CHAPTER 8

LEGACY SOFTWARE SYSTEMS TO WEB SERVICES

8.1 INTRODUCTION

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 and evolving as solutions for software systems for business applications in terms of facilitating interactions between business to business and business to customers.

Web services are gaining significance due to its inherent features like interoperability, simple to implement and exploiting boom in internet infrastructure. Thus this work proposes a framework based strategy using .NET for effortless migration from legacy software systems to web services. Further this work also proposes that software metrics observed during the process of reverse engineering facilitate design of web services from legacy systems. In today’s computing era, interoperability is one of the key non-functional requirements of software systems. This is particularly due to tremendous and endless growth of internet and its infrastructure. Mostly legacy software systems do not inherently have this interoperability feature as part of its structure. Thus one of the reasons for reverse engineering legacy software systems is to make it interoperable. Another vital reason is that to
incorporate new features possible through recent advancements in technologies and exploit their advantages for users’ satisfaction to the fullest extent feasible in the system. In either case as for as legacy software systems are concerned reverse engineering is the first step to be executed.

Software reverse engineering is the process of analyzing a subject system to: identify the system’s components and their interrelationships: create representations of the system at a higher level of abstraction (Chikofsky and CrossII 1990). In today’s computing arena, web services are emerging as solution for software systems due to ubiquitous availability of internet. Web services are emerging as a promising technology for the development and deployment of E-applications and for the effective automation of inter-organizational interactions (Amar Bensaber et al 2008). A web service supports direct interactions with other software agents using XML-based messages exchanged via internet-based protocols (Daniel Austin et al 2002). By overcoming interoperability limitations, web services allow the integration of existing software systems by exploiting the pervasive infrastructure of the world wide web and offer a renewed opportunity to continue using/ reusing the business functions provided by legacy systems (Canfora et al 2006).

Software metrics is emerging as better information support for software project management and producing software effectively and efficiently at cost-effective manner. Software metrics information is very much helpful in reverse engineering process.

Based on the facts discussed above, this work proposes a novel approach which is NET based, model driven and exploits software metrics, for migration from legacy software systems to web services. Software
maintenance, re-engineering, and reuse involving large software systems is complex, costly and risky mainly because of the difficult and time-consuming task of program comprehension (Systa et al 2000). Hence this approach deals the software system at macro-level or high level abstraction of the legacy system. The key objective of this approach is to make an attempt to provide solution for migration process from legacy systems to web services with lesser effort and complexity. This work uses black box approach and so the process is easy to implement and improve it in future. This approach is suitable for software for any business application because the vital strength of web services is that they provide mechanism for seamless interaction between business-to-business and business-to-customer in business enterprises.

8.2 LITERATURE REVIEW

Several approaches have already been explored in the context of reverse engineering legacy systems and migrating into new systems. De Lucia et al (2009) focus on migration activities towards distributed architectures – web and service oriented architectures. Bisbal et al (1999) discuss different approaches commonly used to cope with LISs (Legacy Information System). An abstract, model-driven SOA migration planning approach was presented by Aboulsamh (2009). Wrapping methodology was proposed by Canfora et al (2006) to make interactive functionalities of legacy systems accessible as web services. Weijun Sun et al (2009) propose a model-driven reverse engineering approach to facilitate and raise the degree of automation for the composition of web services.


### 8.3 Proposed Approach

The approach proposed here initiates with simple mapping procedure applied to the legacy system to migrate to web services. This approach requires understanding the legacy system at higher level abstraction which may be possible from the design of the system itself. Legacy system modernization techniques may also be subdivided into white box and black families, where the first family is based on the source code knowledge, while the second one does not require the analysis of the code (Comella-Dorda et al 2000). Black box approach is suitable for the proposed solution using the framework shown in Figure 4.3.

As shown in Table 8.1, input and output is mapped to the client side GUI through forms and reports. Data involved in processing is mapped to the target system of file or database server. Processing logic is mapped to application server or web service server based on its inherent property of reuse level.
Table 8.1 Mapping deployed for migration

<table>
<thead>
<tr>
<th>Artefacts</th>
<th>Migration target system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input and output</td>
<td>Client side GUI</td>
</tr>
<tr>
<td>Business specific logic</td>
<td>Application server</td>
</tr>
<tr>
<td>Reusable Components or functionalities</td>
<td>Web service server</td>
</tr>
<tr>
<td>Data</td>
<td>File or database server</td>
</tr>
</tbody>
</table>

Three categories of solutions were suggested by Bisbal et al (2009), namely redevelopment, wrapping and migration. This approach attempts to incorporate all the solutions in varying degree of extent so that to provide effective solution. Further this approach can accommodate mixture of all these solutions.

