Legacy System Re-engineering Using Service Oriented Architecture

Solomon Tadege
HiLCoE, Software Engineering Programme, Ethiopia
SolTech IT Solutions, Addis Ababa, Ethiopia
solomontadege@gmail.com

Nassir Dino
HiLCoE, Ethiopia
nassir@hilcoe.com.et

Abstract

Legacy software is in use by many organizations. Even though organizations commonly get a great deal of value from their legacy systems, it is becoming mandatory to consider changes for most legacy systems to be modernized in light of business and technology agility. To this effect, in order to bring software systems up-to-date with current technologies, businesses have two choices. Either develop new software to replace the existing systems, or re-engineer existing systems that utilize state-of-the-art technologies. Among the architectural options for re-engineering legacy systems, this paper proposes using Service Oriented Architecture (SOA). SOA based Re-engineering can save considerable time and cost compared to developing new systems from scratch. This also helps to avoid creation of new bugs.

To this effect, a numbers of SOA models are examined and their advantages and disadvantages evaluated in order to see their validity in the current re-engineering implementation environments and possible need for changes. Based on the result of the assessment made, a workable SOA Architectural Method, development cycle and roadmap are framed that will respond to the requirements of engineering strategy to modernize legacy software to SOA platform, which is the main goal of this paper.

Keywords: SOA; Legacy System; Model; Re-engineering

1. Introduction

One of the many reasons for the continuity of a business is its ability to anticipate & adapt to change. Up-to-date business and technology capabilities guarantee to meet the enterprise's target and beyond. In lieu of this, a change of the legacy system to new technology is crucial to a business enterprise growth, modernity, competency and survival, among others.

Legacy systems generally consist of invaluable assets with embedded business logic representing many years of coding, development, update, and modification. However, legacy systems are often undocumented, tightly coupled, and relatively closed and inflexible. In most cases they were developed independently without considering a consistent underlying architecture, resulting in overlapping and redundant functionality and data repository [1]. Modernization of these legacy systems initiates the need for re-engineering.

Re-engineering is one among the legacy system modernization strategies which is "a systematic starting over and reinventing the way a firm, or a business process gets its work done" [2]. Re-engineering is mostly used in the context where a legacy system is involved. Legacy software often needs to be rewritten or "re-engineered" to operate on a new computing platform [3].

In Software Engineering, a SOA is a set of principles and methodologies for designing and developing software in the form of interoperable services [4]. These services are well-defined business functionalities that are built as software components (discrete pieces of code and/or data structures) that can be reused for different purposes.

In moving to legacy systems modernization a number of SOA Models are in use. Widely known among others are Service Oriented Modelling, SOMF, SOA Development Methodology, SOA Methodology, and IBM RUP/SOMA. These are the models investigated and used in this paper. Though useful for their desired requirements they, however, do not fully handle Legacy System Re-engineering using SOA as a necessity. In such environments it is required to get the best of the old and the new, to take pace with the fast growing business and technology while leveraging the existing assets.

The purpose of this paper is, therefore, to maximize the benefit of SOA in re-engineering legacy systems and propose a comprehensive SOA re-engineering model that encompasses using SOA re-engineering strategy Architectural Method, legacy to SOA re-engineering life-cycle and a roadmap for re-engineering legacy systems using SOA, i.e., schematic representation with details on how to implement it with phases, steps, activities, and deliverables.

2. Literature Review

In order to crop up a model that supports Legacy System Re-engineering using SOA in the absence of one to support, literary works on SOA, Legacy system, Re-engineering, vendor SOA user organizations experiences and research works on Service Oriented Enterprise Architecture Framework (SOEAFs) are referenced and evaluated to get clear view of their contribution towards the desired goal.

The purpose of SOA is to realize the goals of service oriented computing. In order to establish SOA successfully, three components are needed: infrastructure, architecture, and processes [5]. Also, technical platform, such as web services and cloud are needed to form the infrastructure.

