ABSTRACT

Legacy software is used in a large number of companies. To make software systems updated with current technologies, companies have two choices either create new software to replace the old or migrating the existing system into a new architecture that uses new technologies. Migrating existing systems can save a lot of time and money compared to the creation of new software.

We aim in this paper to study the migration of legacy software systems with its objectives and the challenges faced by different approaches in complex software systems. We mainly focus on service-oriented migration.

KEYWORDS

Reengineering; Legacy Systems; Migration; Modernization; Service Oriented Architecture (SOA).

1 INTRODUCTION

Today the evolution of software systems has become a key activity in the software industry. The law of entropy of software dictates that over time, most legacy software systems tend to be degraded gradually in quality, unless they are maintained [1]. Although, The Systems are not prepared to support new requirements (change of platforms, new features requested by users, the location of errors, etc.). Since that these cannot be anticipated and can be difficult to achieve because of the structure and the initial architecture of the systems. Therefore, these systems must be adapted to changing requirements.

The evolution of software systems covers several research issues. The changes start with a simple line of code to complete reimplementation of the system. Among the activities dealing with systems evolution there is the approach of migration [1]. This is an approach to reduce the cost of maintenance. As its name indicates, the underlying idea is to "remake" using appropriate methods and techniques. Remake, not only because something has been done poorly, but also because the way to do has changed. Migration is the passage of a current operating environment of a system to another generally better, ensuring continuity of operation. Migration can include the transition to new hardware, software or both simultaneously; it can be made as small scale like migration of a single system or high scale including multiple systems or applications [2]. Migrating of legacy system to new technology is a complex task. The difficulties lie in the understanding of the target system and the methodology of the migration process. All these factors emphasize the importance of minimizing the complexity of the migration in a reengineering project. In our study we are interested by the migration of legacy applications to service oriented systems. Which is one of the most popular target architectures for software modernization. SOA is supposed in order to manage the needs of distributed network oriented software. SOA are independent of platforms and languages.

In this paper, we present an overview of research in reengineering legacy systems and we focus particularly on migration to SOA. Our study aims
at positioning approaches. We raise the main challenges to which the migration approaches have outfaced for analysis and modification of legacy systems. We will present the solutions being proposed and the identified shortcomings.

In Section 2 we describe the basic concepts of Service Oriented Architecture. In Section 3, we present the techniques of Modernization and its different approaches. Section 4, provides a discussion of challenges and proposed solutions by presenting a comparative study of Modernization approaches. Section 5 concludes and prospects our future research in this area. Finally, Section 6 contains bibliography used in our work.

2 SERVICES ORIENTED ARCHITECTURE: BASIC CONCEPTS

The service concept is beginning to invest industrial domain in recent years. It is considered as the basic building block of SOA. The principle of SOA is to structure the information system of an enterprise as a set of services that expose their functional interface and communicate via messages [3].

Although, there is no consensual definition of what a service, defined by [4]. It is widely referenced in the literature. According to [4], a service is defined as "a set of modular applications self-contained and self-descriptive that can be published, located and invoked over the Web". A service can perform actions ranging from simple requests to complex business processes. The services enable integration of heterogeneous information systems using protocols and standardized data formats, allowing a loose coupling and flexibility via the technological choices made. There are several ways to perceive and define Service Oriented Architecture. Most of these definitions focus on the technical side of SOA, although others have characteristics crafts. There is a list (not exhaustive) of the proposed definitions and from multiple sources. These definitions are interesting because they illustrate several points of view on SOA. We will begin with a definition, which is very short. It is cited in [4]."SOA is a collection of components that may be invoked, and whose interface descriptions may be published and discovered [5]."