Wrapping is a black box modernization technique that has been used effectively for migrating legacy systems towards new platforms, by encapsulating the original system with a software layer that exports a modern interface and hides the original one (Canfora et al 2006). The application of wrapping technique is very well suitable for this approach. In case of software support for web services, J2EE and .NET are the two platforms which provide better support for web services.

Further .NET compliant languages list include almost all standard languages like C, C++, COBOL, Pascal etc., and it is expanding to cover more and more languages. The key solution suggested here is keeping functionality in the same legacy system, but using .NET, it can be migrated to web service because .NET is on one hand it is compliant with most of the high level languages and on the other hand it has a lot of facilities to visualize web services effectively.
Hence modules in legacy systems may be transformed into web services with less effort. Today’s legacy systems are written in COBOL or C, while tomorrow’s legacy systems are written in C++ and Java (Systa et al 2000). As .NET is compliant with all familiar and standard high level languages, this approach will work for any system in any .NET compliant language.

The crux of this approach is it needs thorough understanding of the design of legacy system so that mapping to different appropriate targets is feasible and it involves use of tools or slight coding is to be done for making all the artifacts to work together in the form of web services. Though partitioning into several modules in such a way that they are suitable to this approach is initially difficult but it is better for software maintenance, usability and improvement.

8.4 SOFTWARE METRICS IN REVERSE ENGINEERING

Software metrics are useful in a forward software engineering environment and also they are vital in a reverse engineering one (Zhou et al 1999). The approach proposed in this work attempts to use software metrics for effective migration from legacy systems to web services. Though application of several kinds of software metrics is feasible, only some metrics are put into use in this approach.

8.4.1 Complexity Measures

Complexity measures are used to indicate how complex the reverse engineering of a piece of existing code in terms of its size, structure and data (Systa 1999). Based on complexity level, suitable decision may be taken in this approach. If complexity is very low, then redeveloping only that part may be attempted. In some part of the system, high complexity may also force
redevelopment due to its maintenance and other issues like reliability etc., The complexity metric plays a role here in deciding in partitioning, whether redeveloping it or not and mapping to suitable target system. High complexity in a system may even be hid by providing better interface through forms and keeping the functionality unchanged. Software refactoring may also be applied on legacy systems to reduce complexity of the system.

8.4.2 Response Time

The ultimate objective of any software system, apart from its functionality, is its performance particularly in terms of response time. The response time includes the time elapsed from the initiation of request from client and response reaches back to the source of request. The response time of legacy system may be compared with the new system.

To increase the performance that is reduce the response time, various alternatives in terms of placing functionality in application server or web server and placing data in different stages in this approach may be experimentally studied before settling in final solution. Web services with same functionality from different service providers may be used here so that to increase performance by considering Service Level Agreements (SLA) of them.

8.4.3 Network Traffic Load

Network traffic load is another measure particularly in web services as they are consumed through network. Network traffic load can be measured in different alternatives of the system by placing data in different stages of this approach. For example some data which rarely changes may even be placed in client side unless it is not a critical data. Even small functionalities may also be kept in client side. If particular data is often required then based
on its locality of reference some caching mechanism may also be tried. So the use of this metric is only to reduce the response time and also the cost of network traffic.

The metrics discussed above provides insight into the legacy system which will enable for effective migration process. In addition to this, enhancements in the system in terms of its performance are possible in the system. The metrics also helps in quantifying software quality after migration. With the help of metrics, comparative study in different perspectives may also be carried out between pre-migration and post-migration of the legacy system.

8.5 SUMMARY

In this approach, a model-based strategy with involvement of software metrics is proposed for migration from legacy systems to web services. The solution discussed in this approach provides a mixture of wrapping and redevelopment for migration. .NET is used as key platform here so that even for any new language, if it is .NET compliance, then this approach will work for any software system. Further this approach follows black box technique which makes the process easier and minimum source code knowledge is sufficient. In addition to the above, software metrics information gained from the legacy system also used in this approach. Software metrics information can play a vital role in reverse engineering and only few metrics were explored in this work. The usage of this approach and metrics is to be validated by experimenting on sample legacy software systems to make the process robust. The strategy for migration from legacy software systems to web services will work for any legacy system as long as the software used in legacy system is .NET compliance one.