IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 9, Issue 2 (Jan. - Feb. 2013), PP 32-35
www.iosrjournals.org

Survey and Analysis on Migration of Legacy System to Service Oriented Architecture

Krupi Patel[1], Leena Ragha[2]
[1](Department of Information Technology, R.A.I.T. Navi Mumbai, India)
[2](Department of Computer Science, R.A.I.T. Navi Mumbai, India)

Abstract : *A number of new generation languages have been evolved for the past few years and have been adapted by the industries. As a result of which the project done under 3GL are becoming Legacy and also the information that is to be stored is increasing the complexity of the Legacy systems. These are issues because of which companies are migrating their legacy systems to SOA which provides loosely coupled and interoperable services. There are many approaches to migrate a legacy system to SOA. This paper presents a survey to migrate legacy system to SOA. We also propose a roadmap to migrate a legacy system to service oriented architecture.*

Keywords - *Legacy system, Migration, Reengineering, Replacement, Service oriented architecture, Wrapping*

I. Introduction

A Legacy system being an archaic structural and phenomenal significant platform of technology; may it be a system, language or program developed by using the phased out languages & tools of programming; has moved out of the aisles of information technology in use & therefore is hard to maintain and inflexible but despite of the fact these are most important and have strong impact on the organization. As they support basic and complex processes of the organization they are continued to be used.

The Legacy systems legitimately have restrictions to archaic hardware, software and even the phased out know how of the people who possessed the skill set about that; as the succeeding generation is not adaptive to the passed technology; or doesn't prefer to learn it. Integration with newer systems may also be difficult because new software may use completely different technologies.

Service Oriented Architecture is relatively a new branch of Software engineering field. It can be analogically considered as a manual of predefined rules for migration of the legacy systems to the technology which extracts the basic functionality of the legacy system and bring them in the form of services without making substantial changes to the legacy systems. There are several features of SOA that make legacy system modernization appealing in today's world, including loose coupling, abstraction of underlying logic, agility, flexibility, reusability, autonomy, statelessness, discoverability and reduced costs[1].

Lack of interoperability in the backbone structure is the first and foremost basis of migration to SOA and trailing to it the inclusion of aspects that make the system compatible to better technology. Interoperability being the essence of the whole idea is backed up by the inclusion of the new technology to pull out an effective migration.

The rest of this paper is structured as follows. Several SOA approaches and techniques are described in section 2. Overview of Migration methods is described in section 3. A different method is proposed under section 4.The paper is concluded under section 5.

II. SOA Approaches and Techniques

Several sets of techniques have been defined for converting legacy system to SOA. The technique is sub divided into white box and black box families [2, 3].

The white box family is based on the knowledge of source code. The system which we want to migrate to SOA is analysed based on its source code from which the important modules classes and dependencies are extracted. And based in these analysis further steps are performed. The white box technique is subdivided into top-down development and bottom-up service extraction.

In the black box technique the deep knowledge of source code is not necessary as this technique does not deal with the source code in detail. Black box technique just maps the desired output to the exiting services. In all it just provides a new interface to existing application**.** Wrapping is a black box modernization technique That has carried legacy system to the new era technology without high talent manpower & skill set as in legacy systems. Black box approach is relatively easy to apply and does not require thorough knowledge of coding.
These methods show different aspects on SOA migration. They mainly differ in the way they provide solution for two challenging problems of what can be migrated i.e. the legacy element and how these migrations are performed i.e. the migration process

Several approaches have been defined for converting legacy system to SOA these are replacement, redevelopment or reengineering, wrapping and migration [1].

In Replacement we extract the legacy code from the scratch and try to rewrite the whole set of code in the new language or a new technology. Replacement of legacy system to refurbished system may be vital or fatal for the organisation but it provides us with the advantage of correcting or redefining the business processes.

Reengineering comprises of processes carried out e.g. reverse engineering, restructuring, redesigning, and re-implementing software being tool for analysis and adjustment of an application to present it in a new form on completely different platform.

Wrapping is a black box modernisation resource that is instrumental in giving a SOA interface to existing legacy components. Since it is concerned only with interface, it does not at all reveal the inherent complexity if the legacy component and is effective when the probability of replacement being fatal is more or it is too small for reengineering to apply.

