2011).

Having seen the wider adaptability and acceptance of Web Service based Systems by enterprises, it is widely accepted that they offer the following advantages over traditional approaches to distributed computing (Jagdish Chatarji 2004).

- Web Service based Systems offer business Services across the platforms and thereby the application architects can conveniently choose a particular Service for any kind of platforms on which the system need to be developed.
- Location independence is considered to be a major advantage through which the application developer can make use of the Web Services transported from anywhere to anywhere.
• That the Services need not be at a particular system or particular network is the greatest benefit of using Web Services for application development. Such benefit really brings out the developer from the pressure of locating all necessary Services from a single location and use.

• Web Service Systems are built on a completely loosely coupled approach which brings in high level of modularity and the developer need not re-architect the entire application for a minor change in a particular section of the application.

• Web Service based Systems also offer the advantage of Authentication and Authorization support at every level of the application building, which makes the entire system highly secured.

• The search and connectivity to other Services is dynamic, which enables the system developer find the Services which are just fit for the applications being developed and the connectivity which extends the usage of a particular Service by getting it coupled with another Service.

**Short-term benefits of implementation:** Web Service based Systems have been the methodology by the application developers for its short term benefits like enhanced reliability, less hardware acquisition costs, leveraging the existing development skills, accelerated movement to standards-based server and application
consolidation and the availability of a data bridge between incompatible technologies.

**Long-term benefits of implementation:** The Web Service based Systems also offer the ability to build composite applications, self-healing infrastructure that reduces management costs, truly real-time decision-making applications and possibility of the compilation of a unified taxonomy of information across an enterprise and its customers and partners.

**Benefits from the perspective of Business Value:** From the perspective of the business value, the Web Service based Systems provide the benefits of the ability to more quickly meet customer demands, lower costs associated with the acquisition and maintenance of technology, management of business functionality closer to the business units, leveraging the existing investments in technology, and reduced reliance on expensive custom development.

The field of Web Service based Systems and Service Oriented Architecture are the current topic of research and development in Computer Science, which is especially oriented to the needs of gaining and processing information in large-scale distributed systems such as the Internet. It is expected that Web Services will be employed to a higher extent for the realization of new kinds of application systems.

Service Oriented Architecture (SOA) is an architectural style for building Web Service based Systems that uses Web Services available in a network such as
the web. It promotes loose coupling between software components so that they can be reused. A Web Service is an implementation of a well-defined business functionality and such a service can then be consumed by clients in different applications or business processes (Qusay H. Mahmoud 2005). Web Services are:

- **Software components with well-defined interfaces are implementation-independent.** An important aspect of SOA is the separation of the Service interface (the what) from its implementation (the how). Such Services are consumed by clients that are not concerned with how these Services will execute their requests.

- **Self-contained pieces of code which can perform predetermined tasks and loosely coupled, which offer the benefit of independence.**

- **Web Services can be dynamically discovered from the Service Provider and if need be, the Web Services can be combined and the Composite Services can be built from the aggregates of other Services.**

The Universal Description, Discovery, and Integration (UDDI) specification defines a way to publish and discover information about Web Services. UDDI is a public registry designed to house information about businesses and their services in a structured way. Through UDDI, one can publish and discover information about a business and its Web Services. This data can be classified using standard taxonomies so that information can be found based on categorization. Most importantly, UDDI
contains information about the technical interfaces of a business's Service (Karsten Januszewski, Microsoft Corporation).

Web Service based Systems have stretched the Universal Description, Discovery and Integration (UDDI) Web Services standard to the limit, and it's time for a new standard (Joe McKendrick, IBM, 2007). IBM says that the UDDI standard for registries isn’t cutting it, and the time is now for a new registry standard more focused on today’s service based systems realities. "Our clients are telling us that they have an integration pain point," (Andrew Hatley, a manager at IBM's Software Group) "We need to create a new standard and the time is now". Colin Atkinson (2009) says that so far all attempts to set up publicly accessible UDDI-based Service brokers have not been very successful.

For these reasons, Semantic Web Service Discovery has been the chosen area of elaborate study and intensive research by many a research work like this one. Discovery of Semantic Web Services are interesting to users only if those architectures address their issues of interest. After the reviews on Web Service based Systems and Service Oriented Architecture, the following concerns have been identified. These are expected to be the concerns of different technologies and are the basis and motivation for this thesis.

