CHAPTER 4

EMPIRICAL RESEARCH APPROACH

In this part of the thesis, detailed introduction to empirical research in software engineering, various types of empirical studies, validity of empirical data and how this work deploys empirical research with the objective of performance analysis of web services in both implementation and selection are discussed.

4.1 EMPIRICAL RESEARCH IN SOFTWARE ENGINEERING

Empirical research plays a vital role in software engineering because software engineering deals with solutions for real-time problems and business applications. Basili (1996) pointed out that software engineering is a laboratory science and it involves an experimental component to test or disprove theories, to explore new domains. The possible objects for empirical study may be process models, products, techniques, organization and measures of software.

Empirical studies are performed to explore strengths and weaknesses of existing software engineering methods, techniques and tools (Bernd Freimut et al 2002). Experimental software engineering is the sub-discipline of empirical software engineering that uses the experiment to validate, improve, and select software engineering techniques that permit an efficient application development (Zendler 2001). Van keeken (2005) defines that empirical research is the process of collecting data regarding events
happening in an experimental or real life setting with the purpose to conclude something about an earlier formulated hypothesis. Experimentation is required

- to see how and when really work
- to understand their limits
- to improve them.

Empirical research, its importance and guidelines were discussed in some research papers. Kitchenham et al (2002) identified six basic topic areas in guidelines for empirical research as listed below.

- Experimental context
- Experimental design
- Conduct of the experiment and data collection
- Analysis
- Presentation of results
- Interpretation of results

Jeffery and berry (1993) propose a framework for evaluating the success of metrics programs. They consider 4 aspects of metrics: context, inputs, process and product.

4.1.1 Classification of Empirical Studies

Bernd Freimut et al (2002) stated that there are two types of research paradigms in empirical research.
Qualitative research - concerned with studying objects in their natural setting

Quantitative research - concerned with quantifying a relationship or to compare two or more groups with the aim to identify cause-effect relationships

Bernd Freimut et al (2002) divide empirical research into three methods as formal experiments, case studies and surveys which are defined as follows.

Experiment - A detailed and formal investigation that is performed in controlled conditions.

Case study - A detailed investigation of a single case or a number of related cases.

Survey - A broad investigation where information is collected in a standardized form from a group of people or projects.

![Diagram of empirical research strategies]

**Figure 4.1** Combination of empirical research strategies
Wohlin et al (2000) illustrated the use of strategies for empirical research for different situations as shown in Figure 4.1.

Zelkowitz and Wallace (1998) identified 12 experimental models for validating new technologies as listed below.

Project monitoring, case studies, assertion, field studies, literature searches, legacy data, lessons learned, static analysis, replicated experiment, synthetic environment experiment, dynamic analysis, and simulation.

Kitchenham (1996) identified nine different study types as given below.

- quantitative experiment
- quantitative case study
- quantitative survey
- qualitative screening
- qualitative experiment
- qualitative case study
- qualitative survey
- qualitative effect analysis
- benchmarking

Basili (1996) categorizes as, in vivo experiments which are run at a development organization and in vitro experiments which are conducted in an isolated, controlled laboratory setting.
According to Robson (1993), the purposes of empirical studies can be divided into exploratory, descriptive, and explanatory. Exploratory studies aim at looking for insight in order to find out what is happening. Descriptive studies aim at describing profile of events, organizations, or situations. Explanatory studies aim at looking for explanations of a situation or problem, typically in the form of causal relationships. Empirical data can be classified as quantitative data and qualitative data. Quantitative data is expressed in the form of numbers and qualitative data is expressed in the form of words or pictures.

Empirical studies may be carried out on various objects in software engineering as shown in Figure 4.2 (Bernd Freimut et al 2002). The objects mainly include process, project, product and their associated activities as well as measures of entity.

![Figure 4.2 Possible objects for empirical studies](image)

**Figure 4.2 Possible objects for empirical studies**

Bernd Freimut et al (2002) present an empirical process describes the activities that are generally to be performed when conducting an empirical study to investigate an object and also defined the activities as detailed below.
1. **Study Definition**

   The objective of this step is to determine the goal of the study to be performed.

2. **Design**

   The objective of this step is to operationalize the study goal.

3. **Implementation**

   The objective of this step is to produce, collect, and prepare all the material that is required in order to conduct the empirical study according to the study plan.

4. **Execution**

   The objective of this step is to run the study according to the study plan and collect the required data.

5. **Analysis**

   The objective of this step is to analyze the collected data in order to answer the operationalized study goal.

6. **Package**

   The objective of this step is to report the study and its result so that external parties are able to understand the results and their contexts as well as replicate the study in a different context.
4.1.2 Threats to validity

Empirical studies are subject to validity threats and the list of threats are detailed below (Wohlin et al 2000), (Judd et al 1991), (Yin 1994), (Robson 1993).

1. Construct validity refers to the question, whether the deployed measures suitably reflect the constructs they represent.

