A Survey on Survey of Migration of Legacy Systems

A.Sivagnana ganesan
Dept. of Banking Technology
Pondicherry University
Pondicherry 605014
India
ganesanas@gmail.com

T.Chithralekha
Dept. of Computer Science & Engg.
Pondicherry University
Pondicherry 605014
India
tchithralekha@gmail.com

ABSTRACT

Legacy systems are mission critical complex systems which are hard to maintain owing to shortage of skill sets and monolithic code architecture with tightly coupled tiers, all of which are indication of obsoleteness of the system technology. Thus, they have to be migrated to the latest technology(ies). Migration is an offspring research in Software Engineering which is almost three decades old research and numerous publications have emerged in many topics in the migration domain with focus areas of code migration, architecture migration, case study on migration and effort estimation on migration. In addition, various survey works surveying different aspects of migration have also emerged. This paper provides a survey of these survey works in order to provide a consolidation of these survey works. As an outcome of the survey on survey of migration, a road map of migration evolution from the early stages to the recent works on migration comprising of significant milestones in migration research has been presented in this work.

Keywords

Migration; SOA; Legacy Systems; Software Evolution

1. INTRODUCTION

Legacy Systems are mission critical systems with monolithic code architecture having restrictions to archaic hardware, software and are short of resources in terms of skill sets, documentation and is therefore hard to maintain and are inflexible in the ever changing and emerging technologies and the systems works in silo. Migration of Legacy systems is one of the ways of modernization, the others being replacement, reengineering or redevelopment and wrapping [6][7]. 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.

ICIA-16, August 25-26, 2016, Pondicherry, India
© 2016 ACM. ISBN 978-1-4503-4756-3/16/08…$15.00
DOI: http://dx.doi.org/10.1145/2980258.2980409

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 works carried out with respect to migration approaches, migration strategies, tool supports for various phases of migration, recent trends in migration etc. This paper provides a survey of these survey works in order to provide a consolidation of these survey works. In addition to the above, as an outcome of the survey on survey of migration, a road map of migration evolution from the early stages to the recent works on migration comprising of significant milestones in migration research has been presented in this work. This survey and hence the roadmap could abreast a naive researcher about the domain of migration. The rest of the paper is organized as follows viz., section 2 covers the background information, section 3 briefs about Survey of the different Surveys on Migration, section 4 present an evolution roadmap of Migration of Legacy Systems and section 5 gives the Conclusion.

2. BACKGROUND ON MIGRATION

Migration is moving from obsolete or old languages or platforms towards a more modern environment which allows the system to be easily maintained and adapted to new business requirements [20]. Several definitions about migration have been given by the research community as well as by the industries wherein they have analogously used the terms of software migration, application migration and migration in IT (Hardware and Software). Some of the definitions are as given below:

> *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* [25].
>
> *Migration in IT means the move to a new technical environment, mostly for business reasons and for fulfilling (new) non-functional requirements* [9].
>
> *Software Migration is moving a system into a new technical environment potentially involving data management, communication and programming environment* [9].

Migration can be a combination of Language or Code Migration, Operating System Migration, data migration, User Interface(UI) migration, Architecture migration, System software and Hardware migration or migration of any of these individually [9][21]. Migration can be executed automatically, semi-automatically or manually. Automated migration is found to be applied to migration of languages, databases and platforms using software tools like automated parsers and converters. Here, the transformation is algorithmic in nature and does not require injection of human intelligence into the transformation process [15].

Migration is one among the Information system modernization strategies while the other strategies are Replacement, Reengineering and Wrapping[6][7]. Modernization forms an important phase of Information system Life Cycle as depicted in Figure 1. [6]. Modernization effort is required when maintenance falls behind the business needs after supporting it for sufficient time. Thereafter, the system finally has to be replaced when it could not be evolved further.

Migration is not a single step activity but a process comprising of many phases. The various phases of migration and details are described below,

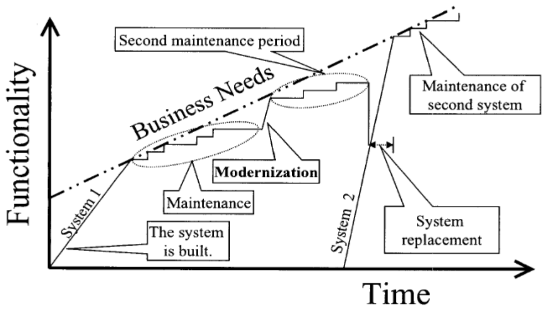

Figure 1 Information System Life Cycle[6]

2.1 Phases of Migration

Migration of Legacy system is a process that comprises of several phases. J. Bisbal et al [4] and Khadka et al [12] have provided the phases of migration in their works. The consolidation of the various phases of migration 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

a) Legacy System Understanding(LSU)

The understanding of the source code and structure of the data of the legacy systems are essential to all migration projects.

