CBDIJournal

Practice Guide

SAE2 Application Modernization Process

In this report we examine how the SAE process can evolve to support Application Modernization, by introducing new disciplines to cover the planning and architectural aspects of Application Modernization, the discovery of knowledge encapsulated in the applications, and finally to perform the reengineering that may result from new business requirements

By Lawrence Wilkes

Originally published in the December 2009 edition of the CBDI Journal

CBDI

Independent Guidance **for** Service Architecture and Engineering

SAE2 Application Modernization Process

In this report we examine how the SAE process can evolve to support Application Modernization, by introducing new disciplines to cover the planning and architectural aspects of Application Modernization, the discovery of knowledge encapsulated in the applications, and finally to perform the reengineering that may result in new business requirements.

By Lawrence Wilkes

Introduction and Context

Application Modernization (AM) is centered on three main capabilities as shown in Figure 1.

1. Producing As-Is models to understand the current system(s)
2. Producing To-Be models that articulate new business requirements and drive the new implementation
3. Managing the transition from the As-Is to the To-Be, which will include:
   - mapping the one to the other to understand what assets might be reused, what gaps exist, and how they might co-exist during the transition
   - transforming some of the As-Is assets so that they can participate in the To-Be

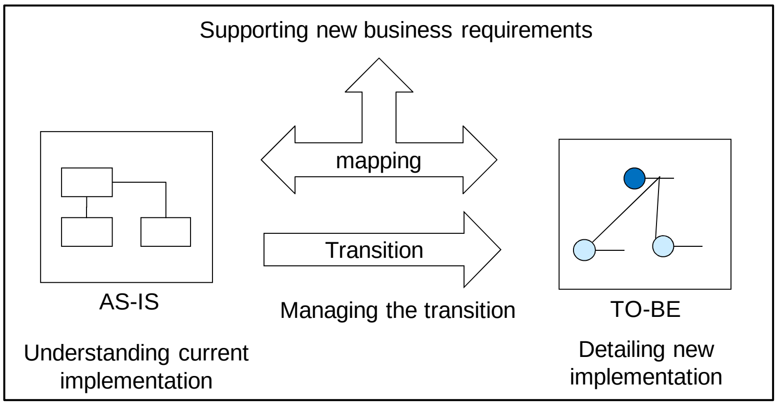
Supporting new business requirements
mapping
Transition
AS-IS
Managing the transition
TO-BE
Understanding current implementation
Detailing new implementation

Figure 1 - The Core of Application Modernization (AM)

These capabilities are described in widely used architecture frameworks shown in Figure 2 and detailed in Table 1. These provide context for different AM scenarios and desired outcomes of AM.

Some AM activities will be driven by an Enterprise Architecture initiative that is considering the modernization of a broad portfolio. This may not be as broad as the whole enterprise of course, but an EA style approach is recommended whenever a portfolio of systems requires modernization.

To provide the necessary agility and avoid the creation of yet more legacy systems, we would expect a Service Architecture adhering to SOA principles to be one of the key elements of the To-Be model.

This is not always the case; frequently legacy transformation activities may just be focused on platform migration, rather than establishing improved architecture and supporting new business requirements. Similarly, many tactical, project-based, application-led scenarios may not include SOA. Though they may deliver some service enabled APIs in the current and or transformed system, there is often insufficient compliance with full SOA principles.

Framework	Context	Example
Enterprise' Architecture (EA)	Provides the overall context for both SOA and Legacy Transformation. Typically, • Broad scope, high level • Portfolio context • Business context • High level Inventory • Models As-Is and To-Be at architecture level	TOGAF, DoDAF, FEA,
Service Oriented Architecture (SOA)	Provides the To-Be for Legacy Transformation. Typically • Appropriate scope & detail • models to-be at architecture level	CBDI-SAE
Legacy Transformation	Typically, • Narrow scope, detailed • System context • Models As-Is and To-Be at implementation level • Provides detailed inventory	SEI Horseshoe Model[1], OMG ADM/KDM[2]

