

INVESTIGATING THE LIFE CYCLE OF LEGACY SYSTEMS MIGRATION

Teta Stamati, Information Systems Group (ISG), Department of Informatics and Telecommunications, University of Athens, Greece, teta@di.uoa.gr

Drakoulis Martakos, Information Systems Group (ISG), Department of Informatics and Telecommunications, University of Athens, Greece, martakos@di.uoa.gr

Abstract

Although legacy systems migration as a subject area is often overlooked in favour of areas such as new technology developments and strategic planning of information technology, most migration projects are considered ill-fated initiatives and a rate of over 80% of these projects run over budget, frequently with system functionality falling short of contract. The main objective of this work is to present a methodological approach that defines the landscape, deals with the semantics of legacy migration and can be applied by organisations that recognise the need to manage the process in a controlled and not in a piecemeal and ad hoc fashion.

Keywords: Legacy Systems, Migration, Life Cycle, Action Research, Petri nets.

1 INTRODUCTION

The term “legacy system” describes an operational system that has been designed, implemented and installed in a radically different environment than that imposed by the current information and telecommunications technology (ICT) strategy. Although these systems are often built over many years by a combination of IT and business experts, they remain in operation within the organisational context and represent a massive, long-term business investment composed usually by extremely efficient robust and finely tuned applications. The common case is that the legacy systems, especially these based on mainframe platforms, are the mainstays of today’s businesses.

Several alternative definitions of what exactly a legacy system is can be retrieved. Ulrich (1994) defined them as “stand-alone applications built during a prior era’s technology, but they are perhaps more widely understood as software systems whose plans and documentation are either poor or non-existent (Connall and Burns, 1993)”. Bennett (1995) referred to the legacy systems as “large software systems that we do not know how to cope with but that are vital to the organisation”, while Brodie and Stonebraker (1995) as “any information system that significantly resists modification and evolution to meet new and constantly changing business requirements”. In all the aforementioned definitions, often the notion of “something valuable” is present, as well as the notion of “old and obsolete” (Lauder and Lind, 1999).

It is unambiguously recognised that legacy systems are crucial for the operation of organisations and thus they are essential for our economical and well-fare activities. The common rule is that if the legacy systems cannot support the business requirements, the business will not be able to remain competitive for long (Brodie and Stonebraker, 1995). Thus, a significant budget and person-hours are monopolised by legacy systems maintenance. De Palma and Woodring (1993) referred to over 40% of the IT costs within an organisation being spent on maintaining its legacy systems, while Brodie and Stonebraker (1995) considered that the process of keep these systems running takes 80-90% of the IT budget. Slee and Slevin (1997) gave estimation in the area of 80% just for the routine maintenance activities.

The common approach for an organisation to the legacy problem is migration of its mission critical legacy systems. The definition of a successful information system migration according to Brodie and Stonebraker (1995) is as follows: “it begins with a mission-critical legacy system of a significant size in full operation and it ends with a fully operational, mission critical target application (or applications components) that replace the essential aspects of the original legacy system”. This involves replacing the problematic hardware and software, including the interfaces, applications and databases that compose an information system infrastructure. Brodie and Stonebraker (1995) claim that legacy information system migration involves starting with a legacy information system and ending with a comparable target information system. This target system is significantly different from the original, but it contains substantial functionality and data from the legacy system.

Within this context the main objective of the present work is to discuss some critical issues as far as the life cycle of a legacy systems migration is concerned. The remainder of the paper is focused on the proposed life cycle model that is the result of research carried out by the authors in the context of action research that took place throughout the deployment of a significant project of systems migration for the Greek Government. The three-step spiral process of planning that involved reconnaissance, taking actions and fact-finding about the results of the actions has been applied within the development life cycle of proposed model. Using a spiral process we alternated between action and critical reflection and afterwards we continuously refined the methodology using the data and interpretation of the understanding developed in the earlier cycles. The model considers that the migration process should be analysed based on two perspectives: technological and business, and each one of these offers either qualitative or quantitative criteria. The model considers as well different categories of users, the users with IT know how and the managers and account for the “different-perspectives” phenomenon. Two distinct phases are identified within the suggested model and are as follows: Migration Assessment and Migration Deployment.