SOA also incorporates Important Elements such as SOA Quality of Service, SOA Service-level Agreement, SOA Monitoring and Management, and SOA Return on Investment.

Importantly, SOA involves governance by which a SOA system is controlled and administrated. The governance includes standards and policies that govern the design, building and implementation of SOA solution and also the policies that need to be realized during runtime. It also comprises QoS policies on performance, security, and transactions and regulatory, business, audit and infrastructure policies [5].

SOA is also supported by enabling technologies which are Web Services and Cloud Computing. Due to the loosely coupled nature of these technologies, SOA has the potential to greatly increase vulnerability in composite applications. Thus, service security needs consideration in the implementation of SOA. According to Woolf [6], security is a difficult but necessary component for any application. Functionality needs to be limited to authorized users and data needs to be protected from interception.

Legacy software systems are those systems that have become obsolete with time, though still remain valuable assets for a company because of their made-to-measure approach to meeting the company's needs. At the outset, with the fast pace of the development of modern technologies and business growth, it is sometimes difficult to remain abreast in terms of the company's software and infrastructure. Systems already implemented require upgrade due to changes in requirements, technology, vulnerability, etc.

The Gartner Group defines a legacy application or system as "an information system that may be based on outdated technologies, but is critical to day-to-day operations" [1]. This definition highlights the importance of legacy systems, but also the challenges for transformation towards new technologies.

Software reengineering is described by Chikofsky and Cross [7] as "The examination and alteration of a system to reconstitute it in a new form". Less formally, re-engineering is the modification of a software system that takes place after it has been reverse engineered, generally to add new functionality, or to correct error.

Software Re-engineering, as a principle, provides four general modernization strategies that are also applicable for modernizing legacy systems using SOA. These are Replacement, Redevelopment or Reengineering, Wrapping, and Migration [8]. Reengineering is the concern of this paper for it handles service and architectural recovery.

Fundamental concepts, principles, enabling technologies, and important elements serve as cornerstones in formulating a new model. Also, modernization strategies in engineering software systems have helped in comprehending the properties of the selected re-engineering strategy.

3. Related Work

SOA Enterprise Architecture Framework (SOAEAF) is the dependencies between SOA and Enterprise Architecture (EA) in such a way that SOA proposes important principles on applications. SOA

depends on EA and its contents and governance in order to succeed. From EA perspective, SOA is a basic concept that enriches the architecture modelling of EA at all architectural layers [9]. In effect, SOA could serve as a mediator between the different elements of an EA [10] and is intended to closely align IT and business.

In the process of evaluating best practices from IT vendors [1, 11], SOA based user organizations experiences [12] and elements identified from SOAEAFs [13, 14] research proposals referenced, characteristics needed in understanding and evaluating the models have been identified. The experience of IT Vendors and user organizations contributed in the need of management involvement and governance in SOA. Also, characteristics defined in SOAEAFs have helped identify selected major transferable phases, steps, activities and deliverables to the new model.

The following seven known SOA Models are evaluated to check their applicability to implement Legacy System Re-engineering using SOA.

- *Service-Oriented Modelling* [15]: focuses on modelling strategies rather than being concerned with source code and supports the activities of service.
- *SoaML* [16]: modelling and design to fit into an overall model-driven development approach and detailed programming algorithms.
- *SOMF* [17]: presents corresponding models whose language notation guides practitioners in designing, architecting, and supporting a service ecosystem.
- *SOA Development Model* [18]: a weak model in defining steps and activities delineated, but with full SOA lifecycle phases. It served as base for defining phases of the cycle in its complete set-up in the proposed model.
- *SOA Re-engineering* [19]: concentrates on three crucial aspects, i.e., understand the legacy system, decomposition of legacy system, and organization of the migration process in a cost effective manner.
- *SMART* [20]: proposes to select from the initial list of candidate services those who are good candidates, specify by defining for each service the inputs and outputs, and the information quality of service expected.
- *IBM RUP/SOMA* [21]: consists of four phases: business transformation analysis, identification, specification, and realization of services.