In this phase, a detailed analysis of the Legacy system is carried out with the techniques of reverse engineering, program understanding and architecture recovery. J.Ransom et al [16] in their work have used an assessment method to gain an adequate depth of understanding from the technical, business and organizational perspective.

b) Target System Understanding(TSU)

The desired architectural representation of the target system is facilitated in this phase. This phase describes the target environment comprising of activities such as defining major components/functionalities of the environment, specific technologies and standards to be used and the state of target system.

c) Migration Feasibility Assessment

The understanding of legacy system and the target system helps in undertaking the feasibility assessments at technical, economical and organization level. The code complexity in technical feasibility and cost benefit analysis in economical feasibility can be included in the assessment

d) Target System Development

The Target system is developed for the requirements specified, which was arrived based on the phases of LSU, TSU and Migration Feasibility Assessment. Program slicing, concept slicing, graph transformation code translation, model driven program transformation, screen scraping, code query technology etc are used for extracting the legacy code as services to be incorporated in the target system.

e) Deployment & Provisioning of Target System

The deployment & provisioning phase is concerned with deployment and management of the developed system or services. The system or services developed are deployed in the corresponding infrastructure

3. SURVEY ON SURVEY OF MIGRATION

Migration research is almost three decades old research and numerous publications have emerged in many topics in the migration domain. 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. These are discrete works and limit the comprehensive understanding of various dimensions of research carried out in Migration. Hence, in this paper we have given a consolidation of various survey works with respect to migration pertaining to earlier and contemporary works. This research work consolidates the survey on the following topics

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

3.1 Migration Approaches in Earlier Works

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. J.Bisbal et al [4] 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 Figure 2.

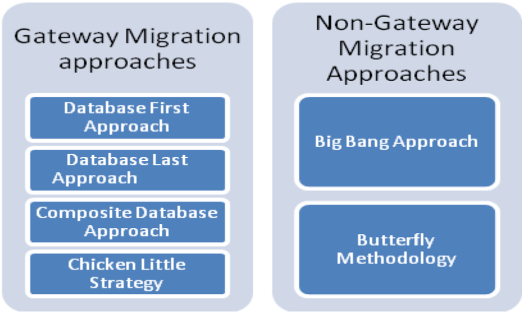

Figure 2 Lists of Early Migration Approaches

Gateway Migration Approaches

Gateway Migration approaches are approaches that use gateways for migration. Gateways are required for simultaneous access of the legacy and target system enabling interoperation between the two heterogeneous information systems. Forward gateway is required for legacy system to access the target system, whereas Reverse gateway is required for target system to access the legacy system.

Non-Gateway Migration Approaches

The Non-Gateway approaches are gateway free approaches that do not uses gateway for the migration of legacy systems. A comparison of the migration approaches is depicted in Table 1 below.

Table 1 Comparison of 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 the target system will not be in production while the system is being migrated	• No Gateways used • Eliminates the need for simultaneous access of 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

As is evident from the detailed comparisons of the above approaches in Table 1, each and every approach has its own advantages as well as disadvantages. Moreover, 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.

3.2 Tools support for Migration in Earlier Works

Migration comprises of various phases as stated in section 2.1. For each of these phases, tools are available. Researchers have surveyed the different tools which are applicable in different phases of migration in their individual works. Tools support available for the phases of migration as consolidated by Bisbal et al [4] are as shown in Figure 3. Bisbal et al have mentioned that the tool list provided is not exhaustive or full. The reason being that almost all approaches and methods proposed by researchers come with tool support for their migration. The Tools and the purpose for which they are intended are as shown in Annexure – A.