- The available connection information is just not enough in Web Service registries of federated organizations; it should be able to share more about the Service requester’s connection information to make better predictions (Chen Zhou 2004).
- Web Service Server Architecture should be enhanced from its current state and it should support automated Web Services Registration, Availability and Quality Assessment (Witold Abramowicz, BIS 2009 International Workshop).

- Maintaining data integrity should be the top prioritized concern and the Web Service registries should not have any weakness in maintaining the data integrity of Services once they have been registered with the system (Colin Atkinson 2009).

- The Web Services’ information on an UDDI registry is to be updated as and when the changes are done by the Web Services Providers and hence the UDDI registries are expected to keep the information for registered Services up-to-date (Colin Atkinson 2009).

- For a Web Service registry to be successful as a resource recovery Service, it must be able to respond to requests from the Service Consumers in the least possible time with fewer message passes across the network.

- A Web Service registry must ensure high availability of the registered information to its users and the available information must be consistent across all the registries which are maintaining information about the services.

The concerns of an effective Semantic Web Service Discovery are well laid out and that is the reason why the aim of this thesis is to deliver an effective model
and an error free approach to provide Services information which is consistent. Due care has been taken to see that consistent information reaches the users in minimum response time.

A service lattice reveals the invisible relation between the services, easing the discovery and identifying the possible alternatives. The major drawback of the previous work is that the semantic information is not considered during clustering of web services. The present research considers lattices as they are easy to represent and also provide the hidden relation between the web services that are present. An attempt has also been made to intake the semantic information and produce high performance in the web services.

This research proposes a similarity measure that calculates the semantic similarity of the operations provided by the web services. The proposed similarity measure considers the harmonic mean of WordNet and Normalized Google Distance for calculating the similarity between the operations.
3.2 AIM, GOALS AND OBJECTIVES

3.2.1 Introduction

The primary aim of web service is to facilitate interactions between the software systems using XML standards. The architecture of web services is also designed in such a way that it employs XML standards to define and describe web services. This architecture also enables some of the web services functionalities such as discovery and composition to be done automatically to a certain extent.

3.2.2 Aim

The main aim of this research is to propose a framework to semantically cluster and discover the Web Services using QoS based User Preferential Model. The motivation has been the inherent shortfalls of the existing models of webservice discovery in the Web Service Computing environment.

3.2.3 Goals

The goals of this research are set so as to formulate a Semantic Service Discovery Model (SSDM) using User Preference Modelling (UPM) approach to improve the performance in terms of various QoS factors corresponding to the web service requested in the UDDI. So, in principle, the goals of the proposed research have two folds as follows:

i. To design a Web Service Clustering Model that enables time effective and User Preferential Discovery of Web Services.
ii. To develop a framework to discover the Web Services semantically from the Cluster developed based on the QoS Properties.

The First goal is defined so as to develop a web service clustering model based on the similarity measure developed, and the second goal is defined so as to design an effective Semantic Web Service Discovery model using the Clustering and the User Preferential approach for improving the efficiency of the service retrieval operation. All the goals are measurable and of course proved with appropriate set of experiments.

3.2.4 Objectives

In the view of achieving the goals defined, the research is organized into several phases and the major objectives are listed as follows:

i. To design a Web Service Clustering Model for appropriate segregation of Web Services using Lattice Concept.

ii. To formulate a User Preferential Model (UPM) to bridge the Semantic Gap between the Service Request and the Service Discovered.

iii. To build a Semantic Service Discovery Model (SSDM) based on Clustering and UPM Models.

iv. To develop a Test Bed to validate the performance of the proposed models using suitable Performance Assessment Factors.
3.3 RESEARCH METHODOLOGY

This section first discusses the proposed Web Service Cluster Model wherein the various similarity measures and the formal concept analyses are outlined one after the other. After the detailed discussion on the proposed Web Service Cluster Model, the Concept Lattice Model, Semantic Web Service Discovery Model (SSDM) and User Preferential Model are discussed before presenting the Overall Experimentation Framework. Each layer of the proposed experimental framework is thoroughly presented to set the base for further discussions on the experimentation and performance criteria evaluation, both theoretical and experimental.