Table 1: Contextual Framework Relationships

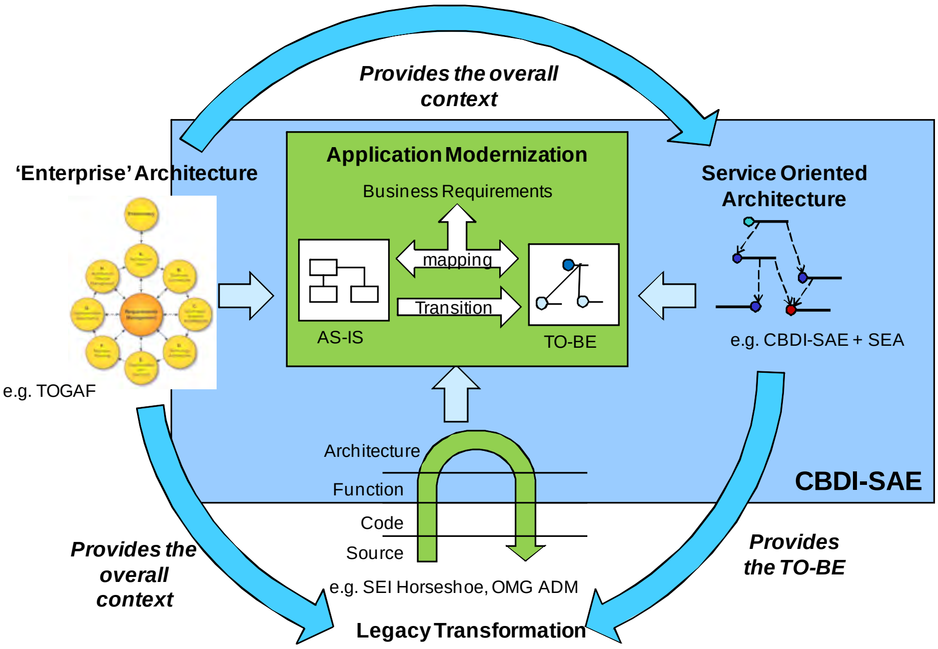
Provides the overall context
'Enterprise' Architecture
e.g. TOGAF
Application Modernization
Business Requirements
mapping
Transition
AS-IS
TO-BE
Service Oriented Architecture
e.g. CBDI-SAE + SEA
CBDI-SAE
Architecture
Function
Code
Source
e.g. SEI Horseshoe, OMG ADM
Provides the overall context
Provides the TO-BE
Legacy Transformation

Figure 2 - Contextual Framework Relationships

SAE Process

As explained in a related CBDI Practice Guide[3], we continue to expand the scope of CBDI-SAE to address these relationships – illustrated by the blue CBDI-SAE box in figure 2. For example we have provided mapping into TOGAF, and now into AM.

To this end we have enhanced the SAE Process as illustrated in Figure 3 to detail new disciplines that are focused on AM requirements, and we look at these in more detail in this report

