

Model Driven Legacy Migration

Patrick DJ Kulandaisamy (patrickj@infosys.com)
Comfactory, SETLabs
Infosys Technologies Limited, Bangalore, India

Abstract

Most enterprises depend on information technology to execute their business processes. This dependency on IT has evolved over past few decades with technology deployments such as mainframe, client/server, desktop applications, distributed component systems, message based integration, and package solutions. IT managers and CIOs who manage these heterogeneous environments face difficulties ranging from non-availability of skilled people to technology incompatibility. In such conditions their choice is to modernize these legacy systems. But many legacy modernization projects in the past have failed to achieve modernization goals. The failure to deliver the promises of legacy modernization can be attributed to lack of systematic migration approach. This paper prescribes model driven legacy migration, which uses variety of models to define, analyze, and execute legacy migration projects.

Introduction

Information system architecture has three generic software layers – system software, platform infrastructure software, and application software. The *System software* is related to hardware infrastructure and comprises of operating systems, network protocols, device drivers etc. *Platform Infrastructure software* is about application execution environment. It includes application servers, transaction/component managers, database servers, and workflow engines. This layer provides essential set of services for business applications. *Application software* is specific to organizational business processes. This layer needs continuous enhancement and alignment with business processes. It is written using high-level programming languages and makes use of the services provided by platform infrastructure software and system software.

This paper elaborates on these layers from migration perspective. Migration models discussed in this paper define elements of information system layers and the best practices in modernization.

Migration Models

Migration is a process that transforms an existing legacy application into a new application. Models bring in clarity, consistency, and technology independence to any process. Combining these two concepts, migration models streamline the process of legacy migration through set of models. These models account for various layers in information systems and act as the primary driver in executing migration projects. Short descriptions of these models follow.

Application Aspects Model: This model defines an application as a container of source code, domain concepts, architecture, and life cycle processes. The prime reason for this model is, to think beyond the code-base in migration project execution. Profiling a legacy application with this model enables accurate selection of modernization strategy.

Data Layer Model: This model defines persistence layer of the legacy system in terms of storage management practices, connectivity mechanisms, tools, and schema. Most legacy systems use file system based persistence, which is not structured and accessible. Data layer modernization is concerned with optimized data structure selection, efficient storage management, flexibility in connectivity and retrieval

methods, and deploying right toolset for data management. The data layer model gives an opportunity to think on these lines while migrating.

Platform Model: The models defined above capture two vital parts of an information system. But a migration project also needs to capture other essential parts that host, serve, manage, and monitor the legacy systems. The platform model captures this information as set of services leveraged by the legacy system from the environment. This model provides uniform definition and unambiguous understanding of both legacy and target platforms.

Migration Map: Migration map is a model that ensures completeness of legacy migration. It describes migration as set of activities, methods, and tools that transforms each element in the source model to an equivalent element in the target model.

Rest of this paper elaborates model driven legacy migration approach, provides details of migration models, and highlights the benefits of this method.

Model Driven Migration Approach

Migration projects normally deal with two different architectures – the monolithic legacy system architecture and the componentized target architecture. The migration process is about effectively breaking the application in legacy architecture and transforming into an application that fits into target architecture. Migration models defined above are technology neutral and hence can capture the elements of legacy and target environments.

Thus, model driven migration can be defined as a process that captures source and target environment with respective models, builds a transformation map for every element in the models, identifies methods-frameworks-tools for migration, and finally formulates an execution plan. Pictorial view of model driven migration approach is given below.

