

Model Driven Legacy Migration

Presented in

OMG's Architecture-Driven Modernization Workshop

March 22-24, 2004

Patrick DJ Kulandaisamy
Infosys Technologies Limited
Bangalore, India

Information System Layers

Application Software

Business process services

Components and programs for business logic

Platform Infrastructure

Component execution environment

Offers Frameworks, Standards, Life Cycle Services

System Software

Operating systems and Network protocols

Legacy Information Systems Characteristics

Home-grown applications with millions of lines of code (undocumented)

Functionality overlaps lead to redundant code-base (requires reuse initiative)

Technology evolution and standards compliance is low

Proprietary runtime infrastructure with limited APIs for external connectivity

Interoperability with modern distributed architectures is complex

Availability of skilled programmers in legacy languages is reducing

Dependency on legacy systems is high for business continuity

Legacy environment is robust and handles large volumes of data

Legacy Migration Methods

Manual Redevelopment (Past)

Discard existing system and start fresh development

Tools Based Modernization (Present)

Quick fix modernization such as web enabling

Extract logic segments and wrap them as components

Model Driven Migration (Future)

Capture legacy system assets through models

Migration through transformation methods for model elements

Migration Models

Application Aspects Model

This model defines application as a container for source code, domain processes and entities, and architectural services

Data Layer Model

Persistence layer is more than schema and data – this model defines them all

Platform Model

Captures application environment in the view of multiple stakeholders including administrators and developers

Migration Map

A blue print for migration. It defines migration as transformation of model elements through activities, methods, frameworks, and tools.

Application Aspects Model

Concept

Business application is an aggregation of the following

- Source Code Elements
- Domain Elements
- Architecture Elements
- Life Cycle Elements

Benefits

Enables effective profile analysis of the application

Guides in selecting modernization options

Broad focus on application assets than source code alone

Online	Batch	Reports	Menu	DataEx
Programs			Scripts	
Template Programs			Algorithm	
Persistence API			Integration API	
Biz Rules			Biz Processes	
Business Entity				
Shared Services				
Architecture and Design				
Framework			Vendor Library	
Data Structure			Data Types	
Compiler and Language Extensions				
Session Management and Transactions				
OS API				
IDE			Case Tools	
Source Repository				
Build Tools and Scripts				
Testing Tools			Test Scripts	
Deployment Tools				

Application Aspects Model Based Profile Analysis

Method for analyzing legacy applications and deciding on modernization road map

Business Quotient

Application alignment and fitness to current business processes

Domain Elements in the model reveal this

Architecture Quotient

Compliance with evolving technology standards is the measure

Architecture elements directly measure this

Operations Quotient

Ease of enhancement and support

Source code elements and life cycle parameters contribute to this measure

BQ	AQ	OQ	Modernization Opt
✓	✓	✓	Maintain/Enhance
✓	✓	✗	Enhance Team skill
✓	✗	✓	Integrate Web Enable
✓	✗	✗	Rehost Platform Migration
✗	✓	✗	Biz IT Alignment Enhance Skill
✗	✓	✓	Biz IT Alignment
✗	✗	✓	Rewrite / COTS
✗	✗	✗	Retire

