

## Analysis of Legacy System in Software Application Development: A Comparative Survey

M. Srinivas\*, G. Ramakrishna\*, K. Rajasekhara Rao\*\*, E. Suresh Babu\*

\* Department of CSE, KLEF, KL University, AP, India

\*\* Director, Sri Prakash College of Engineering, AP, India

---

### Article Info

#### Article history:

Received Jun 16, 2015

Revised Nov 9, 2015

Accepted Nov 26, 2015

---

#### Keyword:

Business Process  
Documentation  
Information  
Legacy systems  
Software development

---

### ABSTRACT

Software evolution is one of the challenging issues in today's business environment. It is necessary for the organizations, which make use of information, and communication technologies will have to align their business processes to compete with global business. The existing large software systems ("legacy" systems) have never been built to cope with the current business requirement for their poor coding, design structures, logic and documentation. Moreover, Legacy applications have various problems such as lack of up to-date documentation, skilled man power, resources of the Legacy applications, and high maintenance costs. Even though the Legacy system is obsolete, it contains detailed business rules and in continuous use, because it satisfies the users' needs and forms the backbone of the information flow of organization. One of the possible solutions is to refactor or modernize those systems into a new platform. It is necessary to analyse the existing legacy system for better understanding the business logic and its functionalities. This paper analyses various techniques proposed for understanding Legacy systems in existence.

*Copyright © 2016 Institute of Advanced Engineering and Science.  
All rights reserved.*

---

### Corresponding Author:

M. Srinivas,  
Department of Computer Science and Engineering,  
K L University,  
Greenfields, Vaddeswaram, Guntur District- 62102, Andhra Pradesh, India.  
Email: srinu\_cse@kluniversity.in

---

## 1. INTRODUCTION

Over the last two decades, legacy systems [1] are one of the business applications implemented in most of the organisation. The legacy applications were developed based on technologies such as mainframes, SAP etc., which continues to provide core services to an organisation. Recently, there are prominent advances in technology, it is necessary to replace the legacy systems with the new one for the development of business applications. However, there will be some of the operational risk that may cause harm to the whole system, if it is not managed properly. Hence, these systems finds difficult to upgrade with the modern technology. On the other hand, there is a need to have efficient strategy to refactor [2] the system into modern one. Several approaches [3] have been proposed to transform these legacy systems, which are classified into redevelopment, wrapping, and migration. Subsequently, to transform these systems ,there is a need to understand and analyse the existing legacy software systems. Over the years, many researchers have proposed various techniques to analyse these systems. In order to understand the functionality of the existing legacy systems to cope up with the current business requirement for a particular domain it is necessary to automate the legacy analysis process to reduce time and cost efforts of the organisation. This paper helps to understand better the significance and challenges of legacy systems.

The Rest of the paper is organised as follows: section-2 discuss about significance of legacy system, Section-3 gives open challenges of legacy systems, and section-4 illustrates the work done so far in legacy

system understanding, section-5 is about Limitations in existing techniques and section-6 about conclusion and future work.