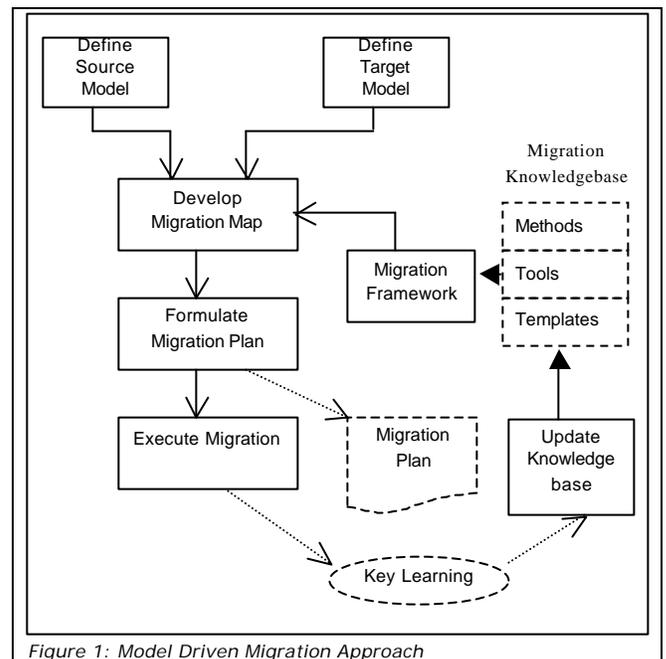


Figure 1: Model Driven Migration Approach

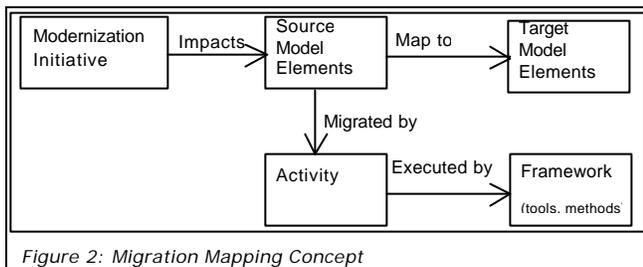
Define Source and Target Models:

Migration models bring uniformity in defining legacy and target environments. With the knowledge of migration models, the first activity is to build source and target environment models. These models capture two different technology environments and their key concepts on comparable terminology.

For example, if the migration initiative is VSAM to DB2 conversion, data layer models for VSAM file system and DB2 database environment will be built. The emphasis here is to map the native terminology to generic concepts. In this example, the generic element 'Entity' in data layer model equates to 'File' in VSAM and 'Table' in DB2. Similarly, 'Data Access' element maps to 'File IO routines' in VSAM and 'SQL Interface' in DB2.

One pair of source-target models will be developed for identified segments of migration impact. For example, Mainframe to J2EE migration will require application, data layer, and platform model pairs, as the impact is on the complete environment. Well-defined source and target models aid in deeper understanding of migration complexity and quicker development of migration map.

Develop Migration Map: Migration map is a fine grain road map for executing a migration project. Source-target models and migration strategy assist in identifying the impact elements of migration. From the source-target models, the transformation to be applied on source elements is evident. For example, in VSAM-DB2 conversion, File-IO routines should transform into SQL calls. Activities for a specific transformation with associated methods, tools are available in migration knowledge base. With this set of information migration map shall be developed. A conceptual view of migration map development is given below.



Execute Migration: The migration map feeds planning and scheduling phase, where migration activities are estimated and assigned. Execution step carries out the migration and uses migration framework. Key learnings from the project are captured and the migration knowledge base is updated.

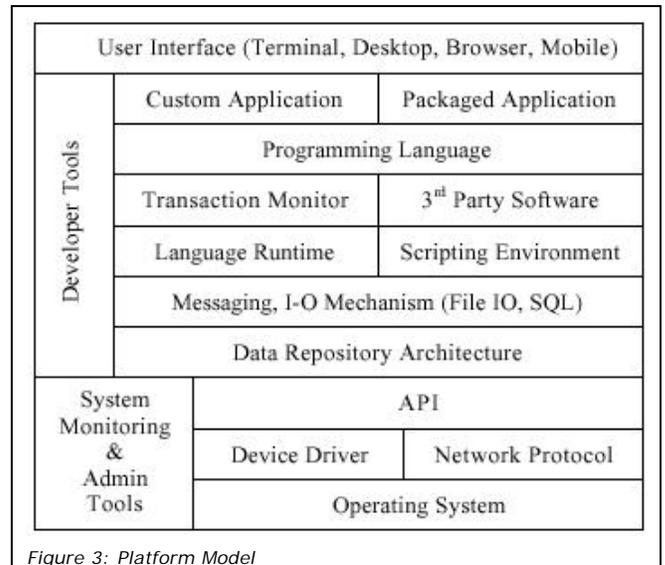
Migration Model Elements

This section provides element details of migration models. The elements in these models guide migration experts to think on various perspectives and ensure completeness of migration.

Platform Model

A platform is defined as a collaborative and cohesive set of components and services arranged in layers, starting from operating system at the bottom to user interface at the top. This model is useful, when the migration focus is to retire an existing hardware platform. Figure 3 below depicts the platform model.