As observed in the brief description of the models referenced in here, it can be concluded that none of them fully responds to the requirement of Legacy System Re-engineering using SOA. Thus, the need for developing a new model remains the main concern of our paper. It should, however, be noted that models contribute in forming Software Engineering principles based skeletal life cycle and help derive additional characteristics essential in the new model building.

In order to propose a workable model in the absence of a complete one, characteristics involved in SOA fundamentals, enabling technologies, and important elements with essential characteristics from SOA frameworks serve as basis of evaluating the available models. These, added with supplementary characteristics identified from the models themselves, serve to institute phases, steps, activities and deliverables required to the proposed model. Summary of the steps involved to develop the model is depicted in Figure 1.

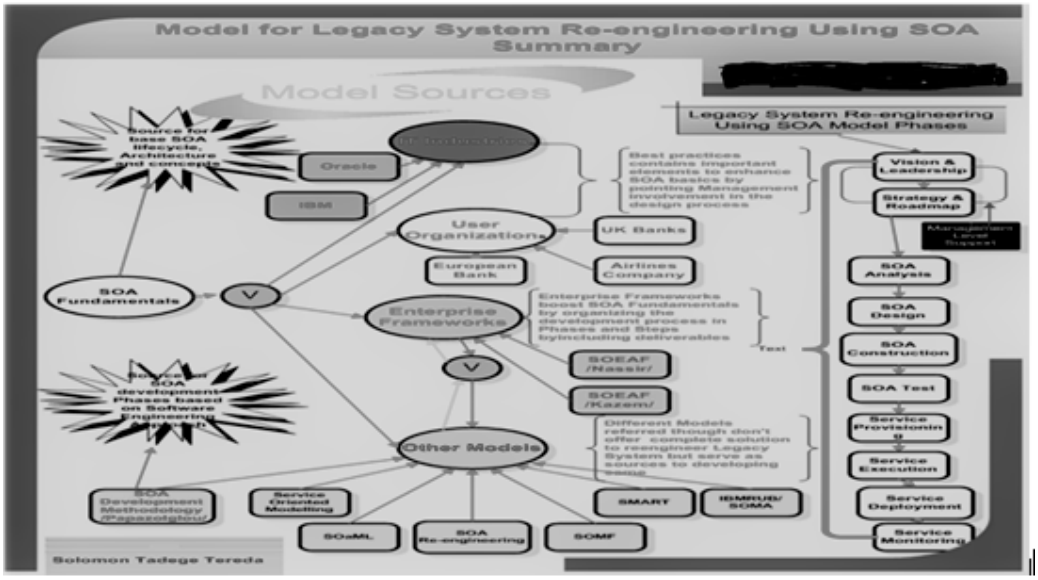

Figure 1: Legacy System Re-engineering Using SOA Model Building Structure

4. The Proposed Solution

4.1 The Selected Strategy

In general while choosing a strategy, diverse aspects should be considered depending on the project under reflection by examining the set of pros and cons in each. Other possibilities might include the type of company that plans to develop SOA based on higher risk and a higher cost than replacing them entirely with a new SOA model.

Thus, selecting a strategy should not be a rush decision to solve a given software problem but a task with thorough thought.

In case of this paper, however, since the very idea is that of proposing a model to handle re-engineering legacy system using SOA, the selected modernization strategy is Re-engineering.

4.2 The Proposed SOA Architectural Method

The proposed SOA Architectural Method has its source from basic SOA and upgraded to fit into Re-engineering a legacy system using SOA by including updated service lifecycle and the prevailing enabling technologies such as Cloud and Web services. Also, Quality of Service aspect in the traditional architecture is changed to Domain level end-to-end guideline since quality of service itself is one of the components as emphasized in one of the SOA Enterprise Frameworks. The Architectural Method for the proposed model is shown in Figure 2.