2. Internal validity refers to the question whether observed relationships are due to a cause-effect relationship.

3. External validity refers to the question whether the findings of the case study can be generalized.

4. Reliability or experimental validity refers to the question whether the study can be repeated with the same results.

Van Keeken (2005) discussed two empirical research methods namely benchmarking and monitoring.

Benchmarking is an empirical research method in which one or more representative or standardized tests, called as benchmarks, which are run in trials to compare the characteristics of an object of study. In computer engineering, benchmarks are used to verify and compare performance of databases, CPU’s and various system and application software. Monitoring is an empirical research method to analyse the usage of tools or techniques and evaluate them in different aspects.

Empirical research exposes a lot of valid, latent and pragmatically useful information but it needs to be carried out by following the guidelines in each stage of experimentation. Based on the nature of problem and context, mixture of different strategies of empirical research may be applied.
4.2 EMPIRICAL RESEARCH METHODOLOGY IN THIS WORK

In this context, empirical research in this work followed guidelines as discussed below. Hardware configuration also plays a role in performance and so the systems are used with same configuration and all the experiments are carried out in the same set up so that valid comparison of performance of software may be carried out. The systems used for experimentation were kept isolated from other networks to avoid any interference. That is, the systems were interconnected as an independent network entity with a dedicated hub so that observed metrics data reflects real performance of software. The network analyser tool wireshark was used for network traffic load measurement. The response time and communication payload was measured not only between end to end and also between intermediate stages of the framework. The experiments were repeatedly performed for many times in the same environment to ensure the correctness of metrics data.

![Diagram](image)

Figure 4.3 4-tier architecture framework deployed for performance analysis of web services.
The given framework in Figure 4.3 shows how client and server interact with each other for implementing and utilizing web services. The reason for using application server is that mostly in business applications, specific business logic application software is deployed and it may use various web services or composite web services. The reason for database server is that web services are better suitable for business applications and all business applications nowadays involve huge amount of data, hence, for handling larger amount of data, database is the only choice.

The systems are interconnected through LAN because testing for performance through Internet involves a lot of external factors which may lead to incorrect performance metrics data. Though web services are actually utilized through Internet, considering the dynamic nature and best effort service working principle of Internet, testing of performance of software for web services was carried out in an isolated environment like LAN where external interference is none.

The structure may be rearranged in different configurations as given in Table 4.1. By this flexibility, various scenarios are possible and each may be useful in evaluation of different kinds of software deployed in implementation of web services. The operational scenario of framework in scenario 1 is that the client interacts with application server which has application developed using web services.

The application server, in turn, connects with web service server for a service and the service requests the database server for data. The response originates from database server and traverse in reverse to reach the client. The operational scenario in scenario 2 is that client directly connects with web server which may help in evaluating performance of binding and other characteristics of web services. In case of scenario 3, client directly contacts web services server and web service server connects with database server for
data. In scenario 4 clients connects with application server and application server connects with web service server. Based on problem or requirements, one or more combinations of these scenarios may be deployed for performance analysis of software in web services.

Table 4.1 Various scenarios possible for the framework.

<table>
<thead>
<tr>
<th>Sl. No</th>
<th>Scenario</th>
<th>Client</th>
<th>Application server</th>
<th>Web service server</th>
<th>Database server</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Scenario1 (4 tier)</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>Scenario2 (3 tier)</td>
<td>A</td>
<td>NA</td>
<td>A</td>
<td>NA</td>
</tr>
<tr>
<td>3</td>
<td>Scenario 3(3 tier)</td>
<td>A</td>
<td>NA</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>4</td>
<td>Scenario 4(3 tier)</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>NA</td>
</tr>
</tbody>
</table>

A – available  NA – Not available

4.3 WEB SERVICE PERFORMANCE AND WEB SERVICE SELECTION

Performance parameters play a vital role in selecting a web service for an application. The performance metrics chosen are so vital because they are the only way of expressing and understanding performance of systems. The two main performance metrics for this framework are response time and communication payload involved between different stages of realization and utilization of web services. The proposed metrics suite brings out performance in two dimensions of implementation that is time taken for accomplishing the task and size of data to be transmitted on wire for deployment. Reduction in time is to make a system to provide response faster in order to satisfy users. Reduction in communication payload is to reduce network traffic load in a network which also reduces congestion in a network to a certain extent.
4.3.1 Response Time

The ultimate aim of any software solution is to provide response as quickly as possible so that user gets response faster. Moreover business applications demand for faster reply in no time. This will help in management to take better decisions for better business performance. The response time is measured as time elapsed between request originated from client and response reaches back to the client.

This provides total time spent for visualizing response on user’s terminal. The time spent not only between end to end, also between different stages of implementation of web services may also be measured and analyzed. Such measurement data between the stages helps in understanding contribution of performance of each stage to the total performance of the system. This can be further useful for making constructive changes in terms of performance enhancement in every stage of web services and also this gives deep insight into working of system which is very much required for performance enhancement. The measurement of this metric is done with a network analyzer tool namely wireshark and as shown in Figure 4.4.