3.3.1 Web Service Computing Model

The primary aim of web service is to facilitate interactions between the software systems using XML standards. The architecture of web services is also designed in such a way that it employs XML standards to define and describe web services. This architecture also enables some of the web service functionalities such as discovery and composition to be done automatically to a certain extent. Figure 3.1 shows the classical web services computational model, which clearly indicates that there are three major elements such as service provider, service registry and a service requestor with three major operations such as publish, find and bind.
Figure 3.1 Web Service Computing Model

The elements of the computing model can be elaborated by outlining the operations and the components shown in Figure 3.2, which can be termed as a detailed view of the Web Service Computing Model.

Figure 3.2 An insight of Web Service Computing Model
**Definition 1:** The basic operations of Web Service Computing are defined as,

\[ BOP_{WSC} = \{F, B, P\} \]

where,

- \( BOP_{WSC} \) refers to the Basic Operations of Web Service Computing Model,
- \( F \) refers to the Find Operation and it can be defined as \( F = \{R_F, R_{NF}\} \), where
  - \( R_F \) refers to the Functional Requirements, and
  - \( R_{NF} \) refers to the Non-Functional Requirements.
- \( B \) refers to the Bind Operation and it can be defined as \( B = \{S_{REQ}, S_{RES}\} \), where
  - \( S_{REQ} \) refers to the SOAP Request, and
  - \( S_{RES} \) refers to the SOAP Response carrying XML data.
- \( P \) refers to the Publish Operation and it can be defined as \( P = \{N_{WS}, D_{WS}, W_{WS}\} \), where
  - \( N_{WS} \) refers to the Name of the Web Service,
  - \( D_{WS} \) refers to the Description of the Web Service which is optional, and
  - \( W_{WS} \) refers to the WSDL description of the web service.

The State Space Representation of the architecture in Figure 3.2 can be depicted as shown in Figure 3.3 as below.
3.3.1.1 Service Provider

The service provider is the actual owner of the web service from the business point of view. The web service is created by the service provider and it is given a uniform resource identifier, which enables the users to use the service. The service provider can also be termed as a platform that hosts the web services [3]. The Service Repository in the service provider is the actual location of the web services. The service repository has a unique internet protocol address using which the $S_{REQ}$ and $S_{RES}$ are made during bind operation referred to above. The service repository is accessed by the Web Services Interface in order to satisfy the request and provide response to the service requestor.
The *Account Manager* in the service provider creates an account with the service registry before publishing the web services. The web service descriptions are obtained from the web services in the form of Web Service Description Language (WSDL) files. The *UDDI Interface* component prepares the web service descriptions or models according to the service registry and *publishes* the service to the service registry. The service provider then tests the registry for the published service and its functional description before making the service available to the public.

The service provider also interfaces with the service requestor with a *web service interface* that binds the service requestor to the service provider. In other words, the service requestor uses the service provided by the service provider using the Web Services Interface module. A *Service Request Monitor* component present in the service provider monitors the usage of its service in the service requestor. The service provider and the service requestor are bound by terms which enable the service requestor to use the service provided by the service provider.

### 3.3.1.2 Service Registry

The *service registry* is the central registry of web services where the service providers publish their service which can be searched by the service requestors. This registry is often referred to as Universal Description, Discovery and Integration (UDDI). The *Business Account Manager* component manages the accounts that are created by the service providers. This component helps the service providers to
create an account before they publish their services. The core functionality of the
service registry is to maintain the registry of web services that are being registered
with it.

The *UDDI Business Registry* (UBR) is responsible for creating this registry
of web services. The service provider publishes the web service using the WSDL file
along with the functional descriptions which are parsed by the *Business Interface*
component and the service is registered with the UBR with its own description. Each
entry in the UBR has its own service description and specifications, which the
service requestors search for finding the appropriate web services. The *Business
Interface* component is responsible for registering the web service description
obtained from the UDDI Interface of the Service Provider to the UDDI Business
Registry of the Service Registry.