2 LEGACY SYSTEMS MIGRATION LIFE CYCLE

Traditionally, legacy information systems are considered from a technical perspective, addressing issues such as age, complexity, maintainability, design and technology (Holland & Light, 1999; Kavakli and Loucopoulos, 1999). Within this context, legacy systems migration has been regarded as the replacement of problematic systems infrastructure with newer, more modern hardware and software in order to achieve effective access to corporate data. From a modern perspective, the validity and applicability of a migration process has been broadened as enterprises realise the strong impact of information technology and systems employed, on the way they do business (Kavakli and Loucopoulos, 1999). (Kelly et al., 1999) describe the ‘business legacy’ which is embedded in the legacy information systems and it is the inter-relatedness of business and information systems legacy that makes either Business or Systems Change a very complex process. Considering this broader definition of systems migration, (Stamati et al., 2004) describe a migration initiative as a process, which involves business change and it is more than just the movement or reorganisation of database systems, application programs and program interfaces. In addition, (Stamati et al., 2006) consider that although many practitioners have realised the aforementioned shift from the conventional definition of migration to a more effective consideration, they describe experiences that the proposed theoretical migration approaches are myopic and do not take into account a number of either overt or covert factors that make a migration project a really complex initiative.

Considering the aforementioned, the following paragraphs present a case of a mission critical migration initiative revealing some significant factors that should be seriously considered by both migration engineers and business experts during the life cycle of a legacy systems migration initiative. The case described is based on a spin-off migration project for the Supply Chain Management (SCM) Department of a Greek Public Organisation (hereinafter referred to as “the Organisation”).

For reasons of confidentiality, there is no reference to the name and nature of the Public Organisation and also some details are made somewhat more generic, but the conclusions drawn are valuable and enhance the reasoning for the proposed methodological approach.

2.1 The Context of the Case

The Organisation's SCM Department was involved in the in-house development of the legacy system for more than thirty years. The system formed the central hub and the backbone of the information flow within the Department and was the main vehicle for consolidating information about the supply chain management processes. The old system was built in COBOL with an obsolete Command Line Interface and the data were stored in Index (flat) COBOL Files. Following a time-consuming procedure of selecting an expert company in migration initiatives, the Organisation signed the contract with a specialist firm (hereinafter referred to as "the Contractor").

The Organisation's legacy system was posing some important problems to the host Department. Following an in-house study and analysis, the Organisation identified a significant number of key drivers for having the system migrate and evolve, namely:

- The system was too large, with millions of lines of code and there was a significant lack of documentation. Maintaining the software was an expensive task and the process of finding and correcting system faults was also costly and time consuming due to a growing lack of understanding of the internal functionality of the system.
- The system was written in COBOL which is considered a legacy language. There was a significant loss of institutional knowledge as the original developers were retired. The developers were facing a skills shortage as the academic institutes had already stopped teaching the COBOL.
- The system could not evolve to accommodate new functionality that was required by the Organisation.
- The Department attempted to redefine its strategy from a traditional data processing model to a multi-channel, service oriented model.
- Legal requirements and regulations for the Public Sector were under investigation and experience showed that potential changes would not have taken into account the IT system characteristics.

Throughout the life cycle of the migration initiative, it is noteworthy that these initial drivers evolved during the execution to constraints hindering the smooth evolution of the legacy system. Considering these factors that were ultimately apparent as project constraints, the project steering committee decided to proceed with the following technical scenario: separation of the whole code into functional groups, removal of the redundancies and keeping the core applications blocks in COBOL, consequently the engineering team would build new functionality around these core COBOL applications and link them with a new relational database and windows-based interface and reporting tools.

Although the project was delivered to the Organisation six months later than the initial time frame and the new system involved significant core parts of the legacy system, the project's committee accepted the new system as an operational system. Furthermore, although the new target system consisted of some "old" parts of the legacy system (COBOL applications) preventing potential further enhancement and evolution of the target system, the project's steering committee decided that the new system covered the Department's current needs. Upon the project completion, the feeling that the project committee had was that a significant number of problems could have been avoided if a well-defined methodological action plan had been applied.

2.2 A Methodological Framework for Legacy Systems Migration

Due to the complexity that a migration initiative contains, a methodological action plan is required (Stamati et al., 2005). The practitioners must consider that the technology drive should be present, but should not be overestimated (Kavakli and Loucopoulos, 1999). In our case, a more careful evaluation of the “as-is” situation in the initial mainframe environment was required. Relevant action items were, among others, the evaluation of the pre-existing COBOL code and the underlying data structures. Determination of the “to-be” business and application architecture along with the corresponding “to-be” technical and deployment architecture were essential for the completion of the project. Furthermore, critical evaluation of the migration scenario between the “as-is” and the “to-be” situations should have been further analysed and designed in depth.

For any migration process to be accomplished, a well-structured, conscious plan needs to be specified. A number of predefined procedures linked between them represent the transformation of a current organisational state to a desired future state. In the proposed model these procedures are described as two self-contained interoperating phases. The model does not represent some form of magical tool, to achieve migration, but instead it provides a framework that can be used for guidance throughout the life cycle of a migration initiative.

