A Model driven Approach for Migrating from Legacy Software Systems to Web Services

K.Velmurugan[1]
Assistant Professor

M.A. Maluk Mohamed[2]
Professor

1,2 Department of Computer Science and Engineering

Anjalai Ammal Mahalingam Engineering College
Thanjavur – 614403
+91-9952406932
ssg_velmurugan@mamce.org
tnkvel@yahoo.co.in

M.A.M. College of Engineering
Trichy - 621105
+91-9443386622
ssg_malukmd@mamce.org

ABSTRACT

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.

Categories and Subject Descriptors

D.2 [**Software Engineering**]: Distribution, Maintenance and Enhancement – *reverse engineering.*

General Terms

Design, Measurement, Experimentation

Keywords

Reverse engineering, web services, software metrics

1. INTRODUCTION

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. That is why; 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 [1]. In today’s computing arena, web services 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 [2]. The W3C web services architecture working group defines a web service as: “A software application identified by an URI, whose interfaces and bindings are capable of being defined, described and discovered as XML artifact. A web service supports direct interactions with other software agents using XML-based messages exchanged via internet-based protocols” [3]. 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 [4].

Software metrics is emerging as a better information support for software project management and producing software effectively and efficiently at cost-effective manner. A software metric is quantitative measure of the degree to which a system, component or process possesses a given attribute [5]. 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 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 [6]. Hence this approach deals the software system at macro-level or high level abstraction of the legacy system. This approach attempts to provide solution for migration process 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.

2. RELATED WORKS

Several approaches have already been explored in the context of reverse engineering legacy systems and migrating into new systems. [7] focuses on migration activities towards distributed architectures – web and service oriented architectures. The paper [8] discusses different approaches commonly used to cope with LISs(Legacy Information System). An abstract, model-driven SOA migration planning approach was presented in [9]. Wrapping methodology [4] is proposed to make interactive functionalities of legacy systems accessible as web services. [10] proposes a model-driven reverse engineering approach to facilitate and raise the degree of automation for the composition of web services. The work [11] describes a model-driven approach to facilitate the construction of OWL-S specifications. [12] propose a reverse engineering based approach to specify web service according to the Web Service Modeling Ontology (WSMO). [13] propose an approach to create service ontology with WSMO specification starting from WSDL(Web service Description Language) by using the reverse engineering techniques. The work [14] presents a reverse engineering tool to discover and generate web services automatically from relational databases. The work [15] contributes to an iterative reverse engineering approach for component-based software architectures. [6] show how software metrics can be applied in the context of understanding software system using a reverse engineering environment. Though many approaches have been developed for reverse engineering legacy systems, and particularly in migrating to SOA and web services, this work also propose an easy but effective approach for migration from legacy software systems to web services. Moreover this approach also involves use of software metrics information as part of the process.

3. PROPOSED MODEL-DRIVEN APPROACH

The approach proposed here is simple mapping procedure applied to the legacy system to migrate to web services. This approach requires understanding the legacy system at high level abstraction which can 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 [16]. Black box approach is suitable for the proposed solution using the model shown in figure 1. Black box approach is easy to apply and does not require in-depth knowledge of coding which is so tedious in legacy systems.

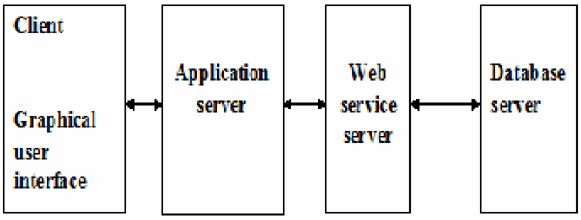

Figure 1. Model deployed for migration

As shown in the table 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 1. Mapping deployed for migration

Artefacts	Migration target system
Input and output	Client side GUI
Business specific logic	Application server
Reusable Components or functionalities	Web service server
Data	File or database server

Three categories of solutions were suggested in [8], 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 [4]. The application of wrapping technique is very well suitable for this approach. J2EE and .NET are the two platforms which provide better support for web services in terms of hosting web services. One of the most vital features of .net is that it allows program in multiple languages. Further .NET compliant languages list includes 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. Today's legacy systems are written in COBOL or C, while tomorrow's legacy systems are written in C++ and Java [17]. 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. Then modules in legacy systems may be mapped to the model appropriately and made as web services by providing suitable interface and wrapping with .net facility to host web services. Once legacy systems are migrated to web services, then through composition of web services, interplay of web services is feasible. It may result in flexible but efficient software systems after migrating from legacy software systems to web services. To summarize, the process involves understanding the design of legacy systems, modularization and then make it as web service through programming in multiple languages and web service hosting features of .net.

4. SOFTWARE METRICS

If software metrics are useful in a forward software engineering environment, they are vital in a reverse engineering one [18]. 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.

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 [17]. 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.