Data Layer Model

Concept

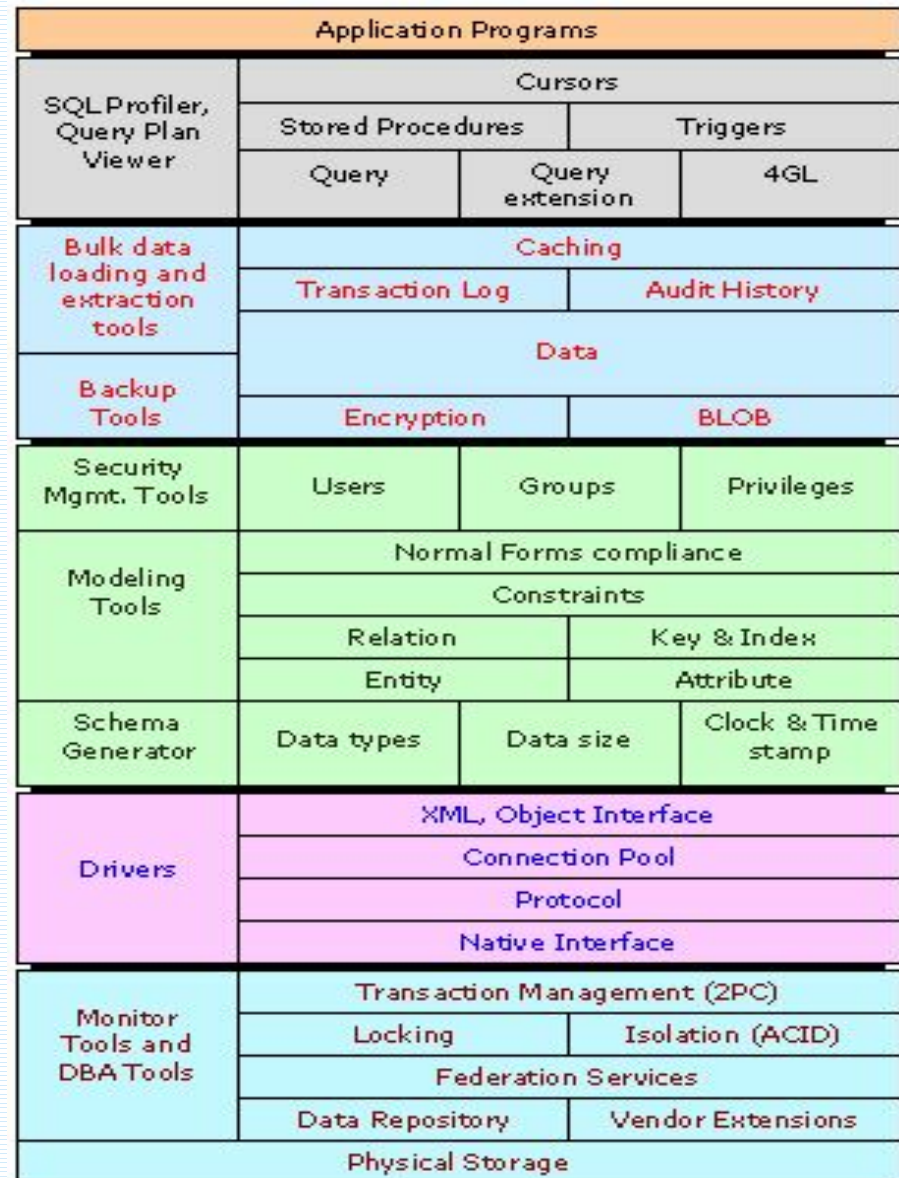
Persistence layer in applications have following constituents

- Storage Access Methods
- Data and History
- Schema
- Connectivity Options
- Data Server Features

Benefits

Data layer model captures finer level elements of persistence

Ensures migration methods are identified for all the elements



Platform Model

Concept

Application platform is a cohesive and collaborative set of services

Benefits

Captures overall environment – from hardware details to user interface

Gathers details from the perspective of business users, administrators, development community, vendors.

Captures dependency information and co-existence of applications in the platform

User Interface (Terminal, Desktop, Browser, Mobile)		
Developer Tools	Custom Application	Packaged Application
	Programming Language	
	Transaction Monitor	3 rd Party Software
	Language Runtime	Scripting Environment
	Messaging, I-O Mechanism (File IO, SQL)	
	Data Repository Architecture	
System Monitoring & Admin Tools	API	
	Device Driver	Network Protocol
	Operating System	

Migration Map

Fine grain road map with activities and methods for executing migration project

Concept

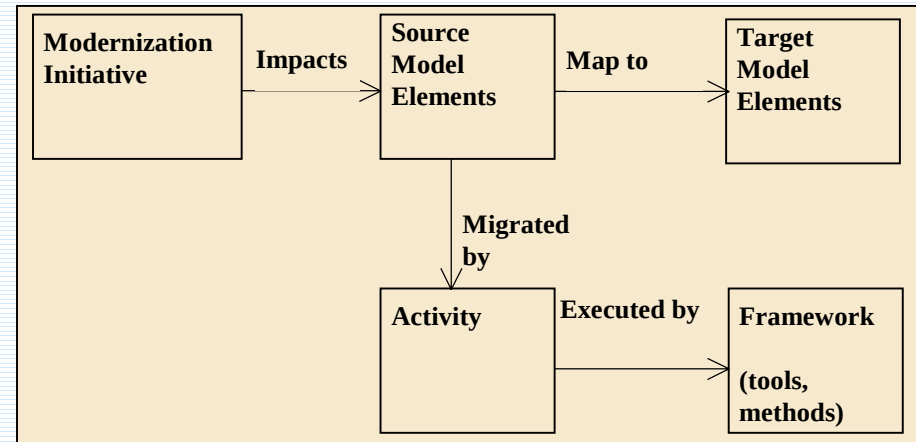
Migration transforms source elements into target elements through activities of migration framework