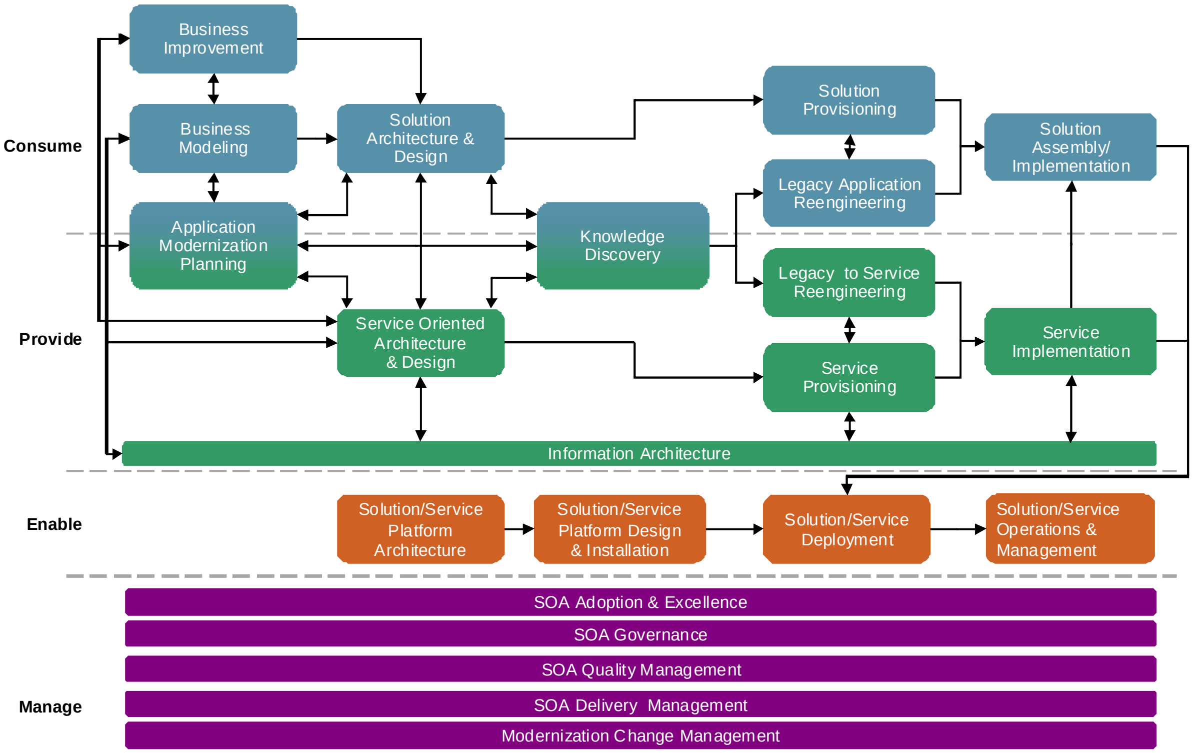
Consume
Provide
Enable
Manage
Business Improvement
Business Modeling
Application Modernization Planning
Solution Architecture & Design
Knowledge Discovery
Service Oriented Architecture & Design
Solution Provisioning
Legacy Application Reengineering
Legacy to Service Reengineering
Service Provisioning
Solution Assembly/ Implementation
Service Implementation
Information Architecture
Solution/Service Platform Architecture
Solution/Service Platform Design & Installation
Solution/Service Deployment
Solution/Service Operations & Management
SOA Adoption & Excellence
SOA Governance
SOA Quality Management
SOA Delivery Management
Modernization Change Management

Figure 3 - SAE2 Process

SAE2 Process Disciplines – Planning and Architecture

Let’s start by looking at the planning and architecture level activities to support AM as illustrated in Figure 4. We have introduced three new disciplines here - with Architecture Modernization Planning (AMP) and Knowledge Discovery (KD) replacing the existing Legacy to Service Transition Planning.

Application Modernization Planning (AMP)

- Defines outline architecture for AM
- Delivers increment to Application Portfolio Plan
- Goals are to understand the As-Is architecture, identify candidate components for reengineering or reuse in the To-Be architecture, and to plan transition
- Provides architecture level mapping of candidates to the To-Be Solution and Service Architectures
- Deliver just enough portfolio perspective to ensure goals and objectives are meat, but without undue delay.

Knowledge Discovery (KD)

- Defines detailed architecture and design for AM and transformation
- Goals are to understand the As-Is system, and extract knowledge of the current assets
    - at the Architecture level: This maps to the Knowledge Discovery Metamodel (KDM) Abstractions Layer – providing Current Systems Models
    - at the detailed level: This maps to KDM Resource, Program Elements and Infrastructure packages, capturing the business rules, code structure, etc, in the current asset(s)
- Provides detailed level mapping of As-Is resources and elements to the components and services in the To-Be Solution and Service Architectures

Information Architecture (IA)

As we broaden the scope of SAE2 we recognize the need to have a separate discipline for Information Architecture and to break out the process from Service Oriented Architecture & Design (SOAD). This provides greater clarity in how IA deliverables are used by AMP and SOAD activities, and at the same time refined by the outputs from AMP and SOAD.