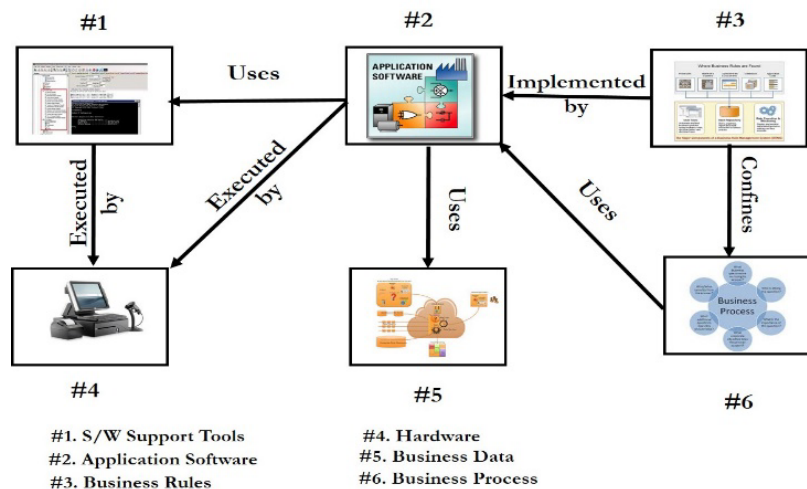


Figure 1. Outline of legacy system

## 2. LEGACY SYSTEM SIGNIFICANCE

Legacy Software systems are very large and complex as outlined in figure-1, which depicts, the software applications developed using Software tools that embeds all the business rules related to the business processes of an organization. These software applications process the business data through computer systems and generate information, which helps the management to achieve the goals and objectives of the organization. The Legacy software systems which are built to cope with the needs of business organizations must provide its functionality to run the business processes in a smooth manner, and should afford long-term benefits to the customers. These software systems embedded with all the daily routines of the organization. However, these systems provide insufficient information to the users, and they are not user-friendly. It is difficult to modify and replace these software applications with latest technologies. Moreover, refactoring or modernizing Legacy systems requires Business Processes Reengineering in an organization. The business will be influenced to a greater extent, if any change or failure occurs to that Legacy system.

## 3. MAJOR OPEN CHALLENGES OF LEGACY SYSTEMS

The Legacy systems pose several challenges for executing business processes in an organization. In particular

- Legacy systems are slow and expensive due to their obsolete hardware
- Maintenance of Legacy system software is costly because fault detection is expensive and time-consuming due to improper documentation, lack of knowledge about functionalities of the system
- Lack of interfaces provides greater difficulty to integrate with other systems.
- Evolution of legacy systems to provide new functionality required by the organization is virtually impossible

This paper analyzes the legacy systems. The analysis of the legacy code helps us to solve the issues such as upgrading the legacy system, effectively understanding the business rules, proper documentation for the existing undocumented code and finally managing the ripple effect. However, to migrate or refactor, maintain, and rewrite the existing system requires good understanding of the structure and its functionality.

## 4. REVIEW ON LEGACY SYSTEM ANALYSIS

Analysis of the legacy systems helps to understand the exact situation of the system and its entire process. It lays a great foundation for the further development, advancement or shift to the advanced technology.

Felix Cuadrado et al [2] proposed a technique to the recovery of the legacy system architecture which ensures proper documentation of the legacy system. For better understanding of the legacy system, the Author has extracted low-level details from configuration options, users' manual and source code. In addition to that high-level Details acquired from requirements or design documents. He used Different tools for recovery process, the selection of the tool depend on technologies, size, documentation, etc. The proposed recovery technique utilizes the QAR implement for documentation analysis, static analysis and dynamic analysis. For Information extraction, the prime resource used is user manual. They utilized a profiling implement for extracting runtime information and reverse-engineering implements like Jude and Omondo UML Studio to extract the static view of the system, which will automatically analyze Java source code, engendering UML class diagrams at class and package levels. The above implement detects inheritance and dependency cognations between elements. The Java Profiler of TPTP 4.2 tool was utilized for Dynamic analysis, which captures run time information- method invocation, execution time and number of instances in recollection. Higher caliber architecture was obtained from the abstraction process which consists of a series of filtering actions. The analysis was executed cumulating both the static and dynamic system views.

Rafik A. Salama et al [3] proposed a mechanism called Modernization Strategy Selection framework (MSSF), and Decision Making Tool, which analyses the source code to measure its quality. The source code selected was measured using the CCCC API. The author suggested a hybrid modernization approach, for the organizations to evaluate each component vs. service in a separate thread to focus on its criteria's values alone

Saad Alahmari et al [4] proposed methodology and efficacious guidelines for the identification of precise services from legacy code. They introduced Meta model that defines the characteristics of business processes and services types. The approach fixates on identifying these services predicated on a Model-Driven Architecture approach fortified by guidelines over a wide range of possible services types. They accentuated the importance of the classification of services types to define their properties correctly. They used UML activity diagrams that identifies coarse-grained services, BPMN business process diagrams were adapted to identify fine grained services, as well as coarse-grained services. (as composite service)

Grace Lewis et al [5] used SMART tool to analyze the legacy components for determining changes to be made to enable migration of legacy system. This tool uses three sources of information to fortify the analysis activity- First Information cognate to the issues, quandaries that were noted by the team in revelation process, second information provided by a Service Migration Inventory (SMI) that extract the many desired behavior of services executing within SOAs into a set of topics and third source of information entails the utilization of code analysis and architecture reconstruction tools to analyze the subsisting source code for legacy components

Smith, D et al [6] proposed a SMART technique with different behavior which assists organizations in analyzing legacy capabilities for use as services in an SOA environment. The goal of this activity is to obtain descriptive data like the name, function, size, language, operating platform, and age of the legacy components that are considered for migration. In the recovery process, Architecture code, complexity design paradigms, level of documentation, coupling, interfaces, and dependencies between components extracted from Technical personnel. In addition, the tool proposes to extract information related to quality, maturity of legacy components, problems, change history, user satisfaction, longer term needs, cost for maintenance and finally effort needed for these analyses.

