A Comparative Review of Migration of Legacy Systems

A, Sivagnana Ganesan
Department of Banking Technology
Pondicherry University
Puducherry, India

T. Chithralekha
Department of Computer Science
Pondicherry University
Puducherry, India

Abstract— **The Modernization of Legacy System is made appealing by several features of SOA in today's world and migration of applications to the Cloud. In order to carry out a comparative review on the decades old migration approaches are classified in to Early Migration Approaches, SOA Migration Approaches and Cloud Migration. A comparative review is presented among this categories to depict the migration requirements. This paper also provides the research opportunities in the context of migration of legacy systems focusing on Legacy System Understanding, Study of Artefacts and their Relationships**.

Keywords— Migration; SOA; Legacy Systems; Software Evolution

I. INTRODUCTION

Several approaches towards migration of monolithic and procedural legacy systems to varied architectural environment of distributed architectures which inter-alia contains client server architectures, web based and service oriented architectures have been presented by the researchers over the years.

Migration is an offspring research in Software Engineering which is almost three decades old and numerous publications have emerged in many topics in the migration domain with focus areas of code migration, architecture migration, and case study on migration and effort estimation on migration. The maturity of the research in this area is reflected by the various survey and some of the definitions on migration are as follows:

Migration is the passage of a current operating environment of a system to another usually the best and can range from single systems to multiple systems or applications, the transition can be to a new hardware or software or both ensuring continuity of operations [1].

Migration in IT means the move to a new technical environment, mostly for business reasons and for fulfilling (new) non-functional requirements [2].

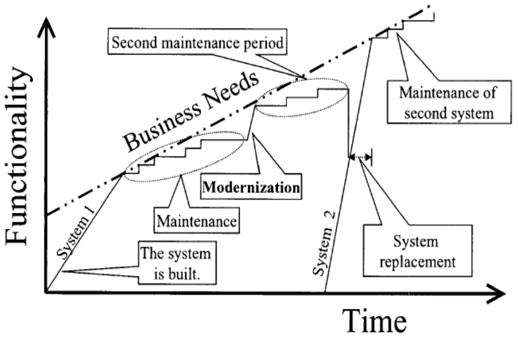

Fig. 1. Information System Life Cycle [3]

Software Migration is moving a system into a new technical environment potentially involving data management, communication and programming environment [2].

Migration is one among the Information system modernization strategies while the other strategies are Replacement, Reengineering and Wrapping [3][4]. In Information system Life Cycle Modernization forms an important phase as depicted in Fig. 1 [3]. This paper intends to review the migration approaches of legacy systems based on the roadmap. Section 2 describes about the road map and its classification, Section 3 describes about the migration approaches and its comparison, Section 4 is about research opportunities and Section 5 is about Conclusion.

II. MIGRATION APPROACHES OF LEGACY SYSTEMS – A COMPARISION

The approaches in Migration of Legacy Systems have evolved over time during the last three decades. In the context of migration of legacy systems, the infrastructure has evolved from mainframe to multilayered virtual systems as depicted in the Fig. 2, which is a paradigm shift over three decades. This being the one side, on the other side, the software development has also evolved in parallel has get along with the paradigm infrastructure by way of evolution from procedural systems to Cloud. The migration in terms of infrastructure has shifted ranging from mainframe to client server to the latest multi layered virtual systems and still there is significant quantum of mainframe applications that runs on the procedural language such as COBOL. The paradigm shift in Technologies can be classified as three types and are as follows.

- ✓ Procedural up to Web Technology
- ✓ Web Services and SOA
- ✓ Cloud Technology

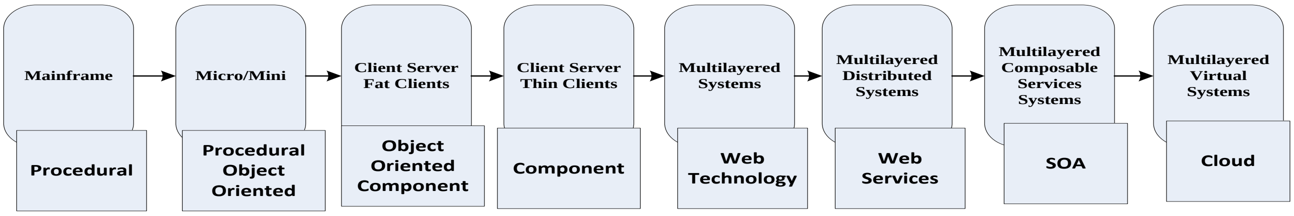
Mainframe
Procedural
Micro/Mini
Procedural Object Oriented
Client Server Fat Clients
Object Oriented Component
Client Server Thin Clients
Component
Multilayered Systems
Web Technology
Multilayered Distributed Systems
Web Services
Multilayered Composable Services Systems
SOA
Multilayered Virtual Systems
Cloud