This definition presents a very simplistic way of what can be done with a service or with its description. It was not occupied by the notion of architecture or of the manner in which the service can be designed. A technical definition was presented in [6], "A Service Oriented Architecture (SOA) is a software architecture which relies on the key concepts of service, service repository, and service bus. A service consists of a contract, one or more interfaces, and an implementation". The authors point on the technical side of SOA, which consists of a front end application that uses one or more services. These services are published in service registers and communication between these services is provided by a service bus. From a business point of view [7], the SOA is presented as follows:"SOA is a conceptual business architecture where business functionality, or application logic, is made available to SOA users, or consumers, as shared, reusable services over a computer network. The "Services" in SOA are modules of business or functionality of application with interfaces exposed, and are relied upon by messages." This definition give a great attention to the business orientation of the SOA so that we can have the impression that SOA is purely a business concept and the technical level exists only for the maintenance of networks and communication between services. Based on these definitions, SOA is defined as follows: SOA is an architectural style that allows the reorganization of the Information System. It enables the encapsulation of functionality of information system in a loosely coupled set of services belonging to both the business level and technical level of the company. Services, provided with a contract of use and an interface description, will be published in service registers so that they can be invoked by other services. This definition gives a double vision of SOA a business vision and a technical vision. This separation of preoccupation is to ensure business agility.

2.1 Key Roles Of SOA

In an SOA three key roles are commonly identified: (i) the service provider, (ii) Service
Consumer and (iii) Directory of services. The interaction between these three roles is described in Figure 1. The service provider creates the service, and then publishes its description in a service book. This description specifies both the operations available and their method invocation. The customer accesses the service directory to search and to find the desired services. Then, a connection is established between the client and the service provider to ensure the invocation of the service [3].

![Figure 1. Meta-model of Service Oriented Architecture [3].](image)

### 2.2 Benefits of Service Oriented Architecture

The SOA offers the promise of a better alignment of information system on the needs of the business through reuse of services that make its value. Clearly, to draw all the benefits of reusable services, it must adopt a new approach to build its information system, different from those used previously. It should not aim only at the technical level of the company, but must rather be on all the information system of the company. In other words, the main challenge is to develop a pragmatic approach to implementing SOA within the enterprise [3].

### 3 MODERNIZATION

Modernization is a solution to migrate existing systems to a new architecture for improved longevity of business processes, in our study we focus in particular on the migration of legacy applications to service oriented systems. Three modernization strategies exist: Migration, Redevelopment and Wrapping.

#### 3.1 Modernization Of Legacy Information Systems By Migration

Migration is the passage of a current operating environment of a system to another usually best, ensuring continuity of operation [2]. Migration can include the transition to new hardware, software or both at the same time, it can be made at small scale as migration of a single system or high scale including multiple systems or applications [2]. Migration of legacy system may be a very expensive task that bears a certain risk of failure. To make a successful migration, a migration process model is evidently necessary. At present, however, there is no general model. Authors in [8] considers that the migration is composed of five main tasks, which are illustrated, in Figure 2:

1. Justification;
2. Understanding Legacy System;
3. Development of the Target System;
4. Migration;
5. Test;

![Figure 2. Main activities in migration of legacy system [8]](image)
computer organizations to face increasingly complex challenges. SOA can be both architecture, and a programming model, a way of conceiving software design. Service oriented architecture enables the design of software systems providing services to other applications using published and discoverable interfaces, and where services can be requested via a network. The implementation of SOA using Web services technologies, have created a new method of building applications within a programming model more powerful and flexible. It can reduce development costs and property, and the risks of implementation [9].

Migration to Service Oriented Architecture is a challenging task complicated by the lack of suitable tools and approaches.

### 3.1.1 Approach Proposed By Lewis et al (SMART)

The authors of this approach presented a process called SMART (Service oriented Migration And Reuse Technique). This approach consists of six phases, namely: establishing the context of migration, this phase is to gather as much information on the objectives, expectations of migration and users of migrating legacy systems and future system. This information is received from discussions with developers using a guide said SMIG « the Services Migration Interview Guide ». Then the authors [10] propose to control the feasibility of migration to test if the IS of the company is a good candidate for migration to SOA or not. Once the migration system is feasible, the authors of this approach propose to select from the initial list the candidates’ services established in the phase of the study of the context of migration, those who are good candidates. All services would be selected in this phase, then, well specified by defining for each service the inputs and outputs, and the information quality of service expected (QoS). The next step is to describe existing capacity using the guide «SMIG» to collect information on the parts of code that implement the business logic of services defined in the previous phase, then the phase of the SOA environment description that identifies the technical information and expected performance of candidate services of the future SOA environment. The results of the two phases described above are reconciled to identify the services constituting the SOA environment. This step is called (gap analysis). To complete this approach, the authors of SMART propose to choose the migration strategy to achieve the expected goal by using all information resulting from the previous steps.