Marchetto et al proposed [7] an approach consists in analyzing Legacy system application to understand the system architecture, and identify the functionalities. They proposed mechanism to recover the functionalities from textual use cases, functional requisites in natural language and user manuals. They have opted to represent the application functionalities and their relationships (e.g., extend and include) utilizing an UML Use Case diagram. When the size of the application is considerable, documentation not up to date, the Author suggested Reverse engineering techniques to represent the architecture of the application.

DinhKhoa Nguyen et al [8] proposed a novel business service engineering methodology - GAMBUSE that identifies and conceptualizes business services in a business domain, is based on a stratified reference Service Meta-Model (SMM) that specifies and correlates all modeling constructs for business processes. During this step, the Service Schema Specifications (SSS) of the as-is and to-be process that contain their activities, business entities, attributes, constraints, business rules, etc., are instantiated from the SMM.

He Yuan Huang et al [9] proposed approach to recognize the component(s) to be reused, evaluate the dependencies on other components, ascertain the depended components and produce legacy surrogates for depended components

Ravi Khada et al proposed two [10] mechanisms. One of them is reverse engineering which performs its actions by understanding and analyzing the source code. Other is knowledge based which performs its action by understanding the knowledge on domain and by analyzing the experience of the initial

promoters or developers. By understanding the end user experience with the help of the documents available / interview process, one can understand the applications of the legacy systems.

Vemuri, P et al [11] used theory of Feature analysis to detect the features from a legacy system which identifies the key features of the legacy system from end-users, the domain cognizance and expertise that exists in the legacy system team, to Identify test cases from the regression test suite that represent a particular feature or a group of features. They proposed code-profiling implements which produce internal product scripts that would trace the code traversal paths and frequency via different modules and packages of the legacy system, to execute the test cases identified for each feature and accumulate the metrics produced by the profiling implements. The metrics provide an understanding of un-used areas of legacy code, level of code catching that exists in the legacy modules per every feature

Table 1. Review of understanding and analysis of legacy system

S.no	Techniques	Specific Tools /Models used	Reference
1	Reverse Engineering, Architectural Retrieval, Documentation, User's Manual, Static Analysis, Dynamic Analysis, Generalization	QAR Jude and Omondo UML Studio Java Profiler of TPTP 4.2	Félix Cuadrado, García, nas, & Parada, 200
2	Source code exploration	CCCC API	Rafik A. Salama, Sherif G. Aly
3	Model Transformation, Portfolio Analysis	BPMN model	SaadAlahmari, David De
4	Interviewing, Process Modelling, Reverse Engineering Questionnaires	Use case diagrams Analysis Model	Roure, Ed Zaluska
5	Architectural Reconstruction		
6	Documentation, Source Code Exploration, ,Interviewing	SMART Tool	Grace Lewis et.al
7	Documentation , Interviewing, Source Code Exploration, Architectural Reconstruction	SMART Tool	Grace Lewis et.al 2007
8	Reverse Engineering, Documentation, Functional Requirements	Reverse engineering tools	Alessandro Marchetto
9	Service Schema Specification	Service Meta-Model (SMM)	DinhKhoa Nguyen et.al
10	System reconstruction, Documentation, Dynamic Analysis	Dependency Analyser	He Yuan Huang et.al
11	Knowledge transfer programs, Model-Driven engineering Utilizing the human feedback	Reverse Engineering tools	Ravi Khadka et.al
12	Test Cases, Feature Analysis	code-profiling tools	Prasad Vemuri et.al

In summary the Table-1 gives complete understanding and analysis of legacy system proposed by various researchers.

