Migration of Existing or Legacy Software Systems into Web Service-based Architectures (Reengineering Process): A Systematic Literature Review

Siti Rochimah
Department of Informatics,
Faculty of Information Technology,
Institut Teknologi Sepuluh Nopember (ITS)
Surabaya.
Kampus ITS Sukolilo,
Surabaya, Indonesia.

Alhaji Sheku Sankoh
Department of Informatics,
Faculty of Information Technology,
Institut Teknologi Sepuluh Nopember (ITS)
Surabaya.
Kampus ITS Sukolilo,
Surabaya, Indonesia.

ABSTRACT

In this study, the techniques of migrating legacy software systems into web service-based architecture has been discussed. This procedure is carried out based on the fact that software and business companies want to migrate along with previous systems, to a more recent technology, which are of higher benefit to both businesses and their client. With the above issue in mind, this study executed a systematic literature review to look into the various techniques that were existing in published literatures or articles and come up with a comprehensive summary that highlighted their strength and weaknesses for future improvement.

This research obtained it's publish articles or literature by exploring two popular publication sources. Namely, IEEExplore and ScienceDirect. We considered including article or literatures that were published within the past ten years (2005 – 2015) and with definite keywords that were used as search queries to obtain the articles from the aforementioned repositories. The seventeen (17) selected articles or literatures were then processed based on certain inclusion criteria, by considering existing methods of migrating legacy software systems into web service-based architecture.

The accomplished systematic literature review has provided a classification of legacy software migration methods and has highlighted the fact that each of the methods has at least one advantage and disadvantage that are seen as benefits of each of those methods. The method types used were then categorized as either analytical or wrapping methods. In the case of analytical methods, the migration process is performed by examining the existing legacy software system to distinguish how the migration process can be carried out with limited bottlenecks. Whilst inside the wrapping method, the migration is done by directly wrapping the source code of the existing system to generate web services. The advantage of Analytical methods is a structured migration process and its disadvantage is the required effort needed to carry out the process to obtain best results. The advantage of the wrapping methods is the possibility to obtain very quick results and its disadvantage is that the results are highly dependent on the effectiveness their algorithms.

Keywords

Legacy software migration, web service-based architecture, wrapping techniques, analytical techniques

1. INTRODUCTION

The speedy growth of the World Wide Web has started taking full control of the medium of cooperation among several systems. Nowadays, web applications are legacy systems that demonstrate valuable resources to their host systems. Compare to a website, which supplies its users with only a chance to read information through the World Wide Web (www) framework, a Web Application is an authentic software system that exploits the World Wide Web infrastructure to give its clients the chance to change the status of their systems together with their current job status.

Web services are interoperable, standards-based software components which can be accessed via Internet connections. They can be applied as construction blocks to construct software applications whose functionality may be simple or complex. They are extensively recognized in the IT industries. Foremost IT corporations such as Microsoft, Oracle, SAP and Sun support web services. These technologies are becoming broadly established in commercial enterprise and multi-national companies. Web services are definite in terms of a stack of developing XML-based open standards for service description (WSDL), composition (WSFL), and invocation (SOAP)[1].

Today the evolution of software systems has become a central activity in the software business. 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. In this possibility, maintain means to enhance a software system updated with resent technologies. To upkeep a software, companies have two choices, either create new software to replace the old or migrating the existing system into a new architecture that uses novel technologies. Migrating an existing system can keep a good deal of time and money compared to the founding of novel software[2]. This work aims to achieve constructive planning approaches, methodologies, or techniques used in migrating legacy system into a web service, which is one of the most popular target architectures for software modernization (software migration).

1.1. Web Service

There are so many existing software applications that are precise to various areas in the field of computer science. To name a few, are Desktop, Mobile, and Web-based applications. The aforementioned applications has many functionalities and components that are combined in their modules to enhance the implementation of their desired functions. Web services are interruptible standard based software components that can be gain access to via internet[1]. Web services technology are growing rapidly and widely supported by upcoming web application. They consist of a circle of related specifications that outline the way in which components should be conditioned through the Web Service Description Language (WSDL). That is, the direction in which they should be raised so that they can be found and reused through a Universal Description, Discovery, and Integration API. Their Simple Object Access Protocol API (SOAP) also demonstrates how they should be appalled at run time.

Service Oriented Architectures (SOA), is the foundation for several web services. It's figuring out to be the basis of service oriented computing. Service Oriented Architectures goes along with some mechanisms, for e.g. consumer attendants, service providers and service repository. Merely, the service usage, such as delivery, acquisition, consumption, composition and so on, are based on SOA architecture. This architecture is a significant paradigm which assist the use of service management. Looking at it in the point of view of a service, SOA is a significant architectural evolution that honors the life cycle of a software [3].

Furthermore, application with the same functionalities, are possible to have the chance to access the same web service through the internet. Therefore, the web service architecture is especially applicable when multiple applications running on various technologies and platforms that need to communicate with each other[1]. Base on this situation the response to develop automated reengineering methods for updating legacy systems into web services from the bases of existing functionalities that are already available through web application, were introduced in several organizations and Business Corporation. Which implies that there is no need to get the same functionality repeatedly.

Ultimately, this research aims at carrying out a systematic literature review, which focus deeply on reviewing existing research reports and literatures for transforming legacy software application into web service-based that have been written over the past 10 years (2005-2015). Thus, we achieve objective analysis (based on existing literatures) that can aid developers to transform legacy software applications into web service-based. This suggested approach will be based on the three components below, as indicated in the figure1 below. In computing, a legacy system is an outdated method, technology or computer system. Application program relating to software of previous computer system are also part of this term. It can also be depreciatory term, referencing a system as "legacy" often implies that the system is out of date or needed to be replenish[2]. A software migration system refers to the transformation, rewriting of a legacy system to a recent one.

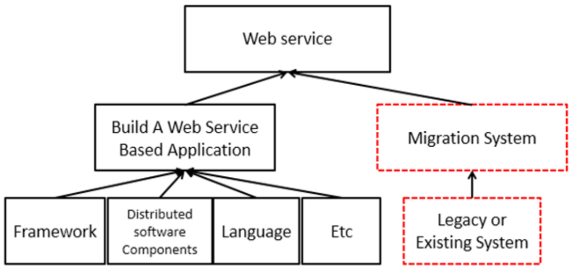
Web service
Build A Web Service Based Application
Migration System
Framework
Distributed software Components
Language
Etc
Legacy or Existing System

Figure 1 Three stage model of Web Service architecture

In this case, the migration of a system means the migration of legacy software system into a web service. This research focuses on literatures that wrote on the approaches, methodologies or techniques used in migrating system into web service-based. There are a lot of topics that discusses about web service such as: technology used in web service, framework, languages, architecture and other topics.