Fig. 2. Evolution Roadmap

Many researchers have provided a survey of migration with respect to source code, design, architecture, case study and effort estimation. In addition, various survey works surveying different aspects of migration have also emerged. In the previous work[5] we have given a consolidation of various survey works with respect to migration pertaining to earlier and contemporary works under the following topics.

- Migration Approaches
- Tool support for Migration
- SOA Migration
- Cloud Migration

A. *Early Migration Approaches*

J.Bisbal et al [6] in their work on Migration have conducted a detailed study and have described about the migration approaches which are data intense approaches that prioritize migration of databases. The approaches described by him can be classified as Gateway approaches and non-gateway approaches as shown below in Fig. 3.

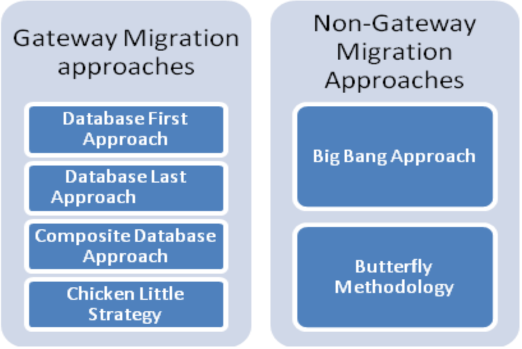

Fig. 3. Lists of Early Migration Approaches

The gateway approaches and non-gateway approaches in the early migration works along with their advantages and disadvantages of the approaches have been tabulated which can be seen in Table I.

TABLE I: COMPARISON OF EARLY MIGRATION APPROACHES

Approach Name	Approach of Migration	Advantages	Disadvantages
Gateway Migration Approaches			
Database First approach	• Data is migrated first and applications and interfaces are migrated incrementally • During the redevelopment of application and interfaces the Legacy system interoperates with its data environment of target system through a forward gateway	• Reuse of Legacy System is possible • Target system development could be incremental • Suitable for migration of fully Decomposable systems	• The Information system is not operational during data migration • Migration of data takes significant time
Database Last approach	• Application is gradually migrated • The last step in this approach is migration of legacy database • Reverse Gateway is responsible for mapping the target database schema to the Legacy database	• Reuse of Legacy System is possible • Suitable for fully decomposable systems • Target system Development could be incremental	• The Information system is not operational during data migration • Migration of data takes significant time
Composite database approach	• Data and application can be incrementally migrated • Development could be incremental • Transaction Co-coordinator is employed to maintain data integrity of Legacy data and Target DBMS • In the target platform the applications are built gradually • Forward and reverse gateway used • A coordinator is used to maintain integrity between legacy and target databases	• Reuse of Legacy system is possible • Eliminates requirement for a single large migration of data as required in the above two approaches • Suitable for migration of fully decomposable, semi decomposable and non-decomposable systems	• Suffers from the overhead of the Database First and Database Last approaches with added complexity due to introduction of co-coordinator

Chicken Little Strategy	• Differs in functionality and placement of gateways from composite database approach • Uses both Forward and Reverse gateways for providing required interoperability • Legacy and Target systems operate in Parallel throughout the migration process • The operational Information system will be a composite of target & legacy information system using gateways for providing required interoperability	• Operational Information System will be a composite of target and legacy information system • Suitable for migration of fully decomposable systems, semi decomposable and non decomposable systems	• Employs complex gateways • Need for interoperation of legacy and target systems through gateways add greater complexity to the already existing complex process
Non-Gateway Approaches			
Big Bang Methodology	• The redevelopment of Legacy system takes place from scratch to run on a new platform using modern architecture, tools and database.	• Improvement over the existing legacy system is possible as the development is from scratch	• Huge Cost is involved • Takes Longer development time • Legacy System Reuse is not possible
Butterfly Methodology	• Gateway free approach • Applies to whole migration process with main focus specifically on legacy data migration in a mission critical environment • Uses Legacy Data Migration Engine suitable for mission critical system • Here during the migration the target system is not available and is not in production	• No Gateways used • Eliminates the necessity of simultaneous access to both legacy and target systems • Testing can be carried out against the already migrated data • Legacy system to be shut only for minimal time	• Target system will not in production while the system is being migrated