The Figure 2 depicts the issues of Legacy system over the span of two decades. It is observed that the documentation, system design requirements poses the challenges for the maintenance of Legacy system. From the Figure it is also observed that typical system design, poor documentation and change in requirements effects the vendor product due to lack of upgradation. The Figure 3 illustrates the comparison of various languages used to developed business applications with respect to software metrics such as reusability, cohesion, modularity, documentation and complexity of methods. It is observed that the languages such as PASCAL, COBOL, C, CPPSC provides more complex methods compared to the other modern languages such as JAVA, JSC, .NET AND PHP. When it comes to the reusability metric legacy code in C, CPP, and CPPSC will have more reusable percentage than COBOL, PASCAL. Further it is also observed from the Figure that modularity and documentation metrics in JAVA, C, and CPPSC will provide better analysis than COBOL, PASCAL and .NET

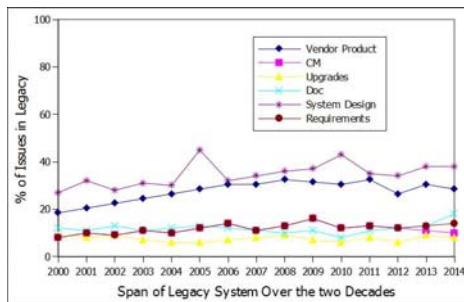


Figure 2. Issues in Legacy Systems

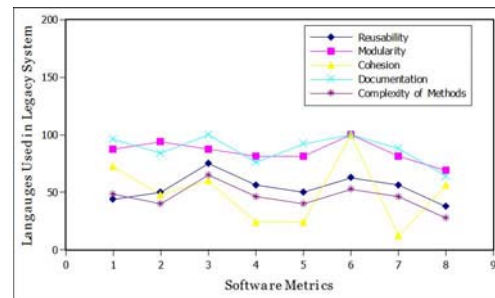


Figure 3. Comparison of languages vs software metrics

## 5. LIMITATIONS IN EXISTING TECHNIQUES

Over the last few decades researchers proposed several techniques for understanding the legacy systems applications. However, there is necessity to analyse these applications as there are some issues and challenges need to answer for effective management of business processes of organization. Subsequently Information about the design, architecture, its components, inter-relationships must be carefully preserved so that the legacy systems functions works effectively. Moreover it is necessary to solve the Knowledge emission problem, which can be overcome by updating documentation from time to time, conducting workshops for the experts in order to solve some technical issues on existing programming languages. Further more, the reverse engineering techniques are not completely automated for effective management

## 6. CONCLUSION AND FUTURE WORK

Software evolution is one of the challenging issues in today's business environment. It is necessary for the organizations, which make use of Information, and communication technologies will have to align their business processes to compete with global business. The existing large software systems ("legacy" systems) have never been built to cope with the current business requirement for their poor coding, design structures, logic and documentation. In this paper, we analysed various techniques proposed for understanding Legacy systems in existence. Subsequently, there is a necessity to have an effective tool for better understanding and analysis of legacy system. In our future work, we will analyse Legacy Systems using ANTLR (Another Tool for Language Recognition) tool, which is pro-dominant parser for analysing, studying, executing and translation of binary or text files. The key aspects of ANTLR, accepts large class of grammars and builds the parse trees for a given input program. Eventually, it is more worth full if ANTLR is integrated with multi-threading environment that creates multiple instances which takes multiple files as input and provide effective outputs for better analysis of legacy systems

## REFERENCES

- [1] Yang H., Zhan Cui, O'Brien, P. "Extracting ontologies from legacy systems for understanding and re engineering", Computer Software and Applications Conference, 1999. COMPSAC '99. Proceedings. The Twenty-Third Annual International conference.
- [2] Moeini A. Rafe V., Mahdian F. "An approach to refactoring legacy systems", *advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, Date 20-22 Aug. 2010
- [3] 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, pp. 53–62 (March 2010).
- [4] Cuadrado, F., García, B., Duenas, J., & Parada, H.A. (2008). "A Case Study on Software Evolution towards Service-Oriented Architectur". Paper presented at the 22nd International Conference on Advanced Information Networking and Applications – Workshops
- [5] Salama, R., & Aly, S.G. (2008). "A Decision Making Tool for the Selection of Service Oriented-Based Legacy Systems Modernization Strategies". Paper presented at the The International Conference on Software Engineering Research and Practice, Las Vegas, USA.
- [6] Alahmari, S., Roure, D.d., & Zaluska, E. (2010). "A Model-Driven Architecture Approach to the Efficient Identification of Services on Service-Oriented Enterprise Architecture". Paper presented at the Proceedings of the 2010 14th IEEE International Enterprise Distributed Object Computing Conference Workshops, Washington, DC, USA.
- [7] Lewis, G., Morris, E., & Smith, D. (2006). "Analyzing the Reuse Potential of Migrating Legacy Components to a Service-Oriented Architecture". Paper presented at the 10th European Conference on Software Maintenance and Reengineering (CMSR'06).