IA will be responsible for deliverables that are used in other disciplines:

- Service Information Models form part of the Service Specification produced in Service Provisioning
- Message schemas are used both in the Service Specification and at run-time
- Data Exchange Mappings that provide the translation of information between current systems and new Services

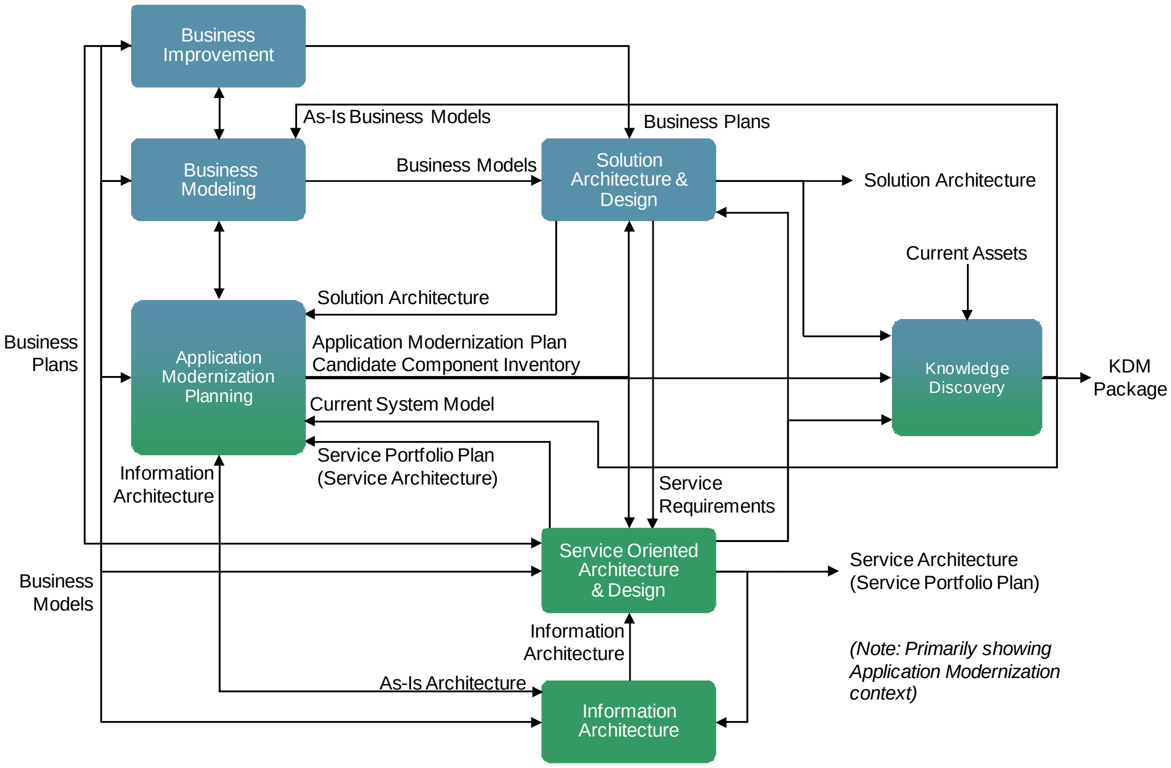

Figure 4 – SAE Process Disciplines – Planning and Architecture

The comments on IA indicate there will often be a process of refinement and iteration of deliverables exchanged between disciplines.

Between AMP and SOAD there will be a need to:

- Refine plans
  - Adjust initial plans with input from agreed architectures
- Refine architectures.
  - Adjust architecture with input from plans
    - Accept/reject candidates
- Iterate through views
  - Service Architecture
    - Accept/reject candidate Services
  - Service Implementation Architecture
    - Accept/reject candidate Automation Units

There will be similar refinements between the AMP and Solution Architecture & Design (SA&D).

Application Modernization Planning (AMP) Process Units and Tasks

The Process Units and Tasks for AMP are explored in table 2 along with their main deliverables.