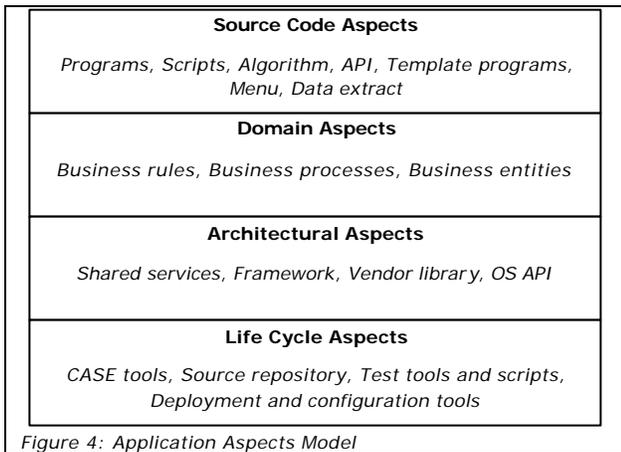
This model accounts for overall environment of the information system. It captures various services, unique features leveraged from enterprise platforms, and the tools available in the environment.



The platform model highlights the need for greater planning in migration. For example, in a mainframe migration initiative, organizations may concentrate on legacy source code migration and may fail to recognize unique developer tools available in mainframe, may fail to verify the availability/equivalent of 3rd party software at target environment and so on. By building a platform model, all these components get accounted and lead to identifying equivalent components in the target platform, otherwise the migration map will be incomplete!

Application Aspects Model

Similar to the big picture a platform model gives, application aspects model defines various elements for an application. It is suggested that, a business application be considered as an aggregation of domain concepts and processes implemented using a programming paradigm over a robust architecture. Life cycle processes that form the backbone of application evolution should also be include in this view. Application aspects model defined on this basis is given below. Each aspect group identified here needs special attention in migration projects.



Characterizing an enterprise application with these elements helps in effective profile analysis and in deciding best suitable modernization strategy for the application. Application profile analysis captures and categorizes application properties into three quotients - business, architecture, and operations. The *Business quotient* reveals the fitness level of the application to the business processes. Domain elements in the application model directly contribute to this measure. The *Architecture quotient* provides technical quality of the application environment and how good it aligns with evolving technology standards. The architecture profile parameters measure portability, accessibility, and reusability. The *Operations quotient* is about ongoing enhancement and support needs of the application. Combination of code base parameters and life cycle parameters in application model define this quotient.

The constituents of domain and architecture aspects highlight the useful knowledge asset the application possesses. This helps in applying appropriate extraction methods. The elements of life cycle aspect define operational state of the application and help in defining equivalent operational processes at the target environment, after migration.

A matrix of these quotients, guide in choosing candidate modernization option.

Business Quotient	Architecture Quotient	Operations Quotient	Modernization Option
High	High	High	Maintain and Enhance

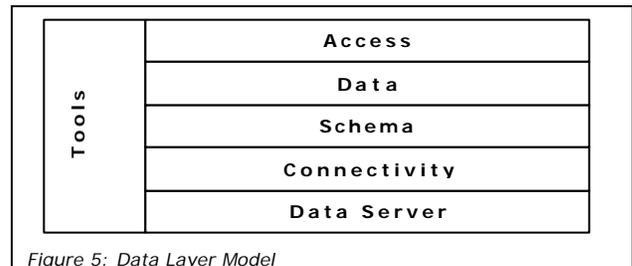
High	High	Low	Enhance team skill
High	Low	High	Integrate Web Enable
High	Low	Low	Refactor Replatform
Low	High	High	Biz – IT alignment
Low	High	Low	Enhance team skill Biz - IT alignment
Low	Low	High	COTS / Rewrite
Low	Low	Low	Retire

Table 2: Application Profile Analysis

Building application aspect model and profile analysis of it lead to modernization option selection, which defines the migration scope, and guides in activity listing in the migration map.

Data Layer Model

Data is one of the vital parts in any enterprise application. In a portfolio of enterprise applications, the data is shared across to deliver useful business value. The need for on demand data availability is rising, but obsolete data storage methods are becoming bottlenecks in delivering it. In persistence layer migration projects, a data layer model that contains following groups should be considered.



Access: Data stored in persistence layer can be manipulated in many ways. It includes methods such as ANSI's structured query language, using vendor extensions to the SQL such as library functions, advanced 4GL programming languages such as PLSQL in Oracle, and proprietary access methods such as COBOL call routines. Capture all the methods supported/used in the legacy environment and those available in target.

Data: This group focuses on the core business data and its management. It highlights the need to consider how history data is archived, how the data versions are managed, and the encryption methods for business sensitive data.

Schema: Elements in schema group capture data layer design. Data type support, entity-relationship definitions, index definitions are addressed in this group. In migration perspective, it highlights unique data types used, and normal form compliance of the schema. It also accounts for the user management and access rights definition that control data access.