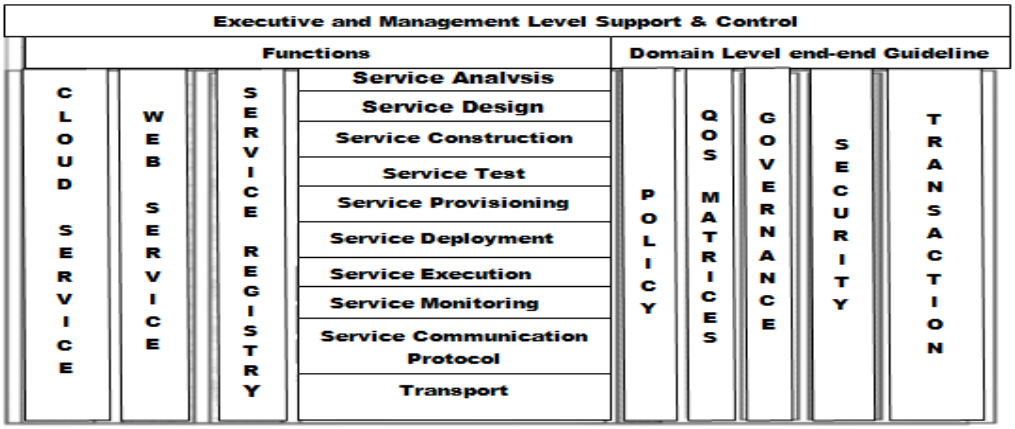

Figure 2: Elements of Service Oriented Architecture

4.3 The Proposed SOA Lifecycle

SOA Service development Lifecycle for legacy system re-engineering is defined as a process involving classified nine phases. The Lifecycle is believed to serve as a guideline in the Re-engineering of legacy based systems using SOA.

Figure 3 shows the Legacy System Re-engineering to SOA development lifecycle involving all nine phases, i.e., Management support and control, SOA Analysis, SOA Design, SOA Service Construction, SOA Service Test, Service Provisioning, Service Deployment, Service Execution, and Service Monitoring whose details are worked under the proposed SOA Roadmap.