Process Units (PU) and Tasks	Definition	Deliverables
PU: Prepare Current System Portfolio Assessment	Determine the current state of existing systems and their contribution to the business. [4]	Current System Portfolio Assessment
PU: Prepare and evolve Application Modernization Plan	Prepare and continue to evolve the Application Modernization Plan in response to business improvement requirements	Application Modernization Plan Comprised of:
Task: Determine Scope and Impact	Identify the scope of modernization requirements and impact of business plan on current solutions within scope	- Impact Analysis
Task: Determine Gap	Determine the gap between current solutions and new business and IT requirements	- Gap Analysis
Task: Determine Application Modernization Approach	Determine the suitable approach (per unit of scope) and assess what capabilities are required to perform approach	- Application Modernization approach - Capability requirements
Task: Prepare Reengineering requirements	Specify requirements for reengineering current asset	- Reengineering requirement
Task: Determine Solution Transition	Determine transition steps from As-Is to To-Be solution	- Solution Transition Plan

Process Units (PU) and Tasks	Definition	Deliverables
Task: Produce Business Case	Justification/ROI for Application Modernization	- Application Modernization Business Case
PU: Identify Candidate Components for Modernization	Produce candidate list of current assets that can participate in Solution and Service Architecture – either reengineered, or As-Is	Candidate Component Inventory Comprised of:
Task: Identify Candidate Components	Analyze current system model (As-Is) and identify and map candidate components to new Solution and Service Architecture (To-Be) requirements	- Candidate Solution Components - Candidate Underlying Services - Candidate Automation Units - Architecture level mapping of candidates to Solution and Service Architectures

Table 2 - Application Modernization Planning Process Units and Tasks

Knowledge Discovery (KD) Process Units and Tasks

The Process Units and Tasks for KD are explored in Table 3 along with their main deliverables.

Process Units (PU) and Tasks	Definition	Deliverables
PU: Produce Current System model	Produce architecture of current solution/system, identifying current Components, APIs, Services, etc *Note: This may be performed outside of this discipline, where perhaps suitable models already exist and do not have to be 'discovered' from current systems*	Current System Model Consisting of:
Task: Current Asset Architecture Analysis	Analyze current assets at architectural level to identify Abstraction packages (as per KDM)	- Abstraction Packages
PU: Analyze Existing Systems	Analyze and extract the knowledge encapsulated in current assets	KDM Packages Consisting of:

Process Units (PU) and Tasks	Definition	Deliverables
Task: Current Asset Detail Analysis	Analyze current assets at detail level to identify Resource, Program Elements and Infrastructure packages (as per KDM) Including As-Is Business Model and Business Rules (to the extent encapsulated in current asset)	- Resource Packages - Program Element Packages - Infrastructure Packages - Candidate Operations - As-Is Business Model
Task: Map to Requirement	Map extracted knowledge to Solution Component Specification Or Map extracted knowledge to Service Specification and/or Map extracted knowledge to Automation Unit Specification	- Mapping packages to requirements and specifications - Map packages to Implementation Architecture

Table 3 – Knowledge Discovery Process Units and Tasks

SAE Process – Provisioning, Implementation & Assembly

In the Provisioning, Implementation & Assembly area we have introduced the Legacy Application Reengineering discipline and the parallel Legacy to Service Reengineering discipline to support AM as illustrated in Figure 5.

The focus of these disciplines is to perform the transformation or reengineering of the current assets to meet new Solution Component or Service Implementation requirements.