The suitability of the approaches differs from project to project and more over each and every approach has its own advantages as well as disadvantages. Further the suitability of the approach varies from project to project and one can be more suitable than the other one. Among the migration approaches mentioned, the Butterfly methodology seems to be better in comparison with the other approaches, considering the fact that the legacy system can be in operation during the migration. In addition, testing can be carried out against the already migrated data and the user can also be trained with that data in the butterfly approach. Also this approach has a controlled complexity.

B. Tool Support for Migration

Migration is not a single step activity but a process that comprises of many phases. The consolidation of the various phases of migration [5][6] from the above works are a) Legacy System Understanding (LSU), b) Target System Understanding (TSU), c) Migration Feasibility Assessment, d) Target System Development and e) Deployment and Provisioning of Target System. Tools support are available for the phases of migration and the same has been consolidated by by Bisbal et al [6]. The tools that have been in use for the different phases are as shown in Table II.

The intent of the tools have been described in our previous work[5]

C. SOA Migration Approaches

Migration of Legacy systems to the target systems such as web services, Service Oriented Architecture has been discussed in the research community as well as by industries. Service oriented paradigm facilitates reuse of business functions provided by legacy systems. Several approaches have been proposed for migration of legacy systems to SOA. A classification on SOA Migration has been discussed in our previous work [5].

The approach to SOA Migration involves the process of migrating legacy assets to SOA, Integrating the legacy software into SOA. S.Ali et al [7] in their work have discussed about the classification of migration approaches to SOA by various researchers. They are invasive approach, non-invasive approach. Invasive approach involves a deep and detail analysis of source code. Non-Invasive approach is one where a layer (wrapper) is added to hide internal complexities so as to present new interfaces as services without modifying the source code. Further, S.Ali et al have classified the approaches as (i) decision making approaches, (ii) partial approaches and (iii) technical approaches. Decision making approach helps to take a particular decision amongst the options of migration, integration, replace or leaving the system as such based on cost and feasibility. Partial approaches give partial solutions in which the portion of the evolution of Legacy to SOA is covered whereas methods of technical nature like code analysis, service identification and service extraction have been included in the Technical approaches. An overview of the some of the migration approaches to SOA is as given below

TABLE II: COMPARISON OF SOA MIGRATION APPROACH

Phase	Tools
Legacy System Understanding	Software Refinary, Rigi, Tools by Companies (IBM, Compuware, Intersolve, Microfocus ,Bachman), Bachman Reengineering Product Set, Software Code Interviewer(SCI), DBMAIN, Seedata, Jude, Omando UML Studio, Eclipse TPTP, ARMIN, E-BUS Toolkit, Understand Refine/C, Imagix 4D
Target System Development	WCL/QFG, GUISYS, CO*STAR, Unify, Forte, Dynasty, ENCINA, Tuxedo, Topend
Deployment and Provisioning	ACUAGL, Persistance Software, Apertus' Enterprise/Integration, Open Horizon's Connection, Enterprise/Access, Tools by Performance Software, Tools by Sector 7, Cyrano Suite

1. *SMART (Service Migration and Reuse Technique)*

Service-Oriented and Reuse Technique (SMART) is a technique which facilitates organizations in taking decisions on exposing the analyzed functionality of Legacy Systems as services in a SOA. D.Smith [8] in his work outlines SMART for potentially reusing the legacy components as servicies after an initial analysis. In [9] the techniques summary, involves Understanding the target SOA requirements, Analyzing the current system; and performing a gap analysis between the target system and the current one and Developing a migration strategy.