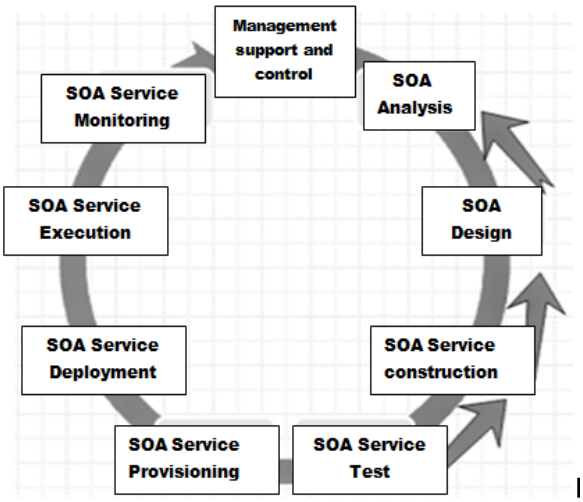

Figure 3: Legacy Re-engineering to SOA Development Lifecycle

4.4 The Proposed SOA Roadmap

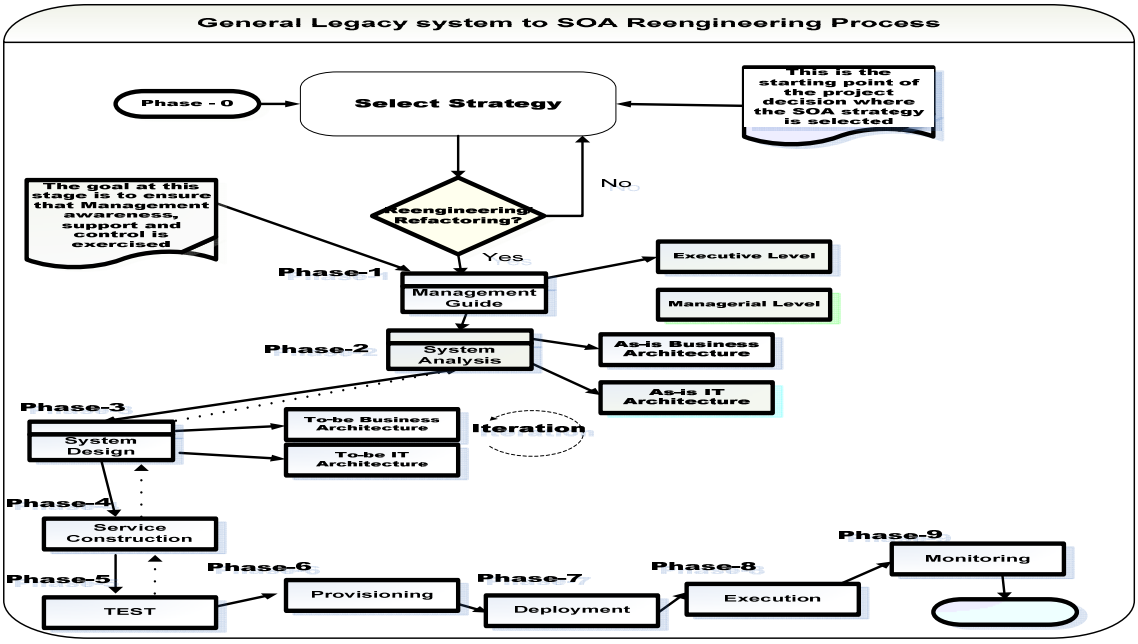

Figure 4: High Level Legacy to SOA Implementation Roadmap

The Roadmap provides more detailed graphic supported view of phases defined under SOA lifecycle for the realization of the established phases as shown in Figure 4. The summary of high level representation and processes showing all phases, steps, activities and deliverables is presented in Table 1.

Table 1: Legacy to SOA Re-engineering Roadmap Summary

Phases	*Steps*	*Activities*	*Deliverables*
Phase 1: Management Support and Control	Vision & Leadership	Ensuring IT and Business alignment, commitment to implement SOA, resource identification, business case: objective, alternatives, financial analysis, handling, project risk & assumptions	Business case: objective, alternatives, financial analysis, Rewards for success tracking document

Phases	*Steps*	*Activities*	*Deliverables*
	Strategy & Roadmap	Crafting Vision and Mission Statement of SOA Governance, Define Project SOA strategy, finalise the scope of the project, Develop project management plan, define principles, policies, standards that support vision and mission involving business and IT representative in decision making, setting goals & objectives of implementation	Project organizational structure, Scope statement, Project Management Plan (PMP), Set of policies, Set of Governance (if centrally set)
Phase 2: SOA Analysis	Business As-Is Architecture	Extracting organization's business strategy components, Identifying detail business structure and services, adopting legacy system, recovering legacy system architecture, analyzing legacy code and documentation, extracting as-is business services, identifying and defining processes, defining candidate services, defining application's information systems architecture, analyzing findings from the above activities	Business conceptual model and architecture, application logical architecture, IT and business strategy components, business function model, process model, As-Is business services, As-Is application's information systems architecture
	IT As-IS Architecture	Identifying technical IT assets, identifying non-technical IT resources, examining security set-up and architecture, identifying IT infrastructure needs, examining data model, analyzing findings from the above activities	Technical IT assets, non-technical IT resources security set-up and architecture, IT infrastructure data model
Phase 3: SOA Design	Business To-be Architecture	Identifying To-Be business services, updating and redefining process model, updating and redefining application's information systems architecture, updating and redefining To-Be business services (core, non-core and external services, updating and redefining logical architecture, implementing services identified, considering SOA governance: business and application architecture, focusing on non-functional requirements, performing iteratively discovering and providing services and assure its compliance with the vision	Conceptual, services, information, application architectures, governance, security policy, service and technology usage, compliance
	IT To-be Architecture	Updating and redefining Data architecture and model, Actualizing physical Architecture Testing Application services, Checking Infrastructure services, Ensuring security implementation in and out	Updated data architecture, updated data model, application services, infrastructure services, physical architecture design, technology usage and security infrastructure
Phase 4: SOA Service Construction	Service & Process Realization	Defining the service interface description, defining the service implementation description, implementing services, implementing process realization scenario identified, modifying existing services	Service definition interfacing description implementation description
	Web & Cloud Implementation	Developing the Web services implementation, implementation of cloud services	Web Services and Cloud Services
	Business Process Realization	Implementing process realization scenario identified, evaluating diverse business process realization scenarios in terms of costs, risks, benefits and return of investment in accordance with business requirements and priorities	Business process realization scenarios, cost scenarios, risk scenarios, benefits and RoI
Phase 5: SOA Service Test	Dynamic Testing	Functional testing, performance and stress testing, assembly testing, interface testing	Functional test report, performance test report, assembly test report, stress test report, interface test report
	Other Testing Elements	Testing network congestion, testing security, testing install-ability, testing compatibility, testing usability, upgrades testing	Network congestion test report, security test report, install-ability tests report, compatibility tests report, usability tests report, upgrades tests report, compatibility tests report
Phase 6: Service Provisioning	Service Governance	Reviewing findings using business gap analysis result to ensure: the right types of services have been selected, all requirements for new services have been identified and so forth, service conforms to enterprise specific or government mandated privacy rules, service implementation compromise enterprise-specific intellectual property, and so on	Conformance report, governance review report