Benefits

Facilitates upfront blue print for migration

Ensures every element is considered in migration initiative

Guides in selecting tools for migration activities



Migration Mapping

E-to-E	Activity	Method	Tool

Migration Map Format

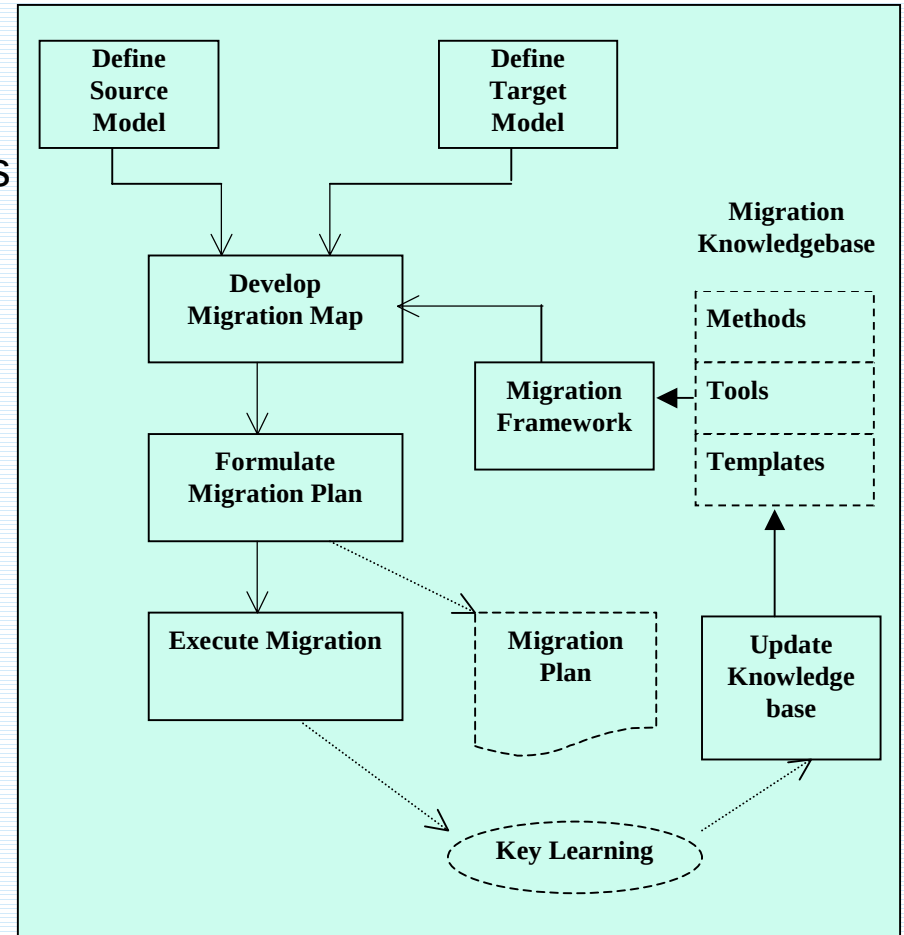
Model Driven Migration Approach

Concept

Equate legacy and target environment on model elements and define methods of migration

It is a process with following steps

Build Source & Target Models
Build Transformation Map
Choose methods for migration
Estimate and Execute
Capture Learning



Benefits of Migration Models

Migration coverage beyond source code and data is the core value of these models

Provides the big picture and defines scope of migration

Migration map is an excellent tool to blue print the transformation

Ensures that migration is planned for every source element

Highlight training needs for effective operations at target environment

Knowledge base captures migration expertise for future migrations

Models serve as valuable documents about enterprise platforms

Estimation and execution effectiveness are assured

Future Work on Migration Models

Current models are primitive. We acknowledge the need for more work in this direction

Define meta-models of these models using UML

Detailed models for elements such as Business Rules

Models to represent knowledge assets of legacy applications

Automatic profile analysis and Migration map generation methods

Models for migration knowledge base

Integration and interoperability with internal models of leading tools

A migration workbench that leverages these models

Thank you