- [8] Lewis G., & Smith D.B. (2007). "Developing realistic approaches for the migration of legacy components to service-oriented architecture environments", Paper presented at the 2nd international conference on Trends in enterprise application architecture.
- [9] Marchetto A., & Ricca F. (2008). "Transforming a java application in an equivalent web-services based application: toward a tool supported stepwise approach". Paper presented at the 10th International Symposium on Web Site Evolution (WSE'08).
- [10] Nguyen D., van den Heuvel W.J., Papazoglou M., de Castro V., & Marcos E. (2009). GAMBUSE: A gap analysis methodology for engineering SOA-based applications. *Conceptual Modelling: Foundations and Applications*, 293-318.
- [11] Huang H.Y., Tan H.F., Zhu J., & Zhao W. (2008). "A Lightweight Approach to Partially Reuse Existing Component-Based System in Service-Oriented Environment". Paper presented at the 10th international conference on Software Reuse: High Confidence Software Reuse in Large Systems, Berlin, Heidelberg.
- [12] Ravi Khadka et al, "Legacy to SOA Evolution: A Systematic Literature Review"
- [13] Vemuri, P. (2008), "Modernizing a legacy system to SOA - Feature analysis approach." Paper presented at the TENCON 2008 - 2008 IEEE Region 10 Conference.

## BIOGRAPHIES OF AUTHORS



M. Srinivas received his MCA from IGNOU, New Delhi and M.Tech.degree in Computer Science and Engineering from Acharya Nagarjuna University and presently pursuing the Ph.D. in K L University, Guntur (Dist), AP, India. He is working presently as an Associate professor in the department of computer science and Engineering in KLU University. He is having 19 years Experience in Teaching in reputed institutions. His research interests include Software Engineering, Datamining. He is life member of professional bodies like CSI, ISTE. He taught various subjects in computer science and engineering like software engineering, computer organization, Database Management systems, Cryptography and network security, Digital Logic design, UML ,OOPS etc.



Dr. Ramakrishna Gavirneni currently working as Professor in department of CSE KLU University, Vaddeswaram. Carried out research at Saha Institute, Calcutta for about five and half years from 1966-1971 in theoretical physics using computers at ISI, Calcutta, IIT Khargapur, IIT Kanpur, & IIT Madras extensively for solving problems in nuclear models and Obtained Ph.D. in 1975. He carried out R & D and executed several projects at ECIL, Hyderabad from 1975 to 2003. Was trained on the state of art software for Supervisory Control & Data Acquisition (SCADA) Software Pipeline Leak Detection Software at LOGICA, LONDON, UK in 1983 Designed for ECIL a 2D Graphics Software Package for Printed circuit board layout at National Centre for Software Development & Computing Techniques (NCSDDCT), TIFR, Bombay from 1975-1977. Also he developed models (using computers) for analysis of lens design at NIT, Warangal which have been used in the dept. of Physics to carryout R & D Projects.



Prof. Kurra Rajasekhara Rao is a Professor of Computer Science & Engineering and presently working as Director, Sri Prakash College of Engineering (SPCE), Tuni. He worked at KLCE/K.L. University for 20 years as a faculty member in various positions as HOD of CSE, HOD of IT, Vice-Principal, Principal, K L College of Engineering (Autonomous), and Dean (Administration), Dean (Faculty & Student Affairs) Dean (Exams & Evaluation) of KLU. He is having more than 30+ years of teaching and research experience, Prof. KRR is actively engaged in the research related to Embedded Systems, Software Engineering and Knowledge Management. He had obtained Ph.D in Computer Science & Engineering from Acharya Nagarjuna University (ANU), Guntur, A.P. under the able guidance of Prof.P.Thirumurthy. He published more than 80 papers in various International/National Journals and Conferences. He produced 4 Ph.Ds and he is member of IETE, INSCA, IEL, ISTE, CSI



Mr. E. Suresh Babu received his B.Tech degree in Computer Science from RGM College of Engineering, Nandyal, M.Tech degree in Computer Science from V.T. University Belgaum and pursuing PhD in Computer Science & Engineering from J.N.T. University Kakinada. Currently, he is working as an Associate Professor in the Department of CSE in K L University Vijayawada; He has 12 years of teaching experience. He has published 8 research papers in various International Journal and 6 research papers in various National and International Conferences. He has attended 32 seminars and workshops. His areas of interests are wireless communication and MANETs, Security, Mobile Sensor Networks.