Figure 4.4 Response time measurement
Actual Response Time = \( tt - t_{\text{conn}} - t_g \)

\( tt \) - total time spent from origin of request till the acknowledgement

\( t_{\text{conn}} \) - time spent for connection establishment

\( t_g \) - time spent for delivering output in the client side GUI

ART covers effective time spent for service request and response takes place between servers.

### 4.3.2 Communication Payload

Another important metric is communication payload between the systems as part of implementing and utilizing web services. This metric is measured in terms of number of packets required for different activities for getting response with some data payload. Further this may be classified into different kinds based on type of packets like TCP, HTTP etc., and also based on type of tasks such as request, response, acknowledgement etc. This study helps in understanding the communication payload required for different software platforms and this may be used in enhancing technological way of handling the tasks. Though this metric gets lesser priority than response time, the importance lies in size of data movements through wire or air.

The reason is that web services are mostly used through Internet and so it involves considerable amount of network traffic and transmission mediums like wire or air, are available with some cost involved with it. Further, congestion is the major problem in today’s network including mobile networks due to numerous customers and services and this metric helps in looking for further directions to reduce traffic load and congestion also to a certain extent. This metric also helps in making a step into research direction of employing new techniques to reduce communication payload without compromising functionality.
4.4 RESEARCH WORKS DONE

This research work, with the objective of performance aware implementation and selection of web services, is carried out in the form of five different works as discussed below.

4.4.1 Analysis of Software Platforms for Implementation of Web Services

Web services play a vital role in the paradigms of distributed computing and grid computing. Web services can be implemented in different platforms, but the most commonly used platforms are .NET and J2EE as they provide a wide variety of tools for creation and integration of web services to any existing business application. The selection of appropriate platform for implementation is purely based on performance offered by the particular platform.

This work focuses on analyzing the performance of a generic model composed of three tiers for implementing and using web services. This generic model is implemented with four different combinations of .NET and J2EE in different tiers. The analysis of performance is carried out with a novel set of four performance metrics proposed and the metrics are based on time spent for actual response and network traffic load involved between the different tiers.

4.4.2 Service Traffic Load Measurement in Implementation of Web Services

In this Internet era, network has become a vital part of the solution. Thus most of the software solutions and its performance rely on networks. Web services rely on networks particularly Internet, for realizing them in different applications. Whenever network is involved in the system, network
traffic load measurement plays a role in determining performance of the system. Thus, this work emphasizes the need for network traffic load measurement in evaluating performance of implementation of web services.

4.4.3 Databases in Implementation of Web Services

In the current era of software engineering, web services are increasingly being deployed in business applications, due to its unique features such as flexibility, interoperability and other features. Most of the business applications involve extensive use of database operations for data management in back end. Further business applications demand very high level of performance from software solutions and it is a continual and never-ending process.

Thus this work focuses on measurement and analysis of performance metrics of database queries of various databases in implementation of web services. In this experimentation, web services are implemented in two popular and standard platforms and database queries are realized through all commercial and standard databases. Performance measurement is done by implementing a common sample application on each realization and using a pair of performance metrics, response time and packet count, that is, number of packets involved in communication between the layers of implementation.

This novel work summarizes the various performance aspects of query execution of databases, in web services, particularly web services with a basic set of database DML queries in the back end and concludes with results on optimum performance offered by database in execution of database queries in realization of web services.
4.4.4 Legacy Software Systems to Web Services

One of the vital reasons for reverse-engineering legacy software systems is to make it inter-operable. Moreover technological advancements and changes in usability also motivate reverse engineering to exploit new features and incorporate them in legacy software systems. In this context, web services are emerging as better solution for software systems due to its nature of interoperable, simple to implement and also they exploit boom in Internet infrastructure. This work proposes a model based approach that can be applied at macro-level to migrate from legacy software systems to web services. It is also proposed that software metrics observed during reverse engineering also helps in better design of web services.

4.4.5 Web Service Selection in Cloud Broker

Cloud computing deploys web services as base due to the fact that cloud computing means computing through Internet. Cloud computing has become so popular and it is rapidly growing in terms of number of cloud services and cloud service providers. Many cloud service providers and cloud services are available to user and so the challenge lies in choosing the right one which matches various SLA (Service Level Agreement) parameters of requirements in all aspects of cloud customers. Hence this work proposes SLA parameters ranking concept and tree based approach with the use of sparse matrix for recognizing SLA parameters ranking pattern of requirements, by making user to explicitly state the SLA requirements both in priority-wise and performance-wise. We propose a set of metrics to recognize priority and performance level expected by users and it can be used to suggest an appropriate one from cloud services from several providers.
In a nutshell, this work advocates the need for detailed study on web services performance and also empirically investigates the impact of software deployed for web services implementation in the performance of web services. Service traffic load involved in consuming web service was analysed with possible software platforms and databases. Further SLA (service level agreement) parameters ranking strategy is proposed for efficient web service selection in cloud broker. A set of software metrics proposed for analysing QoS parameters priority ranking data for using it for performance enhancement.