2. RELATED WORKS

Duen-Kai Chen [4] 2010 performed a systematic review of academic papers from IEEE Explore regarding SOA in Networks by looking at the following issues; current state of the application of SOA for networking was highlighted, research efforts in related area were also discussed. Some of the weaknesses of this research is that, the researcher only limits his research on the publications provided in the IEEE Explore Digital Library. Furthermore, his review did not follow a structured systematic literature review methodology.

Another review carried out by Khubaib Amjad Alam et al [5] 2014 was based on a larger study to systematically review all available research on impact analysis and propagation in the context of Business process management (BPM) and Service Oriented Architecture (SOA). John Hutchinson et al [6] 2007 provide a review of a number of approaches that may contribute to a pragmatic strategy for addressing the problem (existing software systems) and outline the significant challenges that remain. The difference between their study and our SLR is that. They did not take in to consideration web application that are not fully service-based.

Further research conducted out by Abhishek Sharma et al [7] 2012 was a systematic mapping study. Their paper provides an overview of the current state of research into testing of web services. To understand their subject, they conducted a systematic mapping on the testing of web service. The results suggest that research into testing web services is still in its preliminary stages.

It is proven that, from the list of existing Systematic Literature Reviews performed in reengineering of web service, there hasn't been any attempt to systematically review legacy software systems into web service-based. This is the unique feature and major contribution of this paper with the expectation to provide a systematic review regarding this topic. A systematic literature review (SLR) is aimed at identification, evaluation and interpretation of all the available research related to a particular field of interest. An SLR must be achieved with a rigorous search plan which can be attributed as fair and unbiased. The search strategy or plan must ensure the completeness of the search for assessment.

3. SLR RESEARCH METHODOLOGY

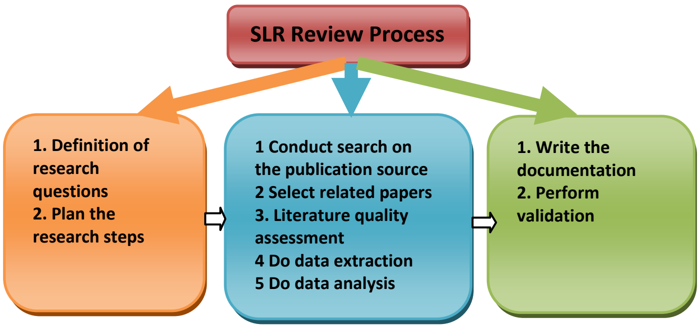
SLR Review Process
1. Definition of research questions
2. Plan the research steps
1 Conduct search on the publication source
2 Select related papers
3. Literature quality assessment
4 Do data extraction
5 Do data analysis
1. Write the documentation
2. Perform validation

Figure 2 SLR Research method steps

The main objective of this paper is to present an overview relating to the transforming legacy software application into web service-based and to evaluate the advantages and disadvantages of the various approaches. As indicated above, the beginning point of this research is the planning phase, which is to determine the scope, research questions, and steps needed to answer the research questions. The next stage is conducting the review itself by searching for papers in two publication sources, filtering process based on inclusion and exclusion criteria, quality assessment, data extraction, and analysis process. The last stage of the research is the documentation stage, which includes the writing of the paper and validation process. Figure 2 shows the research method used in this SLR.

At the point of this report, there are little or no systematic study to provide a rigorous review and analysis of existing research in this domain. The target of this research is to fill this gap by conducting an SLR using SLR guidelines by Kitchenham [8]. Our systematic review process comprises of several steps which have to be performed in a systematic and disciplined way. These steps include, development of a review protocol, conduct systematic review, analysis of results, reporting and visualization of results and discussion of findings.

3.1 Research Questions

There are three research questions (RQ)that are going to be explored, analyzed and concluded regarding to the migration of legacy or existing software application into web-based service. These are specified as follows.

- RQ1: How many kinds of technique used in migrating a legacy or existing software into web service?
- RQ2: What are the methods or approaches use in each technique?
- RQ3: What are the advantages and disadvantages of each technique applied in the translation process?

3.2 Digital Library Sources

In this research, we uses two digital libraries in our searching process. Both of them can be accessed online: IEEE Explore [17] and Science Direct [18]. The aforementioned digital libraries were selected base on the limitation of freely available digital library access in our institution. We must admit to the fact that there are other digital libraries available worldwide. Yet, not all of them are freely usable in our intuition for free article download. They are consequently omitted from being part of our main article sources.

Table 1. Online Digital Library

ID	Digital Library	Website link
DL1	IEEE Xplore	http://ieeexplore.ieee.org/
DL2	ScienceDirect	http://sciencedirect.com/

3.3 Search Keywords

Beneath are the keywords that were employed in our literature search.

- Migration of existing software systems to web service-based architectures
- Software Migration into Web service based
- Reengineering Software into web service

The above keywords were used as a search query base on the two digital libraries mention above to search for any journal article or group discussion paper that were published within the period of 2005 to 2015. The search results will then be picked out in the subsequent stage (inclusion and exclusion criteria).

3.4 Inclusion and Exclusion Criteria

The inclusion criteria determine the literatures that will be included in the data selection stage. While the exclusion criteria determine which articles should not be included during the selection stage. Thus, inclusion and exclusion criteria are tabulated in the table below.

Table 2. Inclusion and Exclusion Criteria in this SLR

Inclusion Criteria (IC) ID	Inclusion Criteria Details
1st IC	Articles that discuss about Migration of existing or legacy software systems to web service-based architectures
2nd IC	Articles that were publish from 2005 to October 2015
3rd IC	Articles present any procedures / methods of Migration of Existing or Legacy Software Systems into Web-based Service
4th IC	Article written and published in English
5th IC	Articles without duplicate titles.
Exclusion Criteria (EC) ID	**Exclusion Criteria Details**
1stEC	Articles that are not talking about migration of software systems Migration into web service
2ndEC	Articles that are written prior to 2005 are not included
3rdEC	Articles that are not written in English
4thEC	Articles with duplicate titles are not included

3.5 Results from search

Table 3 shows the search result obtained from the search queries. A total of 39 literatures were obtained, which hasn't been further selected. The selection process is then piloted into three stages. Figure 3 shows the selection process and the number of literatures selected in each stage.