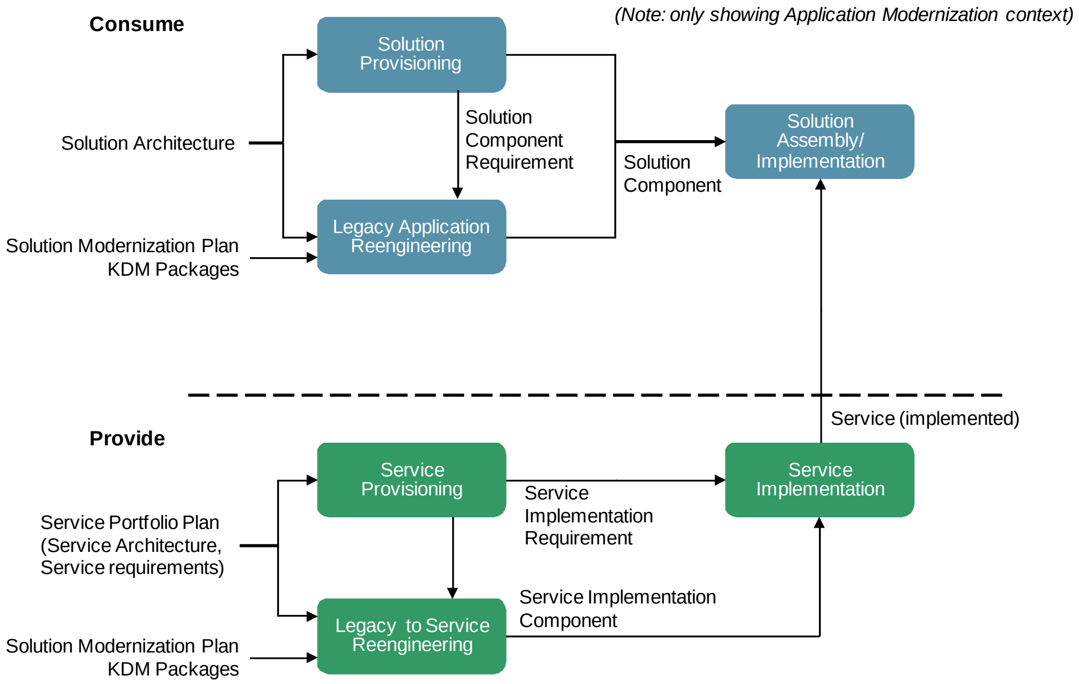
Consume
(Note: only showing Application Modernization context)
Solution Provisioning
Solution Architecture
Solution Component Requirement
Solution Component
Solution Assembly/ Implementation
Legacy Application Reengineering
Solution Modernization Plan
KDM Packages
Provide
Service (implemented)
Service Provisioning
Service Implementation Requirement
Service Implementation
Service Portfolio Plan (Service Architecture, Service requirements)
Service Implementation Component
Legacy to Service Reengineering
Solution Modernization Plan
KDM Packages

Figure 5 - SAE Process – Provisioning, Implementation & Assembly

Reengineering Process Units and Tasks

The Process Units and Tasks for legacy to Service Reengineering are explored in Table 4 along with their main deliverables. Similar Process Units and Tasks will support Legacy Application Reengineering.

The actual reengineering tasks and techniques will be package type and platform specific, and so we are not detailing them at this stage.

Process Units (PU) and Tasks	Definition	Deliverables
PU: Prepare and evolve Legacy Transition Plan	Prepare and evolve the Legacy Transition Plan in response to Service delivery and Application Modernization requirements	Legacy Transition Plan Comprised of:
Task: Service Reengineering Analysis	Identify suitable Service reengineering approaches for use on current assets, and map discovered knowledge to the To-Be requirements	- Recommended Service reengineering approach - Knowledge mapping
Task: Prepare Service Reengineering requirements	Specify requirements for reengineering current assets	- Service reengineering requirement - (including Service Specification and/or Automation Unit Specification)
PU: Reengineer Service Implementation Component	Produce Service Implementation Component by reengineering current application asset	Service Implementation Component