### 3.1.2 Approach Proposed By Semith et al (MASHUP)

The authors [11] of this approach suggested the use of technology applications composite Web or MASHUP (MigrAtion to Service Harmonization Computing Platform). This approach consists of six phases; it starts with modeling the business needs of the target company by a semantics notation similar to Business Process Modeling Notation (BPMN) in order to understand the capabilities of the future information system (IS). Then, the analysis of existing legacy systems to obtain a set of information on the current system that is: the architecture of IS and its components, infrastructure details, the characteristics of user interfaces, quality attributes of service, the current state of the system (the level of maintainability, the level of complexity and coupling of these software components...etc). The next phase consists in the mapping and the identification of services that allows the reconciliation of model’s needs to system software components for the identification of services. The new system will either reuse existing capabilities or develop new components, then the design of the MASHUP server architecture and the definition of SLA (Service Level Agreement). The purpose of this phase is the construction of a repository that contains the necessary information describing the context and quality of service (QoS) within the server. Finally, the authors [11] propose the implementation and deployment of services.

### 3.2 Modernization Of Legacy Information Systems By Redevelopment

Redevelopment of legacy systems implies rewriting legacy systems from the outset. [12] proposed a general approach for redevelopment. At first the legacy code is analyzed to win an understanding of its features. This analysis helps
to identify business processes performed by the legacy system. These business processes are subsequently implemented as services.

There are various advantages to redevelopment. The new software will be designed from the outset to be used in SOA. As the system is written in a way that supports SOA it is easier to maintain and extend in SOA. Redeveloped software does not have additional layers added by wrapping, and therefore does not suffer from the loss of performance caused by additional layers [13].

The redevelopment produces software with desirable qualities but its cost in time and money may be prohibitive.

3.2.1 Approach Proposed By Sahraoui

A novel approach is proposed by authors to reduce the migration complexity. They carry out dynamic program of analysis, visualization of software, recovery of knowledge, and techniques are divided and conquered to face the issue of complexity in legacy software migration project. Analysis tools are also developed to support their approach.

They concentrate on three crucial aspects of their migration methodology to efficiently minimize the complexity of migrating a large legacy software system to a new technology based paradigm, [14].

- **The first aspect is understanding of the legacy system**

The system in question will be analyzed in different ways to reveal the outline of the architecture of system. Programmers are inevitably confronted with the challenging task to understanding legacy software system. How to minimize the difficulty and the complexity of understanding legacy system becomes an urgent problem for researchers.

- **The second aspect is the decomposition of legacy system into small, easy grip, and strongly cohesive divisions.**

Into a legacy system, the user defined the type of data and the data structure can be considered as an indicator of the relations between modules and the remains of the system. Therefore, they can decompose the entire system into several parts, which have a strong cohesion value depending on their module dependency relations.

- **The last aspect is the organization the migration process in a cost effective manner, which reduces the complexity in treating those decomposed parts of target system.**

This is mainly realized by giving priority to the migration sequence of the individual migration units.

After having correctly decomposed legacy system, they'll cope with the problem of which MU (Migration Unit) must be migrated first. For a some MU decomposition diagram, At first its characteristic is analyzed, then compared with available resources of reengineering, finally an appropriate priority strategy is adopted, applied on all of the MUs, and the best candidate is selected to migrate first.

3.3 Modernization of Legacy Information Systems By The Wrapping

The wrapping is another method used for the implementation of SOA from legacy code. the useful legacy code is sticked with code wrapper to be integrated into an SOA system.

Several ways existed to realize this wrapping. Sneed [15] proposed an effective methodology in decoupling code from the original system. The relevant legacy code should first be specified. This may usually be achieved by automated tools of reverse engineering; clustering tools and data flow graphs identify the code fragments, which carry out a desired service or modification of data these code fragments are extracted, and using these segments a new component is built. Using the required data objects the interface of the new component is then formed. Automated software can also do this specific wrapping stage. Finally, a proxy is made to relate the new services in SOA. The proxy is mainly concerned with the verification of the parameters. This enables the wrapped legacy code to be integrated into the SOA model.