Table 3. Search result in online libraries

Keywords	IEEEXplore	Science Direct	Sub- Total
	Journals & Conferences	Journals Conferences	
Migration of existing software systems to web service-based architectures	**2**	**0**	**2**
Software Migration into Web service based	**16**	**2**	**18**
Reengineering Software into web service	**16**	**3**	**19**
TOTAL			**39**

3.6 Article Selection Method

This review follows a structured article selection method which is separated into three phases, shown in figure3 below.

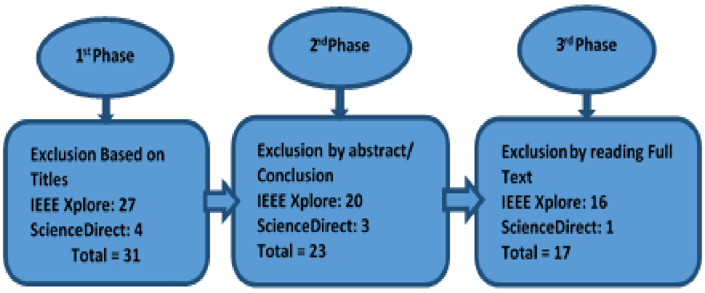

Figure 3 Article Selection Method

From the figure above, one can be able to see the article selections method base on the highlighted three phases which serves as the inclusion and the exclusion criteria that are explained in the next subsection.

3.7 Details of the three Selection Phases

The first step of the selection process is to put the keywords as search queries in the two digital libraries, which produces a total of 39 articles. The titles then selected to find candidates about migration of software or legacy systems into web service-based systems. If the title clearly indicates that it's not about migration of web service-based systems, then the title will be excluded instantly. This selection stage produces 31 candidate articles.

To further limit the candidate articles, the result then processed in the next stage. After reading the abstracts, the candidates are down to 23 titles. The gap from the previous stage is mainly because of the frequently misleading titles about the general migration of legacy software application into web service-based. The most common case is there is no clarity in the title regarding where the web service-based is applied. This misleading title also occurs in many titles, which is why the resulting selection from third stage only produces smaller number.

The next stage is to read the whole study to determine the final selected titles. There are 17 main articles which are produced in this stage that will further processed in quality assessment before they can be used in the data extraction process. Below are the 17 titles, their authors, publication channels, and their publication year.

Table 4.. The Selected Studies based on the three Selection Phases

No	Name(s) of Authors	Titles of Publications	Publication channels	Year
1	Djelloul Bouchiha & Mimoun Malki	Towards Re-engineering Web Applications into Semantic Web Services	IEEE Conference	2010
2	Saeed. Parsa & Leila. Ghods	A new approach to Wrap Legacy Programs in to Web Service	IEEE Conference	2005
3	Miguel Ángel Conde González et al	Adapting LMS Architecture to the SOA: An Architectural Approach	IEEE Conference	2009
4	García-Rodríguez de Guzmán et al	An ADM Approach to Reengineer Relational Databases towards Web Services	IEEE Conference	2007
5	Feng Chen et al	Feature Analysis for Service-Oriented Reengineering	IEEE Conference	2005
6	Grace Lewis et al	Analyzing the Reuse Potential of Migrating Legacy Components to a Service-Oriented Architecture	IEEE Conference	2006
7	Harry M. Sneed	Integrating legacy Software into a Service oriented Architecture	IEEE Conference	2006
8	Jianzhi Li and Hongji Yang	Towards Evolving Web Sites into Grid Services Environment	IEEE Conference	2005
9	Gerardo Canfora et al	A wrapping approach for migrating legacy system interactive functionalities to Service Oriented Architectures	ScienceDirect Journal	2008
10	Ekasari Nugraheni	Migration of Web Application SIMA into Multi-tenant SaaS	IEEE Conference	2013
11	Anca Daniela Ionita, Stefan Aureliu Radulescu	Metamodeling for Assigning Specific Roles in the Migration to Service-Oriented Architecture	IEEE Conference	2012
12	Bin Cai et al	Research and Application of Migrating Legacy Systems to The Private Cloud Platform with CloudStack	IEEE Conference	2012
13	Pushparani Bhallamudi et al	Migrating a Web-Based Application to a Service-Based System – An Experience Report	IEEE Conference	2009
14	Giusy Di Lorenzo et al	Turning Web Applications into Web Services by Wrapping Techniques	IEEE Conference	2007
15	Saad Alahmari et al	A Service Identification Framework for Legacy System Migration into SOA	IEEE Conference	2010
16	Naoufel Kraiem & Zuhoor Al-Khanajari	From legacy Web applications to Web Services based applications	IEEE Conference	2015
17	Sandro Geric, Neven Vrcek	Prerequisite for Successful Implementation of Service Oriented Architecture	IEEE Conference	2009

3.8 Study Quality Assessment

This SLR chooses four quality assessment questions to conclude the quality of the included data.

Q1. Is the method of Migration of Existing or Legacy Software Systems into Web Service-based described clearly and comprehensively?

This question aims at evaluating the clarity aspect of the literature. The description of the method should be highlighted in a clear manna within the articles. It could be within one specific section or more. Lacking this evaluation process, is an indication of irrelevant literature quality.

Q2. Which type of publication channel that was used to publish the selected study or literature?

This question aims at addressing whether the articles were published in journals or conference (IEEE or ScienceDirect).

Q3. When was the selected study or literature published?

This question aims at telling whether the article can be reflected as outdated or not. The more recent articles are, the greater the chance to be considered as significant.

Q4. How many other studies that has referenced the selected study or literature?

This question aim at assessing the quality of contribution and relevance of the article. Which means, a good study is reflected in in the amount of useful research that has refer to it.

The absolute selected studies are then analyzed with the quality assessment questions mentioned above. The articles which don't provide sufficient descriptions of their study are marked as "Not Clear" in Q1. The result of this assessment is presented in Table V.