2. *Sneed Approach*

Sneed in his approach[10][11] of integrating legacy software into a service oriented architecture has demonstrated how legacy can be reused in construction of web services. For adapting to the technical requirements of SOA in making available the functionality of reusing the services as web services, the services has to be extracted from where it has been implemented. This activity includes discovering, evaluating, extracting and adapting of services to SOA .

3. *MASHUP (MigrAtion to Service Harmonization compUting Platform technology)*

The smart way of combining the content from more than one source into an integrated experience is called "mashup" technology. S.Cetin in [12] has proposed the MASHUP migration strategy that addresses both behavioral and architectural aspects of the migration with a six step migration activities viz., 1) Modeling of target enterprise business 2) Analysis of the legacy systems and infrastructure 3) Mapping business requirements to system components and services identification; Maps model requirement to legacy components and service identification 4) Designing a concrete MASHUP architecture with domain specific kits 5) Defining Service Level Agreement 6) Implementing and deploying of services.

4. *SOMA(Service Oriented Modeling and Analysis)*

In order to use the services in a SOA , SOMA [13] facilitates in integration of systems with techniques for analyzing legacy applications either custom or packaged for services identification, specification and realization. Further it breaks out the benefits of the business capacities of every existing application, candidate service identification that might be used to acknowledge benefits of the business objectives under the new architecture. The needs for development of new services or sourcing it from an external provider have been highlighted in addition to the potential problematic areas identification.

TABLE III: COMPARISON OF SOA MIGRATION APPROACH

SMART	Sneed Approach	Mashup	SOMA
➢ Business goals and drivers considered for migration decisions ➢ Strategic approach to SOA adoption	➢ Salvages the legacy code for business rules identification ➢ Wrap the code for making it as web service	➢ Presentation services included ➢ Approach is top-down and bottom up for business requirements and legacy code respectively ➢ Mapping of business requirements to Services ➢ Integration of Services which eliminates redevelopment	➢ Identifies candidate services for business goals realization ➢ Potential problematic areas for new services development ➢ Focuses for full system development that includes some amount of reuse of legacy systems

D. *Cloud Migration Approaches*

Migrating to cloud and using cloud computing thereon facilitates convenient on demand network access to a shared pool of configurable computing resources of networks, servers, storage etc. The Legacy to Cloud migration transforms legacy applications into the services paradigm at both business and technical levels[14]. The Migration approaches and strategies for migration of legacy system to the cloud have been depicted in our previous work [5].

Reuse and Migration of legacy application to Interoperable Cloud Services (REMICS) [15] is a method where the migration to clouds from legacy systems takes place in a service oriented manner, the methodology being model-driven one.

Advanced Software-based Service Provisioning and Migration of Legacy Software, or ARTIST is model-based approach and tools for legacy-to-cloud migration and has a three process steps of pre-migration, migration and post-migration [16].

TABLE IV: COMPARISON OF CLOUD MIGRATION APPROACH

CloudMIG	SMART decision Framework	REMICS	ARTIST
➢ Semi-automatic Migration of existing software systems to Cloud based applications ➢ Migration is supported by re-engineering	➢ A framework based on SMART for identification of testing process ➢ Describes Target Cloud computing environment for performance software testing	➢ Service oriented way of migration to Cloud ➢ Service composition carried out by replacement and wrapping of legacy software components ➢ Describes target cloud deployment architecture in the context of design	➢ Creates software which can utilize cloud benefits ➢ Covers pre-migration, migration and post migration
Migration	Testing	Architecture	Cloud benefits

CloudMIG (Cloud Migration approach) [17] is an approach for supporting re-engineering to semi-automatically migrating existing software systems to cloud based applications.

SMART decision frame work for migrating software testing frame work to the cloud it is based on the SOA migration, Adoption and Reuse technique (SMART). A framework have been presented by A.Ahamed et al [18] for migrating legacy software systems to cloud enabled one, the frame work being an extension of horseshoe model which supports the process driven approach. The comparison of the cloud migration approaches discussed has been tabulated in Table IV.