Table 4 – Legacy to Service Reengineering Process Units and Tasks

Entry and Exit Points

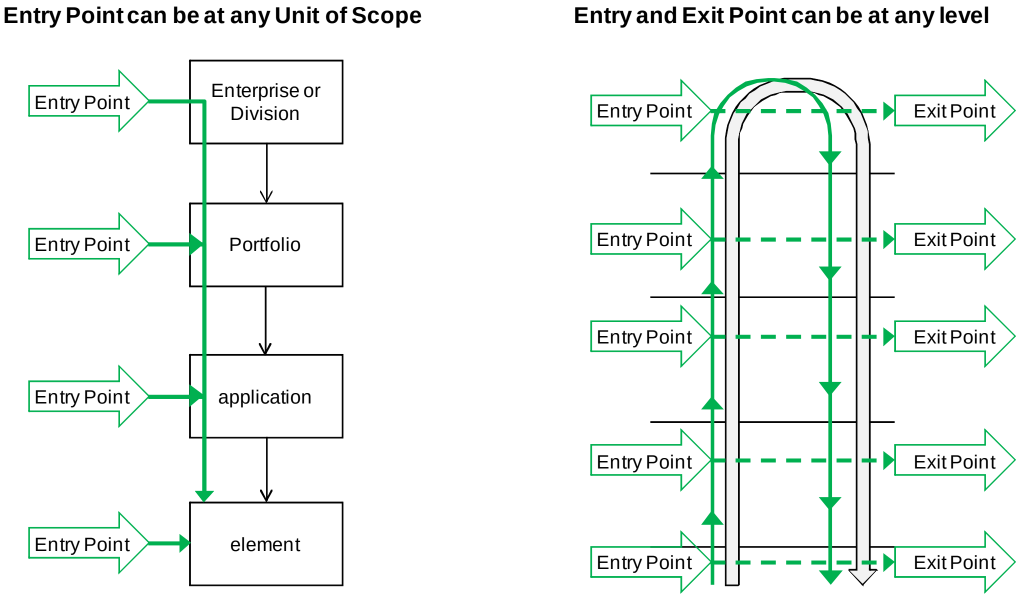

Figure 6 - Entry and Exit Point

It is important to recognize that there are multiple entry and exit points in the Application Modernization process. It isn't always portfolio-based, nor is it always strategic, nor should it be considered 'waterfall' in approach.

Figure 6 illustrates in terms of scope that the entry may be at any appropriate level, and subsequent progress is typically down through the decomposition of units of scope. The broader the unit of scope, the more important it becomes to recover the architecture first in order to understand the underlying decomposition.

In the horseshoe progress is either:

- around the horseshoe in terms of architecture recovery, and subsequent architecture-based development
- or directly across in terms of modernizing an individual element.

The exit point isn't always at the bottom of the horseshoe as the modernization approach may have been determined to be replacement by buying, or outsourcing the implementation. In which case the architecture and specification levels of detail may be sufficient.

Next Steps

In this report we have set out the SAE2 Application Modernization Process. The SAE2 process is distinctive in that it is strongly architecture driven with considered choice of modernization approach.

In related CBDI Guidance[5] we examine in detail the task of determining which Application Modernization approach or approaches should be applied in order to meet specific modernization and business requirements.

In subsequent reports we will continue detailing various process units, tasks and relevant techniques, including understanding how to bridge between the As-Is and the To-Be in terms of patterns, and knowledge mapping. We will also detail how the business case can be developed.

[1] Options Analysis for Reengineering (OAR): Issues and Conceptual Approach
http://www.sei.cmu.edu/library/abstracts/reports/99tn014.cfm

[2] OMG Architecture-Driven Modernization Task Force

http://adm.omg.org/

[3] SAE2 - Framework for Application Modernization, CBDI Journal December 2009

[4] See Legacy Portfolio Assessment. CBDI Journal, Sept 2006

[5] Application Modernization Approaches, CBDI Journal December 2009

About CBDI

CBDI Forum is the Everware-CBDI research capability and portal providing independent guidance on best practice in service oriented architecture and application modernization.

Working with F5000 enterprises and governments the CBDI Research Team is progressively developing structured methodology and reference architecture for all aspects of SOA.

CBDI Products

The CBDI Journal is freely available to registered members. Published quarterly, it provides in-depth treatment of key practice issues for all roles and disciplines involved in planning, architecting, managing and delivering business solutions.

Visit www.cbdiforum.com to register.