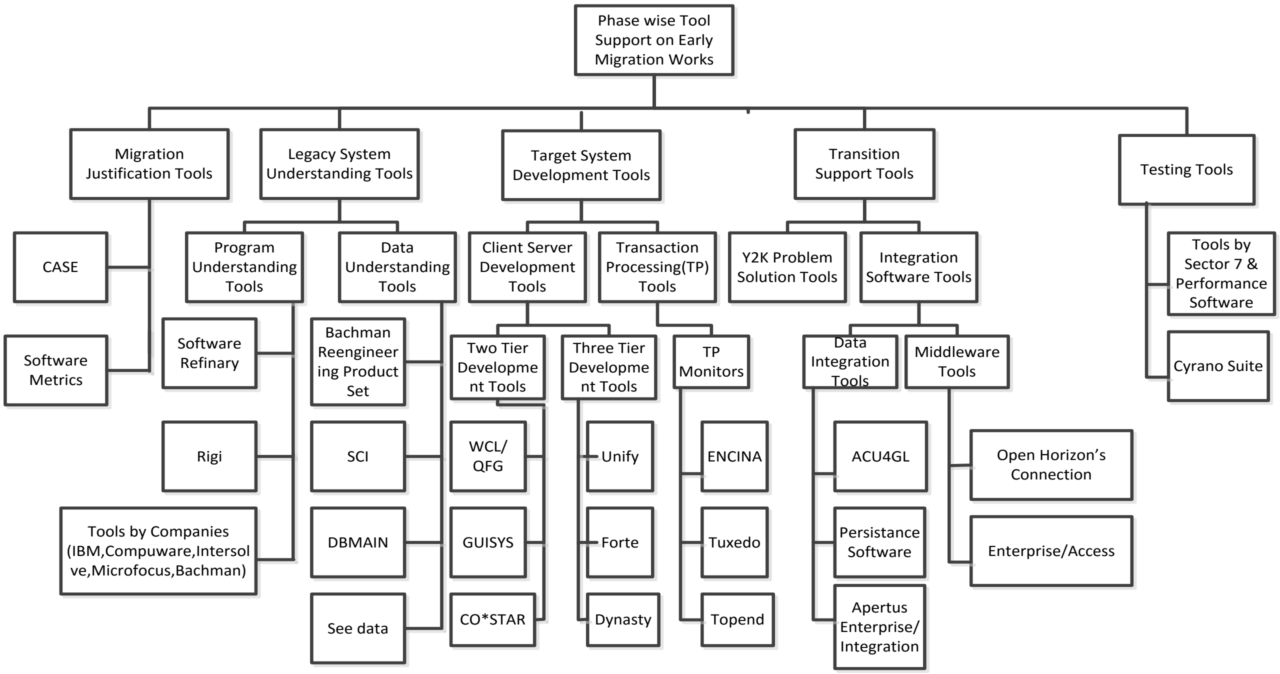
Phase wise Tool Support on Early Migration Works
Migration Justification Tools
Legacy System Understanding Tools
Target System Development Tools
Transition Support Tools
Testing Tools
CASE
Software Metrics
Program Understanding Tools
Data Understanding Tools
Software Refinary
Rigi
Tools by Companies (IBM,Compuware,Intersolve,Microfocus,Bachman)
Bachman Reengineering Product Set
SCI
DBMAIN
See data
Client Server Development Tools
Transaction Processing(TP) Tools
Two Tier Development Tools
Three Tier Development Tools
TP Monitors
WCL/ QFG
GUISYS
CO*STAR
Unify
Forte
Dynasty
ENCINA
Tuxedo
Topend
Y2K Problem Solution Tools
Integration Software Tools
Data Integration Tools
Middleware Tools
ACU4GL
Persistance Software
Apertus Enterprise/ Integration
Open Horizon's Connection
Enterprise/Access
Tools by Sector 7 & Performance Software
Cyrano Suite

Figure 3 Phase wise Tool support on Early Migration Works

3.3 SOA Migration

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. Following the study on surveys of migration, we have classified the SOA Migration as depicted in Figure 4. Brief descriptions of some of the migration approaches are given below.

3.3.1 Approaches to SOA Migration

The approach to SOA Migration involves the process of migrating legacy assets to SOA, Integrating the legacy software into SOA. S.Ali et al [1] 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 provide partial solutions which only cover part of the Legacy system to SOA evolution process. Technical approaches include approaches that use technical methods such as code analysis and service identification and extraction. An overview of the some of the migration approaches to SOA is as given below:

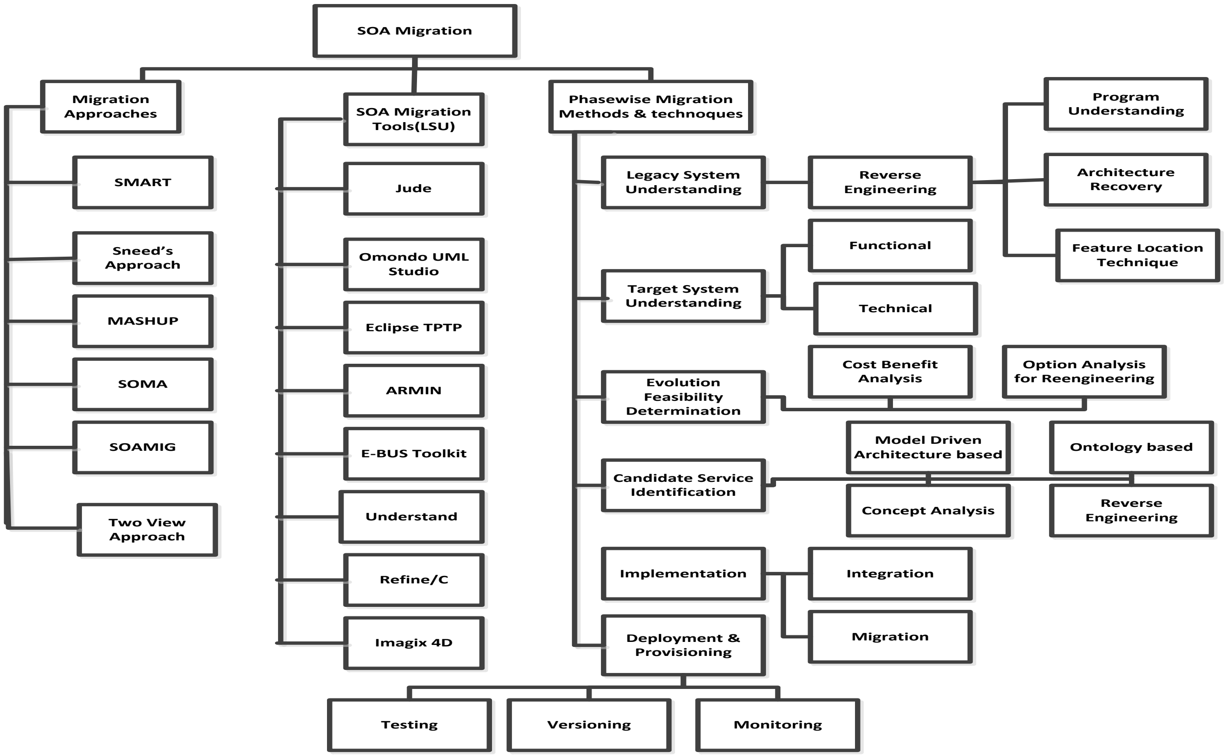

Figure 4 SOA Migration

a) SMART(Service Migration and Reuse Technique)

D.Smith [22] in his work on Migration of Legacy Assets to SOA outlines SMART, a technique for making an initial analysis of legacy components for their potential of being reused as services. SMART is a family of processes which helps to take decisions about the feasibility of exposing legacy systems as services within a SOA environment. SMART Migration planning (SMART-MP) process provides a plan for the migration of selected components to services and identifies the scope for a pilot effort. The elements are (i) SMART-MP Process, (ii) Service Migration Interview Guide(SMIG), (iii) SMART Tool and (iv) Artefact Templates.[2]

b) Sneed's Approach

Sneed in his approach[23] of integrating legacy software into a service oriented architecture has demonstrated how legacy can be reused in construction of web services.

c) 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 [5] has proposed the MASHUP migration strategy that addresses both behavioral and architectural aspects of the migration with a six step migration activities as stated below:

- MODEL - Modeling of target enterprise business
- ANALYSIS - Analysis of the legacy systems and infrastructure
- MAPS AND IDENTIFY – Mapping business requirements to system components and services identification. Maps model requirement to legacy components and service identification
- DESIGN - Designing a concrete MASHUP architecture with domain specific kits.
- DEFINE – Defining Service Level Agreement
- IMPLEMENT & DEPLOY – Implementing and deploying of services.

d) SOMA(Service Oriented Modeling and Analysis)

SOMA [2] facilitates integration of systems with techniques for analyzing legacy applications, custom and packaged, to identify, specify and realize services for use in a service-oriented architecture. Further it breaks out the business functions of each existing application, identifying candidate services that can be used to realize business goals under the new architecture. It also identifies potentially problematic areas and highlights areas where new services need to be developed or sourced from an external provider. The SOAMIG [27] method describes the importance of service design, which is the result of forward engineering and reverse engineering

e) Two View Approach

Maryam Razavian and P.Lago [18] in their work have represented the migration approaches using views to analyze and categorize them, the two view beings knowledge and activity . The categories of activities carried out and knowledge elements used or produced in the SOA migration approaches have been discussed. Further they have come out with a SOA migration frame of reference that can be used either for selecting an approach among existing approaches or in developing a new migration approach.