Connectivity: This group primarily focuses on connectivity mechanisms to data layer. It identifies the drivers supported, connection pooling services offered, and native interfaces provided by the data layer. Capturing these parameters convey the potential impact in source code segments.

Data Server: Elements in this category capture data management software details. The basic information gathered is whether a database management system (DBMS) is used or file system is used. It includes elements for capturing the transaction services, distributed transaction facilities and any unique features/runtime libraries offered by the database vendor.

Tools: Tools play important role in managing the data environments. There are many tools for administrative functions such as backup/restoration, monitoring data growth, optimizing the query performance, managing users and so on. Programmers also use tools to run quick reports on data server, to generate schema from the data definition models, and to analyze data retrieval issues. These productivity improvement tools are very important and should be considered in migration project.

Accounting for all these components in data environment is crucial to the success of data layer migration project. Data layer models for legacy and target data layers benefit the migration team by exposing potential incompatibility issues well in advance.

Migration Map

The migration map is a blue print for execution. It defines which elements in legacy system need transformation, what are the activities to execute the transformation, and which methods/frameworks/tools in the migration knowledge base are applicable.

Following table gives a partial migration map for VSAM to DB2 migration.

E-to-E	Activity	Method	Tool
File-to-Table	Analyze file structure	Hierarchy analysis	
		Attribute grouping	
	Refine Fields	Redundant field elimination	
		Normalization	
		Field resequencing	
	Define Table		E-R Model tools
Field-to-Column	List source data types	Schema scanning	
		Program variable analysis	Source analysis tools
	Categorize data types	Length and precision analysis	
		Business term analysis	
		Min value max value method	
	Define target data types		
FileIO-to-SQL	List access methods	Language verb analysis	
	Classify IO access	CRUD matrix method	Source analysis tools
	Identify SQL equivalents	Language translation	
	Define access components		

Table 1: Migration Map Sample

E-to-E column refers to impact elements that need transformation. The Activity column captures the sequence of tasks that achieve the transformation. While activity specifies what to do, the Method column prescribes engineering ways of doing them. There are many methods for an activity and most migration projects use a combination of these methods. Tools column identifies options for automation and efficiency.

Benefits of Model Driven Migration

1. Migration models account for every element of platform, application and data layer. They offer much-needed big picture in a migration initiative
2. Migration model elements address all the stakeholders of the enterprise – developers, end users, administrators, network managers, DBAs, and vendors. It gives an opportunity to consider migration from everyone's perspective
3. Migration mapping is an excellent tool to think through the migration upfront. It clearly defines migration activities and hence project estimates and timeline will be accurate
4. Migration models and migration map defines how each piece of source environment transforms into target environment
5. Migration map ensures that a migration decision is taken for every element in the legacy system
6. Tools and life cycle elements capture and highlight the need to train IT team in target environment tools and processes
7. The knowledge base captures organization's migration expertise. It serves as expert reference of how migration projects were executed in the organization and can be leveraged in other migration initiatives
8. Platform, application and data layer models are valuable documents about enterprise systems
9. Migration models are extensible and shall accommodate organization specific elements

10. In short, combination of these models offer estimation and execution benefits for legacy modernization projects.

Conclusion

Overall complexity understanding is a key to the success of migration projects. These migration models are an attempt to define formalism in migration. They help in project scope definition, timeline estimation, and execution process formulation. The migration knowledgebase promotes reuse culture in the organization. The model driven migration method induces proactive planning by recommending the migration map development upfront. This method gives the IT team an opportunity for well-planned migration.

Future Work

It is necessary to define these migration models in standard meta-modeling notations such as UML. Extensive work is required in defining element level models – such as business rule model to represent the knowledge recovered from legacy systems.

Author Profile:

Patrick Kulandaisamy is a Senior Technical Architect with Software Engineering and Technology Labs at Infosys Technologies Limited, Bangalore, India. He has more than ten years of work experience in legacy system maintenance and component based application development. Patrick can be contacted on: Tel: +91-80-51164247 email: patrickj@infosys.com.

Infosys Technologies Limited © 2004. All rights reserved.

The copyright for all material in this article is the property of Infosys Technologies Ltd. This article is protected by the copyright laws of India and international treaty provisions. No part of this article may be reproduced or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, or stored in any retrieval system of any nature without the prior permission of the copyright owner, in writing. Infringement of this condition will lead to civil and criminal prosecution.