Table 5. Quality of Assessment for the Individual Literatures

No	Ref. No	Name of Author (s)	Q1	Q2	Q3	Q4
1	[8]	Djelloul Bouchiha & MimounMalki	Yes	IEEE Conference	2010	1
2	[9]	Saeed. Parsa & Leila. Ghods	Yes	IEEE Conference	2008	0
3	[3]	Sandro Geric, Neven Vrcek	Yes	IEEE Conference	2009	0
4	[10]	Miguel Ángel Conde González et al	Yes	IEEE Conference	2009	3
5	[11]	García-Rodríguez de Guzmán et al	Yes	IEEE Conference	2007	1
6	[12]	Feng Chen et al	Yes	IEEE Conference	2005	15
7	[13]	Grace Lewis et al	Yes	IEEE Conference	2006	15
8	[14]	Harry M. Sneed	Yes	IEEE Conference	2006	50
9	[15]	Jianzhi Li and Hongji Yang	Yes	IEEE Conference	2005	1
10	[16]	Gerardo Canfora et al	Yes	ScienceDirect Journal	2008	63
11	[17]	Ekasari Nugraheni	Yes	IEEE Conference	2013	0
12	[18]	Anca Daniela Ionita	Yes	IEEE Conference	2012	0
13	[19]	Bin Cai et al	Yes	IEEE Conference	2012	0
14	[20]	Pushparani Bhallamudi et al	Yes	IEEE Conference	2009	3
15	[21]	Giusy Di Lorenzo et al	Yes	IEEE Conference	2007	2
16	[22]	Saad Alahmari et al	Yes	IEEE Conference	2010	6
17	[23]	Naoufel Kraiem & Zuhoor Al-Khanajari	Yes	IEEE Conference	2015	0

3.9 Data Extraction and Analysis

The data in the selected articles were extracted to answer the above research questions in section 3.1. The data extracted from the articles are related to the Migration of Existing or Legacy Software Systems into Web Service-based methods. The data was analyzed to identify the case study, tools, advantages and disadvantages of each method used in the various studies. The data was obtained based on the thorough reading and examination of the 17published articles. On completion of data gathering, it was analyzed based on the research questions. Details of the data extraction is provided in the Appendix with the exception of the advantages and disadvantages which are placed in section 4.3.

4. RESULTS

Based on the seventeen selected studies or literatures as candidate of this SLR research, there are literatures which propose or improve a methodology technique or framework for migrating legacy systems. That is, literatures which present tools to support migrating legacy system into web service, and a literatures about experience report in migrating legacy system to web service. Each of those will be addressed to the three research questions associated with this SLR.

4.1 RQ1: Kinds of Technique Used In Migrating a Legacy or Existing Software into Web Service

Two main classifications were found with respect to the data set extracted from the 17 selected candidate. The first classification specifies the type of transformation based on "Method type" in the migration process of legacy software systems. The second classification specifies the "Migration resource" to generate web service-based systems.

4.1.1 Classification based on "method type" to transform legacy systems into web service-based:

Within this classification method, there are three types of transformation models which can be used for the purpose of legacy systems migration towards the reengineering process of web service-based systems. These includes: analysis of existing legacy system, wrapping of source code and others. The articles related to these three types of classification are recorded in the Table VI below.

Table 6. Categorization based on the "method type" to migrate legacy systems into web service-based

The Type of Methods	Description	Cited Ref.
Analysis of existing legacy system and Wrapping	This approach proposed a step-by-step analysis of the existing software or legacy system to be transform into web service-based. Which means all the steps are clearly highlighted and follow best practice.	[9][11][12][13][17][18][19][22][23][3]
Wrapping of source code	This approach directly uses a strategy to transform the codes of the existing software or legacy systems.	[8][10][14][15][16][21]
Others(Experience Report)	This type of approach uses another technique or an experience report to in order to transform legacy or existing system.	[20]

There are ten studies which uses analysis of existing or legacy software systems to migrate into web service-based. Six studies uses wrapping of source code directly to transform legacy software in to web service-based without analysis by using a strategy based approach as mention in the table above. Lastly, only one study uses other approach based on an experience report to transform legacy software systems[20].

4.1.2 *Classification based on "migration resource" to migrate legacy software systems into web service-based:*

Apart from the classification of migrating legacy systems based on the "method type", it is also important to know that there is a possibility for migration based on the "migration resource". This classification type is categorize in two sub-categories. Which means, the source program to be migrated could be either a "Web Application", or "Non-Web Application".

A migration resource will be included in the web application category if the previous application are running on a web platform and was built based on a web driven language. Otherwise it will be included in the non-web application category. This analysis process is illustrated in the categorization result shown in Table VII below

Table 7 Categorization based on the migration resource to transform legacy system

Migration Resource	Description	Cited Ref.
Web Application	This classification type is based on the fact that the precious legacy software or application to be migrated into web service-based, was develop to be use in a web platform. And the language use was a web driven language.	[8][10][15][17][20][21][22] [23]
Non-Web Application	This classification type is based on the fact that the previous legacy software or application program to be migrated into web service-based was not a web driven program.	[9][11][12][13][14][16][18][13][3]

From the results shown in Table VII above, eight (8) of the studies uses web-Applications to transform or migrate legacy systems in to web service-based. While nine (9) of the studies uses non-web applications to transform them into web service-based.

The migration of various web applications had brought up so many methods especially for UML web based systems. The proposed transformation method between Unified Modeling Language (UML) to Business Process Modeling Notation (BPMN) [22], i.e. transforming UML activity diagrams to BPMN diagrams. It also adopt an SOA meta-model that defines service types along with the semantics of models. There are also rules derived from UML class diagram by forming queries based on design differencing.

In [23], they propose a method of classification of existing Web-applications. Then, they present a step-by-step decomposition of these Web-applications to identify a high level of granularity and reuse of logics functionality. Next, they propose a software of model transformation and briefly present evaluation of existing software of model transformation according to offered functionalities.

The non-web applications are now also been taken on board intended for their transformation into web service-based systems. Certainly, just like any other system transformation, there are analytical as well as code wrapper tools to assist a smooth migration process. In [18], a metamodeling system was developed. The migration role goes in a step wise manna. First, they look at the source, e.g. a legacy application, or enterprise data, followed by the target, e.g. an SOA-based system, or a non-SQL database. Finally, the transformation is performed for reusing the source and developing the desired target, e.g. wrapping, reengineering, refactoring, restructuring etc.

However, other methods are also applicable based on the non-web application. A wrapper-based architecture for the migration of a legacy system to Web Services was also developed by [16], from which a black-box modernization approach for exposing interactive functionalities of legacy systems as Services was used.

The previous analysis above has indicated the categorization of the various method types which shows that two different approaches in the transformation of legacy system in to web service-based. A lot of transformation towards web service-based indicated that the process is done with the help of an analytical strategy[23][19][18][17][13], which enables fast and easy migration to web service-based.

Consequently, there are other types of methods that uses wrapping approach to achieve a very quick transformation during the process of migration towards web service-based. Research has indicated that, there are three basic steps required to create web services from legacy code. First of all there is a need to Salvaging the legacy code, followed by Rapping the salvaged code and finally, making the code available as a web service[14].