Migration refers to an approach which moves the entire legacy system and its core framework to the new environment. In migration we use wrapping as well as redevelopment approach. In Migration first of all legacy code is identified then it is decoupled and extracted using wrapping and redevelopment approaches**.** Migration strategy is most widely used while migration of legacy system to SOA because it yields the most stable application. Lots of tools are available for migration of the legacy systems to new environment.

III. Overview of Migration Techniques

For the migration of legacy system to SOA a pathway is required. As a legacy systems contains hundreds lines of code out of which we have to extract the main functionality from which our SOA can be constructed. Several authors have proposed different methodologies which abstracted as follows.

Khadka et al.[4] have proposed consolidated legacy-to-SOA migration approach that facilitates the (semi)automated service identification and service extraction from the legacy system and the open-source system. Their method consist of three steps: consolidated to legacy migration approach, candidate service identification, service extraction. This approach consists of two aspects: migration feasibility and technological support.

Razavian et al.[2] has given a conceptual framework for understanding SOA migration. In which they have first of all illustrated the SMART and Sneed's approach working in top down and bottom up manner respectively. Then after that they have given a SOA Migration Framework. Assuming migration process as some kind of reengineering process and they have given the concept of horseshoe model which is generally accepted conceptual model for reengineering.

Matos et al.[5] migration approach is based on source code analysis for identifying the contribution of code fragments to architectural element and graph transformation for architectural migration. They also uses horseshoe to some extent in their approach. Actually the goal is achieved using technique such as code pattern matching and graph transformation. In which the pattern matching is used for technical purpose and graph transformation focuses on business level functionalities.

O'Brien [6] presents a case study showing how the architectural reconstruction was used on a system to support an organization's decision making process. The main consideration is given to the dependencies in the system. Different types of dependencies are identified and categorised as either functional dependencies or data dependencies also the relation between the elements which are highlighted as dependencies are also included. These relations or say dependencies are useful if the system is decomposed in a file structure and the subsystem and components can be mapped to directories.

Cetin et al.[7] propose a mashup migration strategy, addressing both the behavioural and the architectural aspects of the migration process. Their method consists of six steps: model the target enterprise business requirements, analyze the existing legacy system, map the target enterprise model to legacy components and identify services, design concrete mashup server architecture, define service level agreements, and implement and deploy services.

Cuadrado et al.[8] Performed a case study on Software evolution towards SOA aiming to address the issues of legacy system. His approach follows the white box approach. The process is three clearly defined activities: first, architectural recovery of legacy system is carried out, documenting it properly. Second, an evolution plan is defined detailing the candidate architecture and the necessary evolution step. Third, the plan is executed, that is, the system goes forward to SOA.

Table 1 summarizes the different techniques according to the Dimension it possess which can be functional dimension or technical dimension, type of legacy system used, number of steps followed to migrate the legacy system to SOA, analysis depth to show to what depth the system has been analysed, tool supported is any tool have been used and what is the finishing point of that technique.

Table 1: Summery of Migration Technique

Reference	Dimension	Legacy System	No of Steps	Analysis Depth	Tool Supported	Finish point
Khadka et al.	2	System Independent	3	Detail analysis of target SOA	Yes	3rd step under development
Razavian et al.	1	System Independent	3	Legacy system is integrated to target system	-	Completed with 2 case study
Matos et al.	2	Program Independent	4	Code fragment extracted and graph transformation done	Yes	Case study
O'Brien et al.	1	System Independent	2	Architectural reconstruction	Yes	Case study on C++
Cetin et al.	1	Web Application	6	Target system achieved	Yes	Case Study
Cuadrado et al.	1	Program Independent	3	Target system achieved	Yes	Case study

IV. Proposed Methodology

As the legacy system contains hard core of our business they are vital for organisation. But they are hard to maintain and consumes very 50%-70% of the business cost. Recently Service oriented architecture has emerged as a promising architecture style that enables existing legacy system to expose their functionality as service without making significant changes to legacy systems.

The basic steps which are performed in order to migrate a legacy system to SOA is to identify the legacy code, then it is decoupled and extracted, using wrapping or redevelopment these legacy code are migrated to a new platform where other functionalities are added.