3.3.2 SOA Migration Tools

Various tools and techniques have already been successfully developed and used in legacy to SOA evolution. For instance, reverse engineering, program understanding, architectural recovery can be used, often with tool support to generate system artefacts. The implementation phase is concerned with the technical evolution of the whole legacy system to the target system using various techniques, often supported by the tools. Wrapping based legacy to SOA evolution are reported by Sneed (H. M. Sneed, 2008; H. M. Sneed, 2009) in which the author has developed various tools to support the evolution. The intent of the SOA Migration Tools(LSU) depicted in Figure 4 is shown below in Table 2

Table 2 SOA Migration Tools(LSU) [12]

Tool	*Intent*
Jude	Static view extraction – Extraction and diagram manipulation
Omando UML Studio	Static view extraction – Generate Package diagram
Eclipse TPTP	Dynamic view extraction - extract runtime information
ARMIN	Identification of undocumented dependences in source code
E-BUS Toolkit	Architecture reconstruction for various java based systems
Understand	explore features, functional dependencies and compute various metrics
Refine/C	to understand, evaluate, and re-document existing C code
Imagix 4D	A comprehensive program understanding tool for C and C++ programs.

3.3.3. Phase wise Methods & Techniques for Legacy to SOA Migration

The phases of Migration have been briefed already in section 2.1. The phases of SOA Migration that differs from phases of earlier migration works are Candidate Service Identification and Migration feasibility. The phase of Target System Development is used in earlier migration works whereas Target system Understanding is used in SOA Migration. R.Khadka et al [12] [13] [14] in their works have discussed about the Methods and Techniques support for the phases of Migration. Here some of the methods and techniques widely used in the SOA Migration phases as reported by R.Khadka et al in [12] have been described briefly below.

a) Legacy System Understanding

Reverse engineering can be defined as identifying the system's components and their interrelationships and create representations of the system in another form or at a higher level of abstraction. Software systems are reverse engineered in order to understand the system and its structure. (van Geet, 2011)

The some of the important tasks of reverse engineering are Program Analysis, Plan Recognition, Concept Assignment, Re-documentation and Architecture Recovery [24]. *Program understanding or Program comprehension* is a term related to reverse engineering. The goal of the Program understanding is to enable the understanding of functional and data concepts of the program. Here the understanding begins with source code whereas for reverse engineering it can start at binary and executable form of the system or at high-level descriptions of the design. *Architecture Recovery* has been defined as a process which identifies and extracts abstractions at a higher level from the legacy systems [17]. The functional units of source code are identified by *Feature location.*

b) Target System Understanding

The methods and techniques used in this phase cover the functional and technical aspects with focus to selection of architecture, technology, functional specification, SOA environment details and related standards.

c) Migration Feasibility Assessment

The techniques of Cost Benefit Analysis and Option Analysis for Reengineering (OAR) are widely used by the researchers in Migration Feasibility determination. OAR has been used to determine the feasibility decision point.

d) Candidate Service Identification

Among the several methods and techniques used, model driven architecture based methods make use of SOA Meta model in identification of services in legacy code. An ontology based approach stores knowledge of both the application domain and the legacy code. Later concept analysis and relational concept analysis are used for identification of services. Data flow diagrams are generated from legacy code using the techniques of reverse engineering to identify the services.

e) Implementation

R.Khatka et al [13] in their works have classified the methods and techniques in this phase as pertaining to (i) integration and (ii) migration. In legacy system integration there is no substantial changes in the legacy code when used in new environment and whereas in legacy system migration, the legacy code is transformed, internally modified, or reused in a new environment.

f) Deployment & Provisioning

The services evolved have to be managed and the behaviors of the services have to be controlled. The activities required for the above tasks are testing, versioning and monitoring. The cloud migration of legacy systems has been covered as the next topic.

3.4 Cloud Migration

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[26]. The Migration approaches and strategies for migration of legacy system to the cloud have been depicted in Figure 5.

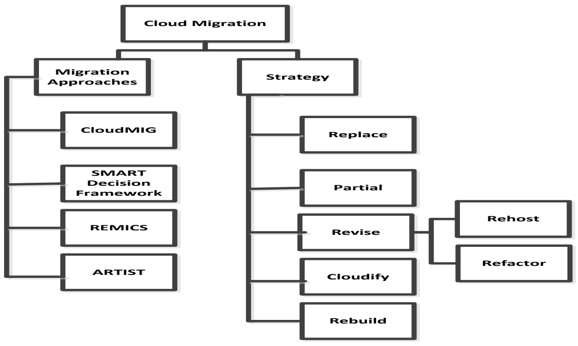

Figure 5 Cloud Migration

3.4.1 Cloud Migration Approaches

As in SOA Migration, separate migration approaches are available for cloud and some of them are briefed hereunder. A.Ahmad and et al [11] in their works have conducted a systematic review of cloud migration through 23 selected studies using a characterization framework and the findings have been presented elaborately. Cloud migration approaches either help in decision making on cloud migration or cloud enable migration of legacy systems [11]. Some of the Cloud Migration approaches have been discussed below briefly:

a) *Cloud Migration approach (CloudMIG)*

CloudMIG [10] is an approach that supports re-engineering for migration of existing software systems to cloud based applications semi automatically. The activities that has been included in this migration approach are Extraction, Generation, Selection, Evaluation and Transformation

b) SOA Migration Adoption and Reuse Technique(*SMART) decision frame work*

This framework is for migration of testing frame work to the cloud which is based on SMART. It helps the organization for identifying the existing process of testing. Further it also describes the target cloud computing environment requirement for performance software testing. [10]

c) Reuse and Migration of legacy application to Interoperable Cloud Services (REMICS)

REMICS [19] is a model-driven methodology and tools for legacy to cloud migrations in a service oriented way. In this approach, SOA and cloud computing patterns are applied for migrating the source architecture to the target cloud capable architecture by application of SOA and cloud computing patterns. Here the service composition is emphasized by way of replacing and wrapping legacy software components. REMICS project has offered a metamodel and UML profile called PIM4Cloud for describing the target cloud deployment architecture in design perspective.

d) Advanced Software-based Service provisioning and Migration of Legacy Software(ARTIST)

The objective of ARTIST is to create software which can utilize cloud benefits rather than running on top of it. The process of ARTIST comprises of three steps viz,, pre-migration, migration and post-migration [10].

3.4.2 Cloud Migration Strategies

The Cloud Migration Strategies have been reported in the works of [3] [26][8]. The strategies are described briefly as follows:

Replace strategy replaces an old application completely with a commercial alternative delivered using the SaaS model by discarding the old application(s).

Partial Migration is a strategy of migration where only a subset of the components of the applications are moved to Cloud[3]. Migration of authentication logic of a system based on Single Sign on (SSO) is one such example of this migration.

In R*ehost* strategy, applications and the middleware software running them are redeployed as is on top of IaaS, instead of on-premises hardware.

Refactoring means running applications on top of PaaS – middleware layer is provided by the cloud vendor but application layer stays mostly the same as before.

Revising is an option in strategy wherein the improvement is made to the applications so that it can fit into the cloud environment first and are then migrated either using the rehost or refactoring strategies

Cloudify the application is strategy where a complete migration of the application takes place.

In *rebuild strategy* applications are completely rewritten on top of PaaS targeting at maximum benefits derived from cloud [8][26].

4. Evolution Roadmap of Migration of Legacy Systems

Over the three decades, the application development and deployment have taken a long journey from mainframe to Cloud. Based on this survey, we have evolved Evolution Roadmap of Migration of Legacy systems as shown in Fig. 6. The evolution roadmap has been depicted in the perspective of Platform and Technology. The sequence of evolution of platform are Mainframe, Micro /Mini Computers, Client Server – Fat Clients, Client Server – Thin Clients, Multi layered systems, Multilayered distributed systems, Multilayered composable Services systems, Multi layered Virtual Systems. The sequence of evolution of technology are procedural systems, object oriented systems, component based systems, Web based systems, Web services, SOA Paradigm and at the end, service realization of all infrastructures in Cloud. This survey of Survey about migration of legacy systems with the help of roadmap clearly depicts the comprehensive picture of entire gamut of legacy systems which will help the researchers in their understanding and carrying out further research on migration of legacy systems.

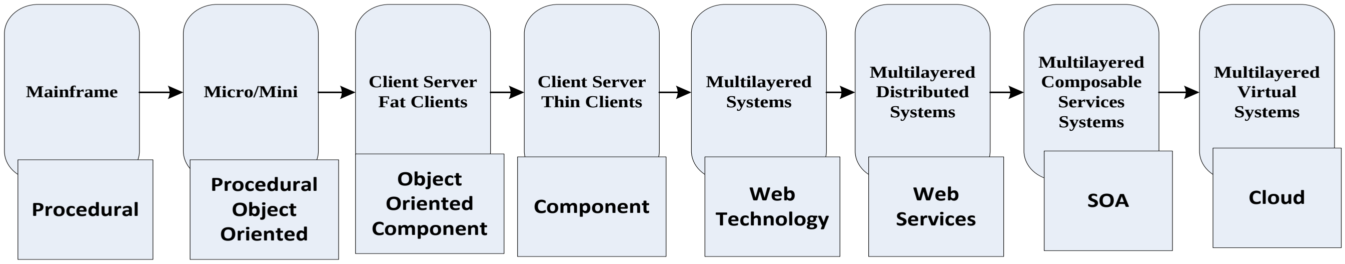
Mainframe
Procedural
Micro/Mini
Procedural Object Oriented
Client Server Fat Clients
Object Oriented Component
Client Server Thin Clients
Component
Multilayered Systems
Web Technology
Multilayered Distributed Systems
Web Services
Multilayered Composable Services Systems
SOA
Multilayered Virtual Systems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