The *Search Interface* component in the service registry is accessed by the
service requestors in order to *find* their needed services. The search interface in turn
searches the UBR for the requested service descriptions and then returns the WSDL
descriptions of the matched service to the service requestor. The clustering in the
service registry is performed either by considering the available descriptions during
the time of publishing a service or by considering the semantic information of the
web service based on the operations provided by the web service.
3.3.1.3 Service Requestor

The service requestor may be the actual user of the web service or a broker that uses the web service to satisfy the request of its client. The service requestor can be viewed as a business that requires certain functions to be satisfied from the business point of view, and can be viewed as an application that is looking for initiating an interaction with a service from the application point of view. The service requestor is usually a web browser using which we communicate with the Internet. The service requestor can also be a standalone application or even another web service that requires a service [3].

The service requestor contacts the search interface in the service registry to locate the desired web services. A web service client is then created based on the WSDL specifications and the client is made to bind with the service provider. A Simple Object Access Protocol (SOAP) request is generated by the client to the web services interface in the service provider which contains the data requested in the form of XML.

The web service interface then accesses the web service to perform the necessary operation and returns the result in the form of SOAP response carrying XML data. In the case of using a broker to use the web services, the HTTP request / response is sent to the broker’s front end which can be written using HTML, JSP, and ASP.NET etc. The middleware then gets the request from the front end and makes it easy for the web services client to read.
3.3.2 Web Service Computing Model and Conceptual Stack Mapping

The service descriptions are in the form of WSDL documents, which enable the client to contact the web service in a manner prescribed by the specifications. A web service may have a self-describing nature due to the WSDL files, which is also based on XML. The service provider publishes the service to the service registry using the WSDL and its functional descriptions.

The service requestor searches the registry and in turn gets a WSDL file which specifies the way through which the web service can be used. The service publication layer describes the way to publish the web services to the service registry known as UDDI. The web services have a self-advertising character due to the presence of a central registry for service discovery by the clients. The UDDI has the service descriptions that are provided by the service providers and they are parsed during the service discovery.

The service flow layer of the stack facilitates the composition of web services into workflows and the representation of this aggregation of Web services as a higher-level web service. It can be used to identify the inherent relation between them. The mapping model can be viewed as how the roles and operations in the computing model are mapped with the collection of standards and protocols in the conceptual stack. Figure 3.4 shows the mapping model between the web service computing model and the conceptual stack.
Figure 3.4 Web Services Computing Model and Conceptual Stack Mapping

The following discussion clearly explains the mapping from one entity in the computing model to the other entity in the conceptual stack.

- The HTTP request and the response from the web browser to the service broker are mapped to the network layer of the conceptual stack. This mapping signifies that web services are often based on HTTP protocols.

- The web services bind operation in the computing model usually requires an XML based messaging layer from the conceptual stack known as SOAP and thus the SOAP request and responses are mapped to the XML based messaging layer of the conceptual stack.
• The mapping between bind operation and the XML based messaging layer also signifies the importance of XML in web services and SOAP, which can be used for exchanging information in a decentralized and distributed manner.

• The service repository of the computing model is mapped to the service publication layer of the conceptual stack as every service in the repository is published to the service registry.

• The service description layer of the conceptual stack is mapped to all the WSDL file specifications and service descriptions and specifications found in the computing model. The service descriptions through WSDL files are inevitable for the clients to get the information about the web services and the service providers use them to publish their web services. Thus the web services descriptions are used during publish, find and bind operations.

• The UDDI Interface of the computing model is also mapped with the service publication layer of the conceptual stack as this interface actually prepares the web services to get published in the service registry (UDDI).

• The Business Interface and the UDDI Business Registry (UBR) components of the computing model are also mapped to the service publication layer of the conceptual stack as the Business Interface component gets the services published in the service registry and the UBR has the services that are published in the service registry.
• The UBR of the computing model is mapped with the service discovery layer of the conceptual stack as the web services are discovered in the UBR, which maintains the registry of all the published web services.

• The find operation of the computing model is also mapped to the service discovery layer of the conceptual stack as we actually perform find operation during the discovery phase. The client or the broker performs the find operation during the discovery phase in order to search the required web services.

• The SOAP requests and responses in the computing model is mapped with the service flow layer of the conceptual stack as the service flow layer signifies the composition of web services and various SOAP request/response pairs are needed for composing web services.

Thus, the Web Services Computational Model is mapped with the Web Services Conceptual Stack and the mapping has been used to study some of the inevitable relationships between the operations and the technology enabling them.