4.2 RQ2: What are the methods or approaches use in each technique?

One kind of techniques used in migrating legacy system is Grid Service Oriented reengineering approach. This proposed approached employs cluster and slicing techniques to extract and translate web sites information into XML document. Then, based on the web sites (WS) resources framework, the extracted information is deployed in the Grid environment as “stateful” resources. This method can be used for reengineer HTML based web sites and it will bring more flexibility, expansibility, and reusability, as well as more reliability to them.

Another research used feature analysis. This feature analysis is combined process including identifying system features, constructing the feature model to organize the identifying features in a consistent way, tracing the relationship between the features and the implementation of legacy system through feature location techniques. With feature based information, service identification and packaging process are performed and resulted into a service delegation, which integrates reusable software components into service constructions.

Architecture Driven Modernization (ADM) focused on the analyzing of legacy system to discover and create function-abilities to be exposed as service using web service. Their research focus on relational databases (SQL-92 databases) as a very extended and valuable type of legacy system. The proposed process consists 5 steps: (1) Database reverse engineering, (2) first service extraction, (3) PIM generation, (4) service discovering, and (5) Web Service Description Language (WSDL) generation.

Giusy Di Lorenzo et al, adopt Black-Box Migration Approach based on wrapping to migrate functionalities of existing web application to web service. This approach is based on a migration process that relies on black-box reverse engineering techniques for modeling the Web application User Interface.

Gerardo Canfora used the same technique from Giusy Di Lorenzo et al research to migrate functionalities of legacy system. Their research tried to solve the problem of transforming the original user interface of the system into the request/response interface of SOA by a wrapper. The wrapper behavior is defined in the form of Finite State Machines retrievable by black-box reverse engineering of the human–computer interface.

Saeed.Parsa et al propose a new approach and an automatic wrapping tool to expose legacy program functionality. The proposed approach consists of two stages of analyzing and web service wrapping of legacy code. Djelloul Bouchiha proposes a framework for reengineering web applications, possibly already including some semantic technology, into WSMO-based SWS. The proposed framework consists in reverse engineering web applications towards conceptual models specified with proposed UML profile, from what syntactic and semantic description of new SWS are generated. Saad Alahmari et al proposes a novel framework which identifies the key services effectively. The framework approach focuses on defining these services based on a Model-Driven Architecture approach supported by a SOA model-Driven Architecture. Ekasari Nugraheni proposes a framework that is used to perform the migration process SIMA application into multi-tenancy model.

4.3 RQ3: Advantages and Disadvantages of Each Technique use in the Transformation process?

This SLR tried to collect the appraisals of writer’s methods and make use of their data to generate objective insight about the advantages and disadvantages of each method in the migration process, particularly in analysis of existing systems and wrapping methods. Below is a tabular representation of the advantages and disadvantages of each of the studies.

Table 8 Advantages and disadvantages of each of the selected studies for legacy systems migration to Service-based

Cited Ref	Title of Literature	Advantage(s)	Disadvantage (s)
[15]	Towards Evolving Web Sites into Grid Services Environment	Dynamic and accessible from any location. This method can be used for reengineer HTML based Web sites and it will bring more flexibility, expansibility and reusability, as well as more reliability	This proposed approach can works well for HTML static Web site, but it does not support the dynamic Web sites, such as web pages written by ASP or JSP.
[14]	Integrating legacy Software into a Service oriented Architecture	This article has proven that specific business functions can be extracted from existing programs, wrapped and integrated into an eBusiness application framework. Doing so avoids the cost and risks of having to develop them from scratch.	The main problem in this approach is the issue of reentrancy. The state of the data contained within a wrapped web service is that of the last caller. Thus, if different processes are using the same service, their data will be mixed.
[13]	Analyzing the Reuse Potential of Migrating Legacy Components to a Service-Oriented Architecture	Their conclusions to the client will helped them avoid a set of mistakes they may have otherwise made. The type of disciplined analysis that they performed appears to have applicability for other organizations that are considering migrations to SOAs.	The Service-Oriented Migration and Reuse Technique (SMART) which is used in this approach has its general applicability to corporation and government agencies that want to migrate to SOA. However, it is not yet fully updated and therefore the update is still on course.
[12]	Feature Analysis for Service-	This approach applied feature analysis in the	Irrespective of it high benefits, this

	Oriented Reengineering	service-oriented reengineering process. They bridge the gap between problem domain and solution domain, by which target services can be identified from legacy systems.	approach is still modestly invasive. Therefore there are still lapses to be corrected.
[11]	An ADM Approach to Reengineer Relational Databases towards Web Services	Their approach has been validated using an industrial application, and particularly the relational database. Their results reveal that not only existing services are discovered, but also many other services for future developments are obtained.	Nonetheless, some future lines arise from this work. Initially only four patterns have been presented, but in reality, more useful patterns exist, so therefore, this work has not yet reach its full potential. Future experiments quite useful
[21]	Turning Web Applications into Web Services by Wrapping Techniques	The problem of migrating the functionalities of existing Web applications towards Service Oriented architectures have been addressed using a black-box wrapping approach. This approach considers a Web application as a special type of 'form-based' system, and proposes to adopt Automaton-based wrappers for transforming the original (non-programmatic) user-oriented interface of the Web application into a programmatic one that exposes the full functionality and data of the application.	However, there is a need to further address the methodological and technological adaptations to this wrapping technique in order to extend the applicability of this approach to other classes of Web applications, such as the 'Rich Internet Applications'.
[16]	A wrapping approach for migrating legacy system interactive functionalities to Service Oriented Architectures	Relative to other wrapping techniques proposed in other literatures, this approach is considered novel because it wraps the whole legacy system, rather than specific and limited parts of its code. Another point of novelty of this approach consists of the idea of implementing the wrapper as an interpreter of a Finite State Automaton that encapsulates the rules of the interaction between the user and legacy system.	In this preliminary experiment there are indications that shows the feasibility of this migration approach that further work is still needed to develop the techniques and tools for aiding the most expensive tasks of the migration process. Meaning the approach is still premature.
[9]	A New Approach to Wrap Legacy Programs into Web Services	Based on the implementation of this approach, the integration cost of legacy systems with modern technologies and programs were duly decreases noticeably, Because the existing tools were used and the black box method which does not contain the source code analysis were also used to expose legacy functions to modern environments. Secondly, the possibility of making changes in Service Bus Class with modern technologies and advances without making any changes on the old system is the another advantage in this approach.	Irrespective of the successes made in this approach, the method is still constrained to Microsoft .Net. therefore, more common programming languages such as Java can be used to further extend this process. Also because of the weak quality of legacy code, there is a need for more qualified code generation in automatic wrapping and executing will increase effectiveness of the method.
[10]	Adapting LMS architecture to the SOA: an Architectural Approach	They developed a services that will provide access to the LMS system from the Moodle interface. These services that provide access to the Moodle core system can also be used from a variety of external applications such as mobile-learning applications. Also, taking into account the existing shortcomings in other SOA adaptation initiatives, the purposed architecture settles them. To do this a layer architecture which considers specifications is proposed. The use of OKI OSIDs and IMS LTI guarantees the portability of the architecture and the bidirectional work.	Looking at this entire proposed system refactoring, more is needed to add to the core functionalities that required the structuring of a clean access layer to the core or internal layer.
[3]	Prerequisites for Successful Implementation of Service-	This research point out and define a basic set of criteria, as a necessary set of prerequisites that	However, been the fact that various organizations has their various operations,