Figure 6 Evolution Roadmap

5. CONCLUSION

In this paper we have surveyed the survey of migration of legacy systems covering early migration works as well as contemporary works on migration. This paper has provided the consolidations of the survey works of migration approaches, migration strategies, tool support for various phases of migration and recent trends in migration. An Evolution Roadmap on migration of Legacy systems have been provided from early stages to the recent works on migration comprising of significant milestones in migration research.

6. REFERENCES

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

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

[3] Beserra, P. V., Camara, A., Ximenes, R., Albuquerque, A.B. and Mendonça, N.C. 2012. Cloudstep: A step-by-step decision process to support legacy application migration to the cloud. *2012 IEEE 6th International Workshop on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems, MESOCA 2012*.

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

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

[6] Comella-Dorda, S., Wallnau, K., Seacord, R.C. and Robert, J. 2000. A survey of black-box modernization approaches for information systems. *Software Maintenance, 2000. Proceedings. International Conference on*. (2000).

[7] Comella-dorda, S., Wallnau, K., Seacord, R.C. and Robert, J. 2000. A Survey of Legacy System Modernization Approaches. 4 (2000), 1–30.

[8] Five Ways to Migrate Applications to the Cloud: *http://www.gartner.com/newsroom/id/1684114*. Accessed: 2016-05-27.

[9] Gimnich, R. and Winter, A. 2007. SOA Migration: Approaches and Experience. *Softwaretechnik-Trends*. November (2007).

[10] Girish, L.S. and Guruprasad, H.S. 2014. Survey on Service Migration to Cloud Architecture.

[11] Jamshidi, P., Ahmad, A. and Pahl, C. 2013. Cloud Migration Research: A Systematic Review. *IEEE Transactions on Cloud Computing*. 1, 2 (2013), 142–157.

[12] Khadka, R., Saeidi, A., Idu, A., Hage, J. and Jansen, S. 2012. Legacy to SOA Evolution: A Systematic Literature Review. *A.D. Ionita, G. Lewis & M. Litoiu (Eds.) Migrating to SOA and Cloud Environments: Challenges in Service Oriented Architecture and Cloud Computing Environments: IGI Global (in press)*.

[13] Khadka, R., Saeidi, A., Idu, A., Hage, J. and Jansen, S. 2012. *Legacy to SOA Evolution: Evaluation Results*.

[14] Khadka, R., Saeidi, A., Jansen, S. and Hage, J. 2013. A structured legacy to SOA migration process and its evaluation in practice. *c2013 IEEE 7th International Symposium on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems, MESOCA 2013*. 4, March (2013), 2–11.

[15] Malinova, A. 2010. Approaches and Techniques for Legacy Software Modernization.

[16] Ransom, J., Warren, I., Sommerville, I., Somerville, I. and Warren, I. 1998. A Method for Assessing Legacy Systems for Evolution. *Software Maintenance and Reengineering, 1998. Proceedings of the Second Euromicro Conference on*. (1998), 128–134.

[17] Rasool, G. and Asif, N. 2007. Software Architecture Recovery. (2007), 422–427.

[18] Razavian, M. and Lago, P. 2015. A systematic literature review on SOA migration.

[19] REMICS: *http://www.remics.eu/home*. Accessed: 2016-06-15.

[20] Rodríguez, A., Caro, A. and Fernández-Medina, E. 2009. Towards framework definition to obtain secure business process from legacy information systems. *Proceeding of the first international workshop on Model driven service engineering and data quality and security - MoSE+DQS '09*. (2009), 17.

[21] Six Steps to Migration Project Success: *http://www.syntelinc.com/insights-and-resources/publications/six-steps-migration-project-success*. Accessed: 2015-11-19.

[22] Smith, D. 2007. Migration of legacy assets to service-oriented architecture environments. *Proceedings - International Conference on Software Engineering*.

[23] Sneed, H.M. and Gmbh, A. 2006. Integrating legacy Software into a Service oriented Architecture. (2006).

[24] Tilley, S. 1998. A Reverse- Engineering Environment Framework. April (1998).

[25] Wafa, S. 2013. 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.

[26] Wan, Z. and Wang, P. 2014. A Survey and Taxonomy of Cloud Migration.