In general, the motivation for any migration process is the incremental transition from an initial organisational situation A, which is unsatisfactory in some aspect, to a desired situation B where the problem is addressed. Possible causes to such change include perceived opportunities, threats, social pressures or political decisions, including for example the opportunities offered by new technologies, increased customer demand for better service quality or the globalisation of markets. The main objective of our approach is the provision of a model based on a set of Knowledge States that define a migration process. We approach migration not as an undirected process, but as a purposive activity driven by specific goals. Hence, its effectiveness depends on being able to make good decisions about what migration goals to pursue, on selecting the appropriate strategies for achieving the desired goals, and on guiding the application of the chosen strategies. Within the proposed model, it is possible to make the distinction between four different types of Knowledge States involved in a migration process, namely:

- Knowledge about the current business processes and the legacy systems that serve these processes (As-Is).
- Knowledge about both the Legacy Systems and Business Assessment (Initiation State).
- Knowledge about the validity of a set of proposed migration transformations and thus, knowledge about the most suitable proposed plan (Migration Scenario State).
- Knowledge about the selected migration solution (To-Be State).

Two additional states in the model could be available: the Null State and the Target State. These describe respectively the state where ‘no knowledge’ about the migration is available and the state where ‘enough knowledge’ has been obtained.

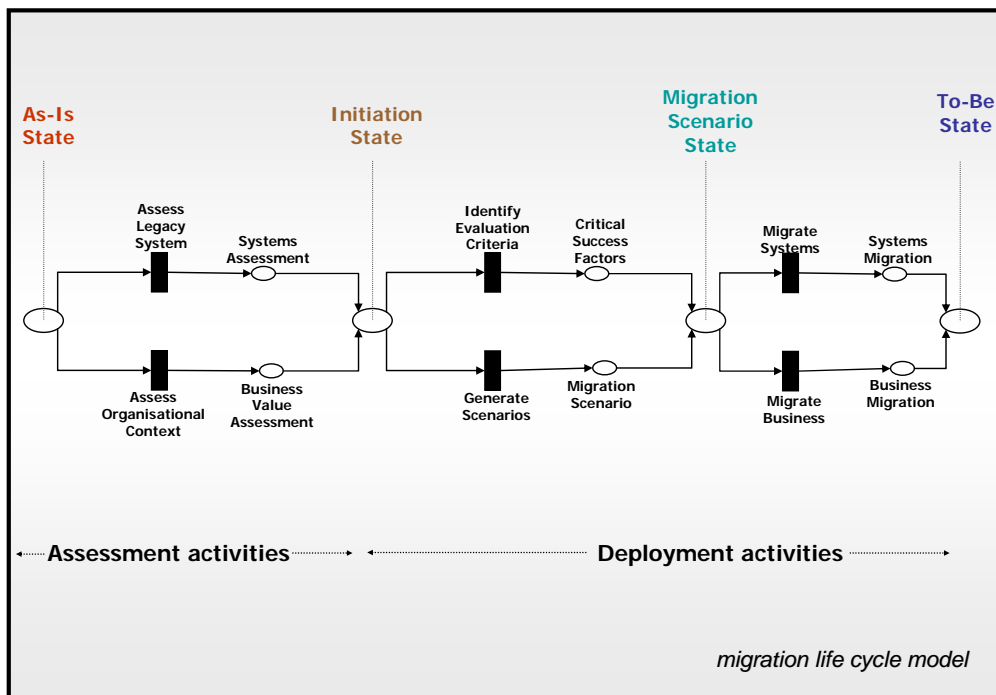


Figure 1. Migration Life Cycle Model.

The Migration Life Cycle Model is constructed in the spirit of Petri nets (Peterson, 1981), whereby places in the net (represented by circles) correspond to knowledge states whilst transitions (represented by rectangles) correspond to strategies. The directed nature of the graph indicates the way one may progress from one knowledge state to the next.

A more detailed analysis of the transition from a state to the next state will be discussed in the following sections.

2.2.1 The Transition from the As-Is State to the Initiation State

Any type of change whether it involves the development of a computerised system or the reengineering of business processes involves many assumptions about the embedding enterprise domain (Kavakli and Loucopoulos, 1999). Poor understanding of the domain is a primary cause of project failure. In order to obtain deep understanding about an enterprise, one needs to understand both the Organisational Context and the Legacy Systems. This is the initial step that starts with the As-Is State and ends with the Initiation of the Migration process. Reaching a decision about how best to evolve a legacy system cannot be made spontaneously; rather it should be based on an assessment of all relevant system attributes (Sommerville, 2001). System assessment is used to gain an understanding of a legacy system, which is fundamental to any system evolution exercise. System assessment should be an initial activity for evolution projects.

We want to stress that understanding the organisational context involves more than understanding the legacy information system. It is imperative to obtain knowledge about the existing enterprise goals and how they are achieved through the current enterprise behaviour as well as acquire knowledge about the functionality of the legacy system to the degree necessary for the migration.