	Oriented Architecture	an organization has to establish if it wants to institute a successful SOA implementation. It went further to show which criteria should be used in this set, and what are the differences and additional prerequisites for some specific domains	this study will not be able to show the entire prerequisites and criteria with excellent assertion.
[22]	A Service Identification Framework for Legacy System Migration into SOA	This study introduces an intensive meta-model that defines uniquely the characteristics of business processes and service types in the way that definitions are mapped. It went further to emphasize the importance of the classification of service types to define service properties.	However, this study failed to expand this approach to automate the service identification process and which is needed to generate service definitions with optimal level of granularity automatically.
[23]	From legacy Web applications to Web Services based applications	This study point out the required challenges of the existing technologies and present the principal technologies meeting these needs, for obtained a high level of interoperability.	This study failed to establish steps for the decomposition of the existing Web applications into Web services without basing their approach in the concept of categorization.
[20]	Migrating a Web-Based Application to a Service-Based System – An Experience Report	This new service-based system is more adaptable to change. Also the production time for isolated small changes has reduced considerably. While the overall cost of migration was less than the estimated cost of re-factoring. Thus SOA provides more benefits than the traditional framework that were previously in existence.	In this study the migration to SOA is driven by a particular system (XYZ), which was in need of migration based on improper functionalities. Therefore, the approach use in this study might not be applicable to other systems. Hence it might require further overview.
[8]	Towards Re-engineering Web Applications into Semantic Web Services	The framework in this study aims at improving development productivity and adoption of Semantic Web Services (SWS). This framework was implemented through a set of tools that reduced the manual intervention of the designer in the reengineering process.	This study still has some amount of limitation considering the industrial perspectives. Therefore, the framework needs to be validated based on a set of industrial complex cases to better integrate and refine the developed prototype tools.
[19]	Research and Application of Migrating Legacy Systems to The Private Cloud Platform with CloudStack	This article expanded and updated the legacy systems of HuaiAn City Flood Control and Drought Prevention Headquarters effectively with the use of open source virtualization server management software cloudstack.	The approach used in this study is limited to a case study and it method might not be applicable to all systems.
[18]	Metamodeling for Assigning Specific Roles in the Migration to Service-Oriented Architecture	This study introduces a modeling paradigm specific for migrating from traditional structures to SOA-based roles, with distinct representations of the source, target and transformation models.	This study fail to extend it scope to model interpreters for the impact assessment of the migration process.
[17]	Migration of Web Application SIMA into Multi-tenant SaaS	This study successfully transform or migrate SIMA to a SaaS for cloud tenants (hosting). The process further conducted customization and configuration of the application including its database architecture and create isolation between the tenants.	This study failed to conduct a comparison study with using shared database with multi-schema approach to seek for the most efficient scheme for the storage and workload of the databases.

5. LIMITATIONS OF THIS RESEARCH

There are three key limitations associated with this research. One of the limitations of this SLR is that, the studies were filtered by a group of selected keywords that were defined within the above sections (section 3.3). Considering that constraint, it is certainly possible that there exist other studies which failed to show up during our searching period. Therefore, it is worthy to admit to the fact that this restriction can cause the search engine to skip other important titles that might be of relevance to this SLR.

The second limitation of our studies is that, our research is based on only two digital libraries (IEEE and ScienceDierct) which are freely accessible to the authors. The free access of the two digital libraries exposed to the authors of this research is possible because of the fact that our host institution (ITS Surabaya) is paying subscription to augment the effort of ongoing research development by student and lecturers. Thus, the amount of digital libraries might be appearing lesser.

Finally, there exist no clear assessment metrics in this research. Nevertheless, this research was carried out within the close supervision of one credible senior researcher and lecturer, with credible experience in the field of software evolution. Hence the result of this SLR can be considered as a sufficient attempt in providing systematic literature review to the research questions highlighted in the area of legacy software system migration to web service-based.

6. CONCLUSIONS

A wide range of research literatures regarding to the migration of legacy systems migration into web service-based have been covered in this work. Many findings were achieve based on data extraction and analysis, which helps the research to be concluded based on two main classification results.

Migration of legacy system is possible at the level of reengineering and can be carried out when the need arises. Based on the analysis of data from this research we discovered that the transformation or migration process can be done using two main approach; (1) by the using a "Method type" i.e. system analysis and code wrapping. (2) By using a "Migration resource" i.e. Web and Non-Web.

The advantage of the system analysis method is the explicit breakdown of the system modules and precision of the migration a legacy software into web service-based to achieve up-to-date systems. The disadvantage of this method is the amount of effort needed to produce the required results. The advantage of the code wrapping method is that, it is usually very quick to obtain the migration results. The disadvantage of code wrapping method is that, the result is highly dependent on the accuracy of the code wrapping algorithm, hence there is a high possibility for errors.

Furthermore, improvement of this research can be done by increasing the scope of the digital libraries and search queries to obtain more literatures to be included in the next studies.

7. REFERENCES

[1] Wikipedia/Web_service,"https://en.wikipedia.org/wiki/Web_service."

[2] Wikipedia,"https://en.wikipedia.org/wiki/Legacy_syste."

[3] Sandro Geric and Neven Vrcek, "Prerequisites for Successful Implementation of Service-Oriented Architecture," Proc. ITI 2009 31st Int Conf Inf. Technol. Interfaces June 22-25 2009 Cavtat Croat., 2009.

