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 into Early Migration Approaches, SOA Migration Approaches and Cloud Migration. A comparative review is presented among these 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].

II. MIGRATION APPROACHES OF LEGACY SYSTEMS – A COMPARISON

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.
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.

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

<p>| TABLE II: COMPARISON OF SOA MIGRATION APPROACH |</p>
<table>
<thead>
<tr>
<th>Phase</th>
<th>Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legacy System Understanding</td>
<td>Software Refinary, Rigi, Tools by Companies (IBM, Compuware, Intersolve, Microfocus, Bachman), Bachman Reengineering Product Set, Software Code Interviewer(SCI), DBMAIN, Scedata, Jude, Oramdo UML Studio, Eclipse TPTP, ARMIN, E-BUS Toolkit, Understand Refine/IC, Imagix 4D</td>
</tr>
<tr>
<td>Target System Development</td>
<td>WCL/QFG, GIUSYS, CO*STAR, Unify, Forte, Dynasty, ENCINA, Tuxedo, Topend</td>
</tr>
<tr>
<td>Deployment and Provisioning</td>
<td>ACUAGL, Persistence Software, Apertus’ Enterprise/Integration, Open Horizon’s Connection, Enterprise/Access, Tools by Performance Software, Tools by Sector 7, Cyrano Suite</td>
</tr>
</tbody>
</table>
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 services 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>
<thead>
<tr>
<th>TABLE III: COMPARISON OF SOA MIGRATION APPROACH</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SMART</strong></td>
</tr>
<tr>
<td>Business goals and drivers considered for migration decisions</td>
</tr>
<tr>
<td>Strategic approach to SOA adoption</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

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].
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 artefacts and their relationship are presented.

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

REFERENCES