Phases	*Steps*	*Activities*	*Deliverables*
	Service Certification	Using properties predicting or achieving some end-system properties, such as performance, safety, scalability and so, predicting the "values" of end-system properties, measuring by end system quality attributes	Service Certification Report, End System Report
	Service Billing Strategies	Ensuring quality of service plays a high role of importance in service billing, ascertaining service is offered for a price	Service billing strategy, service billing guideline
Phase 7: Service Deployment	Deployment House Keeping	Scrutinizing gap analysis output, integrating information systems, applications, implementing application, use and maintenance, implementing agreements (contract, user acceptance), adopting technical standards adoption, implementing changes & compliance procedures, establishing service directory, deploying maintenance service, considering service level agreement, scrutinizing budgeting, ensuring QoS use	Technical documents, security and infrastructure, service description, interface, agreement (contract, user acceptance), QoS matrices, change management guideline, version control guideline
	Rollout	Rolling out new processes to all the participants, other enterprises, applications and other processes	Rolled out system
Phase 8: Service Execution		Ensuring that the new process is carried out by all participants; Ensuring All services are fully deployed and operational, Enabling service requester can find the service definition and invoke all defined service operations, ensuring the run-time functions include: static and dynamic binding, service interactions as a function of Simple Object Access Protocol (SOAP), serialization/de-serialization and messaging and interactions with back-end legacy systems (if necessary)	Execution report
Phase 9: Service Monitoring		Service level measurement and monitoring is the continuous and closed-loop procedure of measuring, monitoring, reporting and improving the quality of service of systems and applications delivered by service-oriented solutions. Continuous evaluation of service level objectives and performance, enabling a set of QoS metrics is gathered on the basis of SLAs, monitoring workloads need and conducting periodic audit, change management, version and asset control, monitoring project progresses through architecture reviews identification, technical standards adoption, compliance.	Progress report, workload report, QoS report, SLA report, QoS matrices, workload report, audit report, change control report

5. Validation using Case Study

In order to investigate the applicability of the proposed model, three organizations, which are considered as high level IT users, are selected. All the organizations are service based. Two are from the banking industry and the other one is an airlines company.

The software development practices have revealed as these organizations use mainly package software and locally developed limited applications.

Package software are fully replaced by new ones whenever changes are required. In here legacy system and improvement analysis is made by the user to prepare specification. All other phases are handled by the vendor except user participation in testing, deployment and execution. Service monitoring is the users' responsibility. Needless to mention, all phases are involved in package software.