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 of this approach may be experimentally studied before settling in final solution.

4.3 Network traffic

Network traffic is another valid measure, particularly in web services, as they are consumed through network. Network traffic can be measured in different alternatives by placing data in different stages of this approach. For example some constant 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. Comparative study in different perspectives may also be carried out between pre-migration and post-migration of the legacy system.

5. CONCLUSIONS AND FUTURE WORKS

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 complaint, 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.

This work may further be extended in many ways. Experimentation of this approach in different legacy systems will bring out pros and cons of this approach. The experimental knowledge gained will be helpful in taking decision for new problems in legacy systems. The usage of software metrics may be expanded by covering more and more metrics in the system. The effect of the metrics in reverse engineering and reengineering may be studied further. The implementation of this approach in legacy systems in different languages may bring out a lot of language-oriented facts and techniques which strengthen this process.

6. REFERENCES

[1] E.J.Chikofsky and J.H.CrossII, Reverse Engineering and Design Recovery: A Taxonomy, IEEE Software, 7(1):13-17, Jan 1990.

[2] Amar Bensaber Djamel et al., "A reverse engineering approach for specifying Semantic Web Service with respect to MDA", Third International Conference on Information and communication Technologies: from theory to practice, ICTTA 2008, pp. 1-8, D.O.I 10.1109/ICTTA.2008.4530336

[3] Web Services architecture requirements, W3C Web Services Architecture Working Draft, Available online at : http://www.w3.org/TR/2002/WD-wsa-reqs-20021114.

[4] Gerardo Canfora et al., "Migrating Interactive Legacy Systems To Web Services", Proceedings of the Conference on Software Maintenance and Reengineering(CSMR'06), IEEE 2006, D.O.I 10.1109/CSMR.2006.34

[5] N.EFenton and S.L.Pfleeger, Software Metrics – A Rigorous Approach, International Thomson Computer Press, London, 2nd edition, 1996

[6] Tarja Systa et al., "Analyazing Java Software by Combining Metrics and Program Visualization", Proceedings of the fourth European software maintenance and reengineering, pp.199-208, D.O.I 10.1109/CSMR.2000.827328

[7] Andrea De Lucia et al., "METAMORPHOS: Methods and Tools for migrAting software systeMs towards web and service Oriented aRchitectures: experimental evaluation, usability and technology transfer", European Conference on Software Maintenance and Reengineering pp.301-304, D.O.I 10.1109/CSMR.2007.33

[8] Jesus Bisbal et al., "Legacy Information Systems: Issues and Directions", IEEE Software, September/October 2009, 103-109

[9] Mohammed A Aboulsamh, "Towards a model-driven approach for planning a standard-based migration of enterprise applications to SOA", 2009 Congress on Services – I, pp. 471-474, D.O.I 10.1109/SERVICES-I.2009.115

[10] Weijsun Sun et al., "A Model-driven Reverse Engineering Approach for Semantic Web Services Composition", World Congress on Software Engineering

PP.101-105, D.O.I 10.1109/WCSE.2009.403

[11] D.Amar Bensaber et al., "Ontology Development for Web Services: Reverse Engineering Approach", Second International conference on research challenges in information science, D.O.I 10.1109/RCIS.2008.4632135 pp. 433-440

[12] Houda EL BOUHISSI et al., "Reverse Engineering Existing Web Service Applications", 2009 16th Working Conference on Reverse Engineering, IEEE, 279-283. D.O.I 10.1109/WCRE.2009.35

[13] Houda EL BOUHISSI et al., "A reverse engineering approach for the web service modeling ontology specifications", The second international conference on sensor technologies and applications, 819-823, D.O.I 10.1109/SENSORCOMM.208.126

[14] Ricardo Perez-Castillo et al., "PRECISO: A Reverse Engineering Tool to Discover Web services from Relational Databases", 2009 16th Working Conference on Reverse Engineering, 309-310, D.O.I 10.1109/WCRE.2009.21

[15] Landry Chouambe et al., "Reverse Engineering Software-Models of Component-Based Systems", 93-102, 12TH European conference on software maintenance and reengineering, D.O.I 10.1109/CSMR.2008.4493304

[16] S.Cornella-Dorda et al., "A survey of black-box modernization approaches for information systems" Proceedings of the International Conference on Software Maintenance, IEEE CS Press, Los Alamitos, CA, 2000, pp. 173-183.

[17] T.Systa, " On the Relationships between Static and Dynamic Models in Reverse Engineering Java Software",

Sixth working conference on reverse engineering, pp.304-313, D.O.I 10.1109/WCRE.1999.806969

[18] Shikun Zhou et al., "A Useful Approach to Developing Reverse Engineering Metrics", The twenty-third computer software and applications conference, 1999 pp. 320-321. D.O.I 10.1109/CMPSAC.1999.812