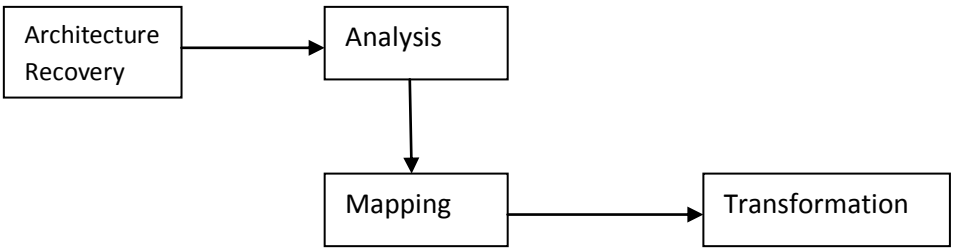

Figure 4.1 General Method

Architecture Recovery- To identify the legacy code architecture is needed for better understanding of the system. The concept of software architecture has gained a lot of attention and has found its way into the software development process in industry. To obtain a SOA from a legacy system first of all the deep knowledge of the legacy system will be taken from the existing documents, user manual and also from the code. And architecture of the existing project will be extracted.

Analysis - Then from the architecture the key services are identified based on the grammar. This analysis of existing system reveals the important data for designing reference architecture. The detail analysis results the gathering of system components, infrastructure details, interface presentation characteristics, QOS attributes, level of maintainability, level of complexity and coupling.

Mapping - Now the architecture for the new system is generated on the basis of the requirement of the system as well as user. Then this architecture is mapped to the legacy system architecture from where we can come to know that which component can be used and which are useless and can be ignored. And what new functionalities are to be added.

Transformation - After retrieving the essential modules to be used we have to our architecture style we have to transform these modules to a new platform. This transformation can be done on the same level of abstraction if the module is complete and do not have any problem then only a new interface can be provided known as wrapping or at advance level of abstraction if the module satisfies only some portion of the requirement then that module can be redeveloped.

V. Conclusion

The modernization of legacy system to SOA is very important and has clear potential benefits. Before developing or purchasing new service components, one should to try to reuse the old one. The technology for doing so is also available. It is important to choose a modernization strategy, technique and approaches very carefully otherwise it may lead to a chaotic end where everything is messed up.

This paper also presents a new approach to migrate a legacy system to a new platform. It consists of four steps: Architecture Recovery, Analysis, Mapping and Transformation.

This method has following contribution. This new method of migration of a legacy system to SOA will be an efficient method with less but sufficient number of steps. This approach will be system independent and can be applied over any legacy system.

References

[1] Asil A. Almonaies, James R. Cordy, and Thomas R. Dean, Legacy System Evolution towards Service-Oriented Architecture, International Workshop on SOA Migration and Evolution, Madrid, Spain, pp. 53–62 (March 2010).
[2] Maryam Razavian and Patricia Lago, Understanding SOA Migration Using a Conceptual Framework, Journal of Systems Integration (MAR'10).
[3] K.Velmurugan, M.A. Maluk Mohamed, A Model driven Approach for Migrating from Legacy Software Systems to Web Services.
[4] Ravi Khadka, Service Identification Strategies in Legacy-to-SOA Migration, Paper presented at the Doctoral consortium of the 26th International Conference on Software Maintenance (ICSM'11).
[5] CarlosMatos, ReikoHeckel, Migrating Legacy Systems to Service-Oriented Architectures, Electronic Communications of the EASST Volume 16 (2009).
[6] Liam O'Brien, Dennis Smith, Grace Lewis, Supporting Migration to Services using Software Architecture Reconstruction, Proceedings of the 13th IEEE International Workshop on Software Technology and Engineering Practice (STEP'05).
[7] Semih Cetin, N. Ilker Altintas, Halit Oguztuzun, Ali H. Dogru, Ozgur Tufekci and Selma Suloglu, A Mashup-Based Strategy for Migration to Service-Oriented Computing, In pervasive Services, IEEE Inetnational Conference, pp. 169-172, (2001).
[8] Félix Cuadrado, Boni García, Juan C. Dueñas, Hugo A. Parada, A Case Study on Software Evolution towards Service-Oriented Architecture, 22nd International Conference on Advanced Information Networking and Applications – Workshops (2008).