Lattice based clustering is one such technique for grouping of web services. This research considers lattices as they are easy to represent and also provide the hidden relation between the web services that are present in the UDDI. In order to discover the relevant web services, a framework is proposed for web service discovery based on semantic lattice clustering thereby reducing the semantic gap.
between the service requested and the service provided. This enables the user to find the relevant services.

### 3.3.3 Research Methodology Framework

The layered view of the experimentation framework is illustrated in the Figure 3.5. It consists of five different layers; Base Layer, Functionality Layer, Performance Attributes Layer, Operations Layer and the Assessment Criteria Layer. The responsibilities of each layer are defined and described in the following sections.

![Diagram of the Experimentation Framework]

*Figure 3.5 Experimentation Framework*
3.3.3.1 Base Layer

The Base layer of the experimentation framework is the most important part of the present work. The current stage of Web Services Publish and Discovery model is largely regulated based on Centralized UDDI registries (Wenly Dong 2007). Although centralized registries can provide effective methods for Web Services discovery, they suffer from problems associated with centralized systems.

The main advantages of Peer to Peer (P2P) systems are their very high robustness and scalability due to inherent decentralization and the ability to utilize large amounts of resources available on peers connected to the UDDI network (Zakaria Maamar 2007). The P2P-based Distributed UDDI framework makes the decentralized system more scalable than traditional Web Service centralized UDDI systems by way of distributing the system function among few peer UDDI nodes and not focusing on only one UDDI server (Fu-zhen SUN, 2010).

3.3.3.2 Functionality Layer

This layer is responsible for addressing various functionalities that the UDDI is concerned with. The main functionalities of UDDI are Composition, Selection, Ranking, Discovery and Publication. Web Service publishing means registering a Web Service in the UDDI registry and making it available to the Service Consumers. This will also guarantee the transfer of Web Services description to the consumer, which will be useful for the consumer to learn the way to interact with that Web Service.
Web Service composition refers to a process of adaptively composing a set of available Web Services into a business process flow, according to predefined business requirements. Web Service Discovery deals with finding a set of services that corresponds to a predetermined user request while Web Service selection deals with choosing a service from a set of discovered services. Web Service ranking is the process of assigning rank to the discovered Web Services based on user requirements to enable easy service selection.

3.3.3.3 Performance Attributes Layer

This layer possesses the core characteristics of this research, whose components are the various performance QoS attributes of the Semantic Web Service Discovery which are managed in the UDDI system. The QoS attributes concerned with this layer are Reliability, Latency, Throughput, Availability, and Response Time. The QoS attributes of Semantic Web Service Discovery are,

- The **Response Time**, it is mainly based on the time duration between a service user sending a request and receiving the corresponding response. It is measured in millisecond.

- The **Availability**, it is the ratio of the number of successful invocations to total invocations. It is measured in percentage.

- The **Throughput**, it is the total number of invocations for a given period of time. It is measured in invocations per second.
The **Latency**, it is the time taken for the server to process a given request. It is measured in milliseconds.

The **Reliability**, it is the ratio of the number of error messages to total messages. It is measured in percentage.

Although all these attributes have been identified, due to time constraint, this research focuses only on select performance attributes which have the maximum effect on Semantic Web Service Discovery model, such as Response Time, Availability, Throughput, Latency and Reliability. The rest of the attributes are left as future scope for researchers to deliberate upon.

### 3.3.3.4 Operations Layer

This layer is responsible for different types of operations performed in the system. Clustering and Semantic Discovery are the two major operations identified for research on this layer. The Semantic Discovery of the web services are done keeping in mind the issues related to Latency, Availability, Reliability, Throughput and Response time of the Services’ information. Operations in this layer are performed in such a way that the system experiences an appreciable performance in terms of the performance attributes discussed in the performance attributes layer.

### 3.3.3.5 Assessment Criteria Layer

The Assessment Criteria Layer of the framework is proposed to take care of the responsibilities of ascertaining whether the proposed User Preferential Model for
Semantic Web Service Discovery framework functions and delivers the services as envisaged. It is proposed to assess the framework on the following basis: Request based, Workload based, Precision & Recall based and F-Measure based. Basically the assessments are done in the performance attributes layer to prove that the base layer of the proposed framework is able to effectively achieve the functions proposed in the functionality layer for processes identified in the operation layer. Various methods of assessments of the proposed system for the expected and proposed qualities are discussed below.