III. RESEARCH OPPORTUNITIES

This survey has given a lead to window of research opportunities particularly in the area of Legacy System Understanding that too in representation, effort estimation, design and implementation of migration of Legacy Systems. More over the other area which can be seen as an opportunity is the artefacts involved in the Migration Process which plays an important role in the Migration of Legacy Systems. Legacy Systems are composed of hardware and software artefacts. Legacy System may have a single system or multiple system/server deployment. Some of the artefacts associated with any migration systems are source code, DBMS, Middleware, System Software and Hardware. Further the relationship among the artefacts can also be studied to identify the intra and inter dependencies among them in the Legacy System migration perspective.

IV. CONCLUSION

In this paper we have reviewed the migration of legacy systems covering early migration works as well as contemporary works on migration. This paper has provided the comparative review of the survey works of migration approaches, and recent trends in migration. The research opportunities in the context of migration of legacy systems in the area of Legacy System Understanding, study of aretefacts and their relationship are presented.

REFERENCES

[1] S. Wafa, "Migration of Legacy Systems to Service Oriented Architecture." In The Second International Conference on Digital Enterprise and Information Systems (DEIS2013) (pp. 166-173). The Society of Digital Information and Wireless Communication., 2013.

[2] R. Gimnich and A. Winter, "SOA Migration: Approaches and Experience," *Softwaretechnik-Trends*, no. November, 2007.

[3] S. Comella-Dorda, K. Wallnau, R. C. Seacord, and J. Robert, "A survey of black-box modernization approaches for information systems," *Softw. Maintenance, 2000. Proceedings. Int. Conf.*, 2000.

[4] S. Comella-dorda, K. Wallnau, R. C. Seacord, and J. Robert, "A Survey of Legacy System Modernization Approaches," no. 4, pp. 1–30, 2000.

[5] A. Sivagnana ganesan and T. Chithralekha, "A Survey on Survey of Migration of Legacy Systems."

[6] J. Bisbal, D. Lawless, B. Wu, J. Grimson, V. Wade, R. Richardson, and D. O. Sullivan, "A Survey of Research into Legacy System Migration," *Integration The Vlsi Journal*. pp. 1–39, 1997.

[7] S. Ali and S. Abdelhak-Djamel, "Evolution approaches towards a Service oriented architecture," *Proceedings of 2012 International Conference on Multimedia Computing and Systems, ICMCS 2012*. pp. 687–692, 2012.

[8] D. Smith, "Migration of legacy assets to service-oriented architecture environments," *Proceedings - International Conference on Software Engineering*. pp. 174–175, 2007.

[9] R. a. Salama and S. G. Aly, "A decision making tool for the selection of service oriented-based legacy systems modernization strategies," *Proceedings of the 2008 International Conference on Software Engineering Research and Practice, SERP 2008*. pp. 396–402, 2008.

[10] H. M. Sneed and A. Gmbh, "Integrating legacy Software into a Service oriented Architecture," 2006.

[11] H. M. Sneed, "Migrating to Web Services A Research Framework.pdf," 2007.

[12] S. Cetin, N. I. Altintas, H. Oguztuzun, A. H. Dogru, O. Tufekci, and S. Suloglu, "A mashup-based strategy for migration to Service-Oriented Computing," *2007 IEEE International Conference on Pervasive Services, ICPS*. pp. 169–172, 2007.

[13] S. Balasubramaniam, G. A. Lewis, E. Morris, S. Simanta, and D. Smith, "SMART : Application of a Method for Migration of Legacy Systems to SOA Environments." pp. 678–690, 2008.

[14] Z. Wan and P. Wang, "A Survey and Taxonomy of Cloud Migration." pp. 175–180, 2014.

[15] "REMICS." [Online]. Available: http://www.remics.eu/home. [Accessed: 15-Jun-2016].

[16] L. S. Girish and H. S. Guruprasad, "Survey on Service Migration to Cloud Architecture," vol. 5, no. 5. pp. 507–510, 2014.

[17] S. Frey and W. Hasselbring, "The CloudMIG Approach: Model-Based Migration of Software Systems to Cloud-Optimized Applications," *Internati J. Adv. Softw.*, vol. 4, no. 3, pp. 342–353, 2011.

[18] A. Ahmad and M. A. Babar, "A framework for architecture-driven migration of legacy systems to cloud-enabled software," *Proceedings of the First International Conference on Dependable and Secure Cloud Computing Architecture - DASCCA '14*. pp. 1–8, 2014.