6. Conclusions and Future Works

The result of an increasingly competitive business environment and the inherent problems in the legacy system has required and forced organizations to look for ways to minimize cost and time. To this end, SOA has been selected and implemented as one of the architectural platforms selected by numerous organizations.

SOA is established on well-grounded core values that encompasses business value, strategic goals, interoperability, shared service, flexibility, design principles, patterns and evolutionary refinement. As added strength SOA has its own design principles and pattern. Also embodied in SOA are SOA Security, Quality of Service, Service Level Agreement and Return on Investment and Governance.

Thus, adhering to the main concern of proposing a workable model applicable to Legacy System Re-engineering using SOA, a new model is developed having its own SOA Architecture, life cycle supported by detailed schematic Roadmap. The model is developed by upgrading and blending works done in SOA fundamentals, SOA user Organizations experience, best practices, Enterprise Architecture Frameworks, and SOA Models.

The following are future works in order to successfully utilize all modernization SOA strategies.

- All inclusive SOA Model that embraces Legacy System re-engineering using SOA model that supports all SOA strategies which are Re-engineering, Migration Wrapping and Replacement.
- SOA Modeling Tool and Language upgrade to address all the requirements of the phases involved.

References

[1] Calvin Lawrence, "Adapting Legacy Systems for SOA", IBM, 2007.

[2] Michael Hammer and James Champy, Reengineering the Corporation, 1993.

[3] Asit Kumar Gahalaut, *et al*., "Reverse Engineering and Design Recovery", International Journal of Engineering Science and Technology Vol. 2, No., 2010.

[4] Open Group Standard: SOA Reference Architecture (C119), The Open Group, December 2011.

[5] Bartolini, C. *et al*., "Whitening SOA Testing", ACM, 2009.

[6] Woolf, Bobby, "Introduction to SOA Governance", ISSW Web Sphere, IBM, 2007.

[7] Chikofsky, E. and Cross, J., "Reverse Engineering and Design Recovery: Taxonomy", IEEE Software, 7(1):13-18, 1990.

[8] 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, March 2010.

[9] Chung, Sam and Lee, Yun-Sik, "Reverse Software Engineering with UML for Web site Maintenance", First International Conference on Web Information Systems Engineering, WISE, Vol. 2, 2000.

[10] Wikipedia, "Rational Software", http://en.wikipedia.org/w/index.php?title=Rational Software&oldid=55820113, Accessed on September 2018.

[11] Krupi Patel and Leena Ragha, "Survey and Analysis on Migration of Legacy System to Service Oriented Architecture".

[12] Souiou Wafa and Bounour Nora, "Migration of Legacy Systems to Service Oriented Architecture", University of Guelna, Algeria, 2013.

[13] Nassir Dino, Awel Dico, Dida Midekso and Worku Alemu, "Case Study of Service Oriented Enterprise Architecture Framework", David Publishing, 2012, March 25, 2012.

[14] Mohammad Kazem Haki, "Service Oriented Enterprise Architecture Framework", University of Lausanne, Switzerland, 04 February 2015.

[15] Michel Bell, "Service-Oriented Modeling: Service Analysis, Design, and Architecture", John Wiley & Sons, Inc, 2008.

[16] OMG, "Service Oriented Architecture Modelling Language (SoaML)", 2008.

[17] Methodology Group, "Service-Oriented Modelling Framework (SOMF)".

[18] Papazoglou, Mike, van den Heuvel, Willem-Jan, "Service-Oriented Design and Development Methodology", International Journal of Web Engineering and Technology.

[19] Wu, L., Sahraoui, H., Valtchev, P., "Coping with Legacy System Migration Complexity", In Proceedings of the 10th IEEE International Conference on Engineering of Complex Computer Systems, pp. 600–609, 2005.

[20] Grace, A., Lewis, J., Morris, E., Dennis, B., Smith, Simanta, "SMART", Carnegie Mellon University, June 2008.

[21] Ali Arsanjani, "Service-Oriented Modeling and Architecture", IBM, December 2014.