2.2.2 *The Transition from the Initiation State to the Migration Scenario State*

Since different migration approaches are appropriate in different information systems a carefully thought decision on the particular migration solution should be chosen. The model proceeds to define a structured way of identifying and evaluating the alternative transitions for legacy systems migration. The identification and evaluation of specific migration plans is called the Scenarios Generation Process. The model adopts the scenario generation techniques because we believe that scenarios are widely used in industry and their usage during a change process is very helpful. Scenarios are defined as possible alternative transitions for change. They are built on current observations and aim at identifying various future states.

The success of a migration effort has to be measurable (Sommerville, 2001). Any effort for migration or evolution of large-scale industrial software systems must be driven by a corporate strategy redefinition. Buy-in from management and end users requires a well-defined strategic management view and planning, based on a clear statement of motivation, objectives and of the constraints imposed by the current environment and continuous operational needs. An insight of the role of both drivers and hindrances and their inter-relatedness should be carefully considered. These key factors must mirror the successful implementation of a migration project. The definition of measurable success factors is essential. It is important that the key factors during a migration initiative ensure the requirements and specifications satisfaction and that the assertions and the detailed requirements will be done jointly. The organisation-wide methodology must be defined and supported by all stakeholders. Executive support, clear vision, well-defined business objectives, good project management, definition of a formal process, firm basic requirements, skilled staff and user involvement are considered as imperative factors for the success of such projects. Acceptance criteria must be initially defined considering the required functionality, the test cases and the performance targets. User management commitment is essential as well. Managers must be results-oriented, act as internal champions for the migration project and be committed to a user/developer partnership model. Finally, quality assurance and performance metrics should be considered as key factors.

2.2.3 *The Transition from the Migration Scenario State to the To-Be State*

It may appear to be late in the day for the design and development of the target system. However, it is not until this stage that sufficient information is available upon which the design may be based. The design and development of the target system can range from the very simple to the very complex. This involves the actual migration process for both the legacy systems and the business processes.

3 CONCLUSION

The concern of migration has shifted in order to include both organisational change as well as change in computer systems that enable such enterprise change. Such a movement has led to the emergence of new definitions that put emphasis on a broader process of migration, and include the cognitive, social and technical context of migration. These definitions are based on the premise that the replacement of a software system in the organisation inevitably brings about change in the way work is organised. In a similar manner, any organisational change should be reflected in the software system requirements. Therefore, migration is concerned both with the design and implementation of a new software system and the management of change in the business systems that might be supported by it.

Acknowledgements

The project is co-funded by the European Social Fund and National Resources – (EPEAEK II) PYTHAGORAS.

References

- Bennett, K.H.. 1995. 'Legacy Systems: Coping With Success'. *IEEE Software*, 12(1): 19-23.
- Brodie, M.L. & Stonebraker, M. 1995. 'Migrating Legacy Systems'. *Morgan Kaufmann Publishers*, London, 2nd edition.
- Connell, D. & Burns, D. 1993. 'Reverse Engineering: Getting a Grip on Legacy Systems'. *Data Management Review*.
- Holland, C., P. and Light, B. 1999. 'Focus issue on Legacy Information Systems and Business Process Change: Introduction'. *Communications of AIS*, 2(9), 98-120.
- Kavakli, V., and Loucopoulos, P. 1999. 'Modelling of Organisational Change using the EKD Framework'. *Communications of AIS*, 2(6).
- Kelly, S., Gibson, N., Holland, C., P., and Light, B. (1999). 'A Business Perspective of Legacy Information Systems'. *Communications of the Association for the Information Systems*, 2(7).
- Lauder, A. and Lind, M. 1999c. 'Legacy Systems: Assets or Liabilities?'. LAP'99, Copenhagen.
- Peterson, J. L. (1981). 'Petri net theory and the modelling of systems'. *Englewood Cliffs; London, Prentice-Hall*.
- Sneed, H.M. 1995. 'Planning the Reengineering of Legacy Systems'. *IEEE Software*, 24-34
- Sommerville, I. 2001. 'Software Engineering'. Pearson Education.
- Stamati, T., Stamati, K., and Martakos, D. 2006. 'Key Factors in Legacy Systems Migration: A Real Life Case'. *8th International Conference on Enterprise Information Systems (ECEIS2006)*, to be appeared.
- Stamati, T., Kanellis, P., Stamati, K., and Martakos, D. 2005. 'Migration of Legacy Information Systems'. *Encyclopedia of Information Science and Technology (EncyclopediaIST)*, 2nd edition, USA.
- Stamati, T., Kanellis, P., Stamati, K., and Martakos, D. 2004. 'Legacy Migration as Planned Organizational Change'. *6th International Conference on Enterprise Information Systems (ICEIS)*, 3(2): 501-508.
- Ulrich, W. 1994. 'From Legacy Systems to Strategic Architectures'. *Software Engineering Strategies*, 2(1), 18-30.