[4] Duen-Kai Chen, "Systematic Review of Applying Service Oriented Architecture in Networking," IEEE Conf., 2010.

[5] Khubaib Amjad Alam, Rodina Binti Ahmad, and Maria Akhtar, "Change Impact Analysis and Propagation in Service Based Business Process Management Systems," IEEE Conf., 2014.

[6] John Hutchinson, Gerald Kotonya, James Walkerdine, Peter Sawyer, Glen Dobson, and Victor Onditi, "Evolving Existing Systems to Service-Oriented Architectures: Perspectives and Challenges," IEEE Int. Conf. Web Serv. ICWS 2007, 2007.

[7] Abhishek Sharma, Theodore D Hellmann, and Frank Maurer, "Testing of Web Services – A Systematic Mapping," IEEE Eighth World Congr. Serv., 2012.

[8] Bouchiha Djelloul, Malki Mimoun, and Mostefai Abd El Kader, "Towards Re-engineering Web Applications into Semantic Web Services," Int. Arab J. Inf. Technol., vol. 6, pp. 1–6, Oct. 2009.

[9] Saeed.Parsa and Leila.Ghods, "A New Approach to Wrap Legacy Programs into Web Services," 11th Int. Conf. Comput. Inf. Technol. ICCIT 2008, 2008.

[10] Miguel Ángel Conde González, Francisco José García Peñalvo, María José Casany Guerrero, and Marc Alier Forment, "Adapting LMS architecture to the SOA: an Architectural Approach," Fourth Int. Conf. Internet Web Appl. Serv., 2009.

[11] García-Rodríguez de Guzmán, I, Polo, and Piattini, "An ADM Approach to Reengineer Relational Databases towards Web Services," IEEE, 2007.

[12] Feng Chen, Shaoyun Li, Hongji Yang, Ching-Huey Wang, and William Cheng-Chung Chu, "Feature Analysis for Service-Oriented Reengineering," Proc. 12th Asia-Pac. Softw. Eng. Conf. APSEC'05, 2005.

[13] Grace Lewis, Edwin Morris, and Dennis Smith, "Analyzing the Reuse Potential of Migrating Legacy Components to a Service-Oriented Architecture," Proc. Conf. Softw. Maint. Reengineering CSMR'06, pp. 1 – 9, 2006.

[14] Harry M. Sneed, "Integrating legacy Software into a Service oriented Architecture," Proc. Conf. Softw. Maint. Reengineering CSMR'06, 2006.

[15] Jianzhi Li and Hongji Yang, "Towards Evolving Web Sites into Grid Services Environment," Seventh IEEE Int. Symp. Web Site Evol. WSE'05, pp. 1 – 8, 2005.

[16] Gerardo Canfora, Anna Rita Fasolino, Gianni Frattolillo, and Porfirio Tramontana, "A wrapping approach for migrating legacy system interactive functionalities to Service Oriented Architectures," J. Syst. Softw. 81 2008 463–480 Sci., 2008.

[17] Ekasari Nugraheni, "Migration of Web Application SIMA into Multi-tenant SaaS," IEEE, 2013.

[18] Anca Daniela Ionita and Stefan Aureliu Radulescu, "Metamodeling for Assigning Specific Roles in the Migration to Service-Oriented Architecture," Third Int. Conf. Emerg. Intell. Data Web Technol., 2012.

[19] Bin Cai, Feng Xu, Feng Ye, and WenhuanZhou, "Research and Application of Migrating Legacy Systems to The Private Cloud Platform with CloudStack," Proceeding IEEE Int. Conf. Autom. Logist. Zhengzhou China August, 2012.

[20] Pushparani Bhallamudi, Scott Tilley, and Arunesh Sinha, "Migrating a Web-Based Application to a Service-Based System – An Experience Report," IEEE, 2009.

[21] Giusy Di Lorenzo, Anna Rita Fasolino, Lorenzo Melcarne, Porfirio Tramontana, and Valeria Vittorini, "Turning Web Applications into Web Services by Wrapping Techniques," IEEE, 2007.

[22] Saad Alahmari, Ed Zaluska, and David De Roure, "A Service Identification Framework for Legacy System Migration into SOA," IEEE Int. Conf. Serv. Comput., 2010.

[23] Naoufel Kraiem and Zuhoor Al-Khanajari, "From legacy Web applications to Web Services based applications," 1st Int. Conf. Electr. Inf. Technol. ICEIT', 2015.

8. AUTHOR PROFILE

Dr. Siti Rochimah received her PhD degree from Universiti Teknologi Malaysia in 2011. She is currently the Head of Software Engineering Laboratory, in the Department of Informatics, Institut Teknologi Sepuluh Nopember (ITS). Surabaya, Indonesia. Her research interest includes software quality and testing, software evolution, and software methodology.

Alhaji Sheku Sankoh is currently working towards a Masters degree in Informatics Engineering (M.Sc.) at the Department of Informatics, Institut Teknologi Sepuluh Nopember (ITS) Surabaya, Indonesia. His M.Sc. programme will be hopefully completed by January 2016 and the Certificate awarded will be in March 2016. He was awarded a Bachelor of Science degree in Business and Information Technology (B.Sc.) in 2012 from Njala University (Sierra Leone). In July 2010, he was certified as a Microsoft Certified Technology Specialist (MCTS SQL Server). His research interests includes software engineering practice and computer vision & Image processing.