- Request based assessments are performed to verify the total number of requests with response time in the semantic clustering method. The proposed User Preferential Model for Semantic Web Service Discovery allows the highest performance when compared to the previous works.

- Workload based assessments are performed to verify the total number of User load with response time for semantic clustering method. It is expected to be as minimal as possible in the proposed User Preferential Model for Semantic Web Service Discovery compared to the traditional clustering model.

- Precision and Recall based assessments are performed to the precision values of the existing and the proposed methods and they clearly indicate that the proposed methods yield high precision and the value tends to be the ideal case. The recall values of the existing and the proposed methods are assessed and they clearly indicate that the proposed methods yield high recall and the value tends to the ideal case.
• F-Measures based assessments indicate the improved values of F-measure, which is a harmonic mean of precision and recall of the discovered services in the proposed method. The F-measure values of the existing and the proposed methods clearly indicate that the proposed methods yield high F-measure and the value tends to the ideal case.

Based on the above discussions on the different methods to be employed for the assessment of the proposed system, it is proposed to demonstrate the better performance of the proposed User Preferential Model for Semantic Web Service Discovery.

3.3.4 Similarity Measures

A novel similarity measure is proposed, which calculates the semantic similarity of the operations for discovering the web services. The proposed similarity measure considers the harmonic mean of WordNet [25] and Normalized Google Distance [26] for calculating the similarity between the operations. If \( \text{op}_{ij} \) and \( \text{op}_{kj} \) are the two \( j^{th} \) operation provided by the two distinct web services \( \text{ws}_i \) and \( \text{ws}_k \), then the semantic similarity can be calculated as \( \text{SemSim}(\text{op}_{ij}, \text{op}_{kj}) \) as shown in equation (3.1),

\[
\text{SemSim}(\text{op}_{ij}, \text{op}_{kj}) = \frac{2 \times \text{WSim}(\text{op}_{ij}, \text{op}_{kj}) \times \text{NGD}(\text{op}_{ij}, \text{op}_{kj})}{\text{WSim}(\text{op}_{ij}, \text{op}_{kj}) + \text{NGD}(\text{op}_{ij}, \text{op}_{kj})} \quad \text{..(3.1)}
\]

where,

• \( \text{WSim}(x,y) \) is the WordNet similarity score of the two words \( x \) and \( y \).
• NGD(x,y) is the Normalized Google Distance between the words x and y.

This research considers both the WordNet and Normalized Google Distance as the combined score would be more appropriate for measuring the relatedness of the web service operations. The present research has chosen these two similarity measures since they measure the semantic similarity between the words as assessed by Google Search Engine and WordNet dictionary.

3.4 SUMMARY

Web service based systems have considerable advantages and they hold a lot of promise. Apart from being inexpensive and reliable they have a self-healing infrastructure that reduces management costs, truly real-time decision-making applications and possibility of the compilation of a unified taxonomy of information. The field of Web Service based Systems and Service Oriented Architecture are the current topic of research and development in Computer Science, which is especially oriented to the needs of gaining and processing information in large-scale distributed systems such as the Internet.

From the perspective of the business value, the Web Service based Systems provide the benefits of the ability to more quickly meet customer demands, and to lower costs associated with the acquisition and maintenance of technology. Earlier research never took into account the semantic information during clustering of web
The main aim of the proposed research is to propose a framework to semantically cluster and discover the Web Services using QoS based User Preferential Model. A similarity measure that calculates the semantic similarity of the operations provided by the web services is proposed. The proposed similarity measure considers the harmonic mean of WordNet and Normalized Google Distance for calculating the similarity between the operations.

The primary goal of this research to design a Web Service Clustering Model that enables time effective and User Preferential Discovery of Web Services. It is also intended to develop a framework to discover the Web Services semantically from the Cluster developed based on the QoS Properties.

This chapter outlines the proposed Web Service Cluster Model wherein the various similarity measures and the formal concept analysis are discussed. After the detailed discussion on the proposed Web Service Cluster Model, the Concept Lattice Model, Semantic Web Service Discovery Model (SSDM) and User Preferential Model are discussed before presenting the Overall Experimentation Framework. Consequently, the mapping model between the web service computing model and the conceptual stack is brought forth. The layered view of the experimentation framework has been elaborately illustrated, defined and described for clarity and logical progress of the research.