[27] Winter, a., Zillmann, C., Herget, A., Teppe, W., Theurer, M., Fuhr, A., Horn, T., Riediger, V., Erdmenger, U., Kaiser, U., Uhlig, D. and Zimmermann, Y. 2011. SOAMIG project: Model-driven software migration towards Service-Oriented Architectures. *CEUR Workshop Proceedings*. 708, 01 (2011), 15–16.

Annexure – A

Tools		Intent
Justification tools	CASE Tools	Supports justification process in migration which can be related to planning phase in application development.
	Software Metrics	
Program Understanding Tools	Software Refinary	Reverse Engineering Tools Generator and supports platforms of Sunsparc; HP 9000/7xx; IBM RS/6000.
	Rigi	Assists in documentation reconstruction and supports platforms of Sunsparc; IBM RS/6000; Pentium PC and supports viewing of parsed C, C++, PL/AS, COBOL, and LaTeX code and language-independent tools.
	Tools by Companies (IBM, Compuware, Intersolve, Microfocus ,Bachman)	isolate the data information in COBOL applications.
Data Understanding Tools	Bachman Reengineering Product Set	focus on recapturing the physical database designs semantics.
	Software Code Interviewer(SCI)	Static analysis reverse engineering tool with an intent of discovering data model from COBOL source code and Job Control Statements.
	DBMAIN	is a graphical, general-purpose, programmable, CASE environment, dedicated to database application engineering and focus on recapturing the physical database designs semantics.
	Seedata	Relational legacy database structure representation using computer graphics.
2 Tier C/s Development Tools	WCL/QFG	Open Client Server Application generator which generates complete visual basic & powerbuilder application linked to 3270 and 5350 legacy application data without any coding. Create Client Server application by adding new graphic oriented workstations using S/390 and AS/400 application as data source.
	GUISYS	Fully automatic migration of mainframe and midrange legacy applications to Client Server environment under window/OS2. Features a knowledge base that monitors the flow of an application when it is in use and analysis the way the screen is presented to the user.
	CO*STAR	Creates a windows interface for legacy application. Enables window/windows application integration without expensive rewrite or porting of core legacy application. Ability to Access and display data from independent data sources in the same window as data from legacy or proprietary databases. Dynamic data exchange between legacy application and emerging windows based path. Screen scrapping tool that depends on the use of a terminal emulator for communications link to the host mainframe legacy system. Screen scraping is the process of replacing the character based front end of a legacy information system with a PC client based graphical user interface.
3 Tier C/s Development Tools	Unify	Provides visual programming tools and supports a variety of Unix platforms and databases, as well as Windows. Provides automated application partitioning for developing three-tier architectures.
3 Tier C/s Development Tools	Forte	Offer "after the fact" application partitioning which means that code and function modules can be reallocated to new servers after the application has been built. Offers important flexibility and performance benefits. Automatic partitioning mechanism of the application components in a 3 tier c/s architecture.
	Dynasty	Open, object-oriented, enterprise-class client/server development environment for distributed computing. Enables the rapid design, development, and evolution of distributed business objects. From a target independent specification, DYNASTY builds distributed client/server applications that are: 3-tier, partitioned, high performance 3GL code, scalable and cross-platform.

Tp monitors	ENCINA	Support and exploit the large UNIX** Symmetric Multi Processor and parallel processors under the AIX Version 4.1.3 operating system. Provides the Distributed Computing Environment based infrastructure to "glue" together applications distributed across the PC Desktop, UNIX servers and the mainframe, with complete transactional integrity.
	Tuxedo	Unix based distributed TPMs providing Distributed Transaction processing and other services of load balancing and application replication.
	Topend	Unix based distributed TPMs providing Distributed Transaction processing and other services of load balancing and application replication.
Data Integration Tools	ACUAGL	provides seamless interface from COBOL to RDBMS and translates COBOL I/O verbs into SQL.
	Persistance Software	Interface Object oriented developed applications with RDBMS.
	Apertus' Enterprise/Integration	Integration of different data sources of an enterprise. Includes support for detection of redundant data and value conflict resolution.
Middleware Tools	Open Horizon's Connection	proprietary ORB for providing secure connection for any end user application to the tiers of database and application.
	Enterprise/Access	Enables deployment of 3 tier enterprise strength client/Server applications and provides a controlled and cost-effective migration path of legacy systems.
Testing Tools	Tools by Performance Software	supports automated testing.
	Tools by Sector 7	It specialises in VMS to Unix and NT system migration, follows five step migration process viz., Assessment, Planning, porting, validation, productization and supports most stages of migration and automated testing.
	Cyrano Suite	specialised in testing client/server based applications.