9. APPENDIX

Table 9 Result of data extraction

No	Cited Ref	Title of paper	Migration resource	Method type	Method used	Dataset	Tools
1	[8]	Towards Re-engineering Web Applications into Semantic Web Services	Web Applications (Web)	Wrapping source code	The proposed framework allows semi-automatic reengineering of Web applications toward SWS. It receives as input the source code of the traditional Web application (HTML + PHP, JSP, or others), a relational database schema, and possibly existing OWL ontologies inherited from the Semantic Web structure of the application.	HTML + PHP, JSP, relational database schema and OWL ontologies	OntoWeR&WSDL description generation tool
2	[9]	A New Approach to Wrap Legacy Programs into Web Services	Legacy Programs written in languages such as C or Pascal into Web Services. (Non-web)	Analysis of existing system & Wrapping	Creating a Service Bus Class which consist of two stages of analyzing and web service wrapping of legacy code. In the analysis stage reverse engineering techniques are applied to extract the functions within a given code. The extracted Methods are then presented to the user to select a number of them. In the wrapping stage, a class comprising the selected methods is created. This class is then scanned by our web service wrapping tool, to convert the class methods into web services.	Any Program written in C or Pascal programming languages	Web service wrapping tool. By using Visual Studio .NET, wrapping legacy program into Web Services can be done.
3	[10]	Adapting LMS architecture to the SOA: an Architectural Approach	From Moodle Learning Management Systems (LMS). (Web)	Wrapping source code	Moodle DFWS Architecture: From the architectural point of view Moodle is based on a model-view-controller controller. This pattern is common in interactive applications that evolve rapidly. This architecture is complemented by other patterns that provide flexibility to the system.	Web Service Library	-Rapper (APIs)
4	[11]	An ADM Approach to Reengineer Relational Databases towards Web Services	Relational databases at a model level (Non-web)	Analysis of existing system	Uses an Architecture Driven Modernization (ADM) focuses the reengineering process using models as first order artefacts. In this way, and following the chain legacy system-to-abstract representation-to-new system	Relational Database Attribute and Platforms.E.g. (SQL-92 databases) metamodels.	PSM, PIM, WSDL. MDA, ADM.
5	[12]	Feature Analysis for Service-Oriented Reengineering	Legacy System (Non-web)	Analysis of existing system & Wrapping	To construct SOA, the Service-Oriented Design (SOD) paradigm attempts to explore a hybrid process. A top-down process is employed most of the time in SOD, which involves in domain analysis, business process analysis and other forward method. However, a bottom-up analysis of these systems also has to be conducted in order to investigate wrapping and refactoring strategies	Library MIS	The implementation of identified services can be generated by the WSW-like tool.
6	[13]	Analyzing the Reuse Potential of Migrating Legacy Components to a Service-Oriented Architecture	Legacy systems or components (Non-web)	Analysis of existing system	Based on the experience illustrated in the case study, we developed an approach for gathering the necessary information and identifying the risks for the migration effort in a more systematic way. This method is the Service-Oriented Migration and Reuse Technique (SMART). SMART was developed by integrating OAR, architecture reconstruction practices with the ARMIN tool, and a detailed analysis of the requirements of the target SOA.	DoD command and control (C2) system (Developed with C++)	SMART, OAR & ARMIN tools
7	[14]	Integrating legacy Software into a Service oriented Architecture	Legacy Software Systems (Non-web)	Wrapping source code	There are three basic steps required to create web services from legacy code. Salvaging the legacy code Rapping the salvaged code and Making the code available as a web service.	A calendar function extracted from the legacy software of a Swiss bank. (COBOL)	SoftWrap
8	[15]	Towards Evolving Web Sites into Grid Services Environment	Web Sites (Web)	Wrapping source code	Grid services-oriented reengineering approach to create stateful resources from conventional HTML Web sites, which applies hierarchical cluster and wrapper techniques to extract and translate Web sites resources.	Hypertext Markup Language (HTML) pages.	Web Service Resource Framework (WSRF)
9	[16]	A wrapping approach for migrating legacy	Legacy Systems (Non-web)	Wrapping source code	Wrapper-based architecture for the migration of a legacy system to Web Services. (A black-box modernization approach for exposing interactive	Legacy Software Systems	Wrapper

		system interactive functionalities to Service Oriented Architectures			functionalities of legacy systems as Services).		
10	[17]	Migration of Web Application SIMA into Multi-tenant SaaS	Re-engineering of SIMA (Web)	Analysis of existing system	A framework that is used to perform the migration process of SIMA application into multi-tenancy model.	SIMA Web Application	Business logic Configuration
11	[18]	Metamodeling for Assigning Specific Roles in the Migration to Service-Oriented Architecture	Legacy Systems (Non-web)	Analysis of existing system & Wrapping	Metamodeling the role migration. • the source, e.g. a legacy application, or enterprise data; • the target, e.g. a SOA-based system, or a non-SQL database; • the transformations performed for reusing the source and developing the desired target, e.g. wrapping, reengineering, refactoring, restructuring etc.	Design	Metamodeling the role migration
12	[19]	Research and Application of Migrating Legacy Systems to The Private Cloud Platform with CloudStack	Legacy Systems (Non-web)	Analysis of existing system	We use vm ware p2v technology (migration from the physical machine to virtual machine) to deal with legacy systems, it can migrate the physical server's operating system, application software and data to the virtual servers managed by VMM (Virtual Machine Monitor)	Citrix Systems	Cloudstack
13	[20]	Migrating a Web-Based Application to a Service-Based System – An Experience Report	Web-Based Applications (Web)	Others (Experience Report)	Method is based on Business and Technical Analysis of existing legacy systems. (Three main ROI benefits were experienced by the migration: aligning business needs with IT functionality, keeping data consistent, and reducing cost.)	A report	System analysis.
14	[21]	Turning Web Applications into Web Services by Wrapping Techniques	Web Application (Web)	Wrapping source code	The Automaton conceptual model. This process includes four consecutive steps: 1. Selection of the Web application functionality to be turned into a Web service 2. Reverse Engineering of the Web application User Interface a. identification of execution scenarios b. characterization of execution scenario steps 3. Interaction Model design a. Evaluation of alternative modelling solutions b. XML-based design of the model 4. Wrapper Deploy and Validation	Web Pages	Wrapper Logical Architecture
15	[22]	A Service Identification Framework for Legacy System Migration into SOA	Legacy Systems (Web)	Analysis of existing system	Their research proposes a transformation method between Unified Modeling Language (UML) to Business Process Modeling Notation (BPMN), i.e. they transform UML activity diagrams to BPMN diagrams. They also adopt an SOA meta-model that defines service types along with the semantics of models.	UML	we used the IBM WebSphere Business Modeler tool
16	[23]	From legacy Web applications to Web Services based applications	Legacy Web Application (Web)	Analysis of existing system	The proposed approach is summarized as follows: they first propose a method of classification of existing Web-applications. Then, they present a step-by-step decomposition of these Web-applications to identify a high level of granularity and reuse of logics functionality. Next, they propose a software of model transformation and we briefly present evaluation of existing software of model transformation according to offer functionality	Web Pages	DYNAMIC GENERATION FACET And ADAPTATION FACET
17	[3]	Prerequisites for Successful Implementation of Service-Oriented Architecture	Benchmark (Non-web)	Analysis of existing system	This article analyses a necessary set of prerequisites which should be fulfil by organization attempting to implement SOA. Since SOA implementation requires significant organizational changes. Such set of prerequisites should comprise not only the technical prerequisites, but should take into account data architecture, system integration and compatibility, security and legislative requirements, business requirements and organizational prerequisites.	Enterprise software	CMMI SOAMMs SIMM