Among the approaches of migration based on the Wrapping, we include this one that we consider interesting to know:
3.3.1 Approach Proposed By Boukadi (CSOMA)

This approach is called CSOMA for (Contextual Service Oriented Modeling and Analysis). This approach builds a service oriented architecture for the enterprise. It starts with an existing architecture “As-Is”, and find a target architecture “To-Be”. The particularity of this approach is that it allows for the adaptation of services at the time of conception. It begins with studying the needs of defining factors and expectations that motivate the establishment of this service-oriented architecture through the use of the standard BMM (Business Motivation Model) and the study of the existing architecture and target. Once needs are defined and modeled, the author [16] proposes to build the SOA business of defining and formalizing business processes to go through the different layers of SI to define business services. This phase involves six steps, it starts with modeling the business process candidates identified by the standard BPMN (Business Process Modeling Notation) then identify the business services by decomposing the strategic goals of each business process in tactical goals and tactical goals in operational goals, then the tactical goals will explore to identify services area. As for operational purposes they are used to study the coupling between the activities of a business process to identify functional services. These functional services will be subsequently modeled and then refined to meet the properties of reuse, autonomy, and granularity of services. These steps will be followed by the identification and modeling of the conceptual aspects of business services. This phase ends with the creation of models schema of service orchestration areas. The next phase of the approach is to build the SOA IT (Information Technology), which begins with an analysis of the applicative portfolio of the company to examine the functions, developed and then identify services entities that provide access to information about business objects. These services are identified through reverse engineering in accordance with a set of rules. The approach of [16] ends with a consolidation phase that is to compare the results of the two phases (business phase and Information Technology phase) to validate the services that have already been identified.

3.4 Comparison Of The Techniques

Table 1 summarizes the strategies for modernization. Every strategy of the modernization uses a different method to get the result.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
</table>
| Migration  | -Stable environment  
-Availability of tools | -Takes time  
-Skilled resources necessary  
-Source code required | |
| Wrapping   | Rapid                                           | -Inflexible  
-Maintenance difficult | |
| Redevelopment | -Improve agility  
-Flexibility  
-Low cost                     | -Source code required  
-Original requirements necessary | |

Table 1. Summary of Strategies For Modernization [17]

An initial set of strengths and weaknesses for every strategy is summarized in table 1 [17]. The required goal can be achieved by combining two or more modernization strategies according to the advantages and disadvantages of every strategy in the context. The reuse of legacy system components and their exposition as services are not always easy. In certain situations, their exposition, as services will have an increased risk and a greater cost than their replacement completely with a new SOA architecture. The problem of modernization of legacy system has no perfect solution. Choosing a strategy is based completely on the objectives of the SOA architecture, the budget and resources available and required time to achieve the project.

4 COMPARATIVE STUDY OF APPROACHES OF MODERNISATION

Modernization of legacy software is facing many challenges:

- The absence of a universally accepted solution for modelization of legacy software systems already in place.
• The absence of appropriate approaches and tools for migrating legacy applications to other platforms.

• Necessity of tools and techniques which maintain or improve the quality characteristics of a software system, regardless its size and complexity.

We compare the approaches with respect to the following criteria [17]:

• **Strategy of Modernization.** One among of redevelopment, wrapping and migration.

• **Deepness of Analysis.** The aim of the strategy is the analysis of the legacy system for understanding its concepts and locating the important functions to be exposed as part of SOA architecture. The analysis may be superficial or deep according to the strategy used.

• **Level of Coverage.** Indicates if the proposed approach presents a complete strategy for the displacement to SOA, or just a particular part of the modernization?

• **Ripeness of Validation.** Indicates if the proposed approach has been applied and validated. The proposed approach is classified as an idea, a method proved by a case study, or a commercially demonstrated technique.

According to the comparative study that we conducted between modernization techniques we have concluded that all the techniques supply a case study to support their demands, excepting SMART [10] that is presented as a set of guidelines to manage the process of migration. As SMART is legacy system independent, it might be used in one of the other techniques to evaluate the legacy system implicated in the migration.

### 5 CONCLUSION

Migration is a support for the evolution of existing software but can also be seen as a means of facilitating technology transfer. In this paper, we presented the modernization of legacy systems with emphasis on the challenges to which the modernization approaches are facing.
for analysis and modification of legacy systems. In our work we focused on the technique of migration of legacy systems to Service Oriented Architecture. This technique is an appropriate solution but it is a complex engineering work. We discussed the solutions that have been proposed to reduce this complexity aiming in the near future to develop unified process for the treatment of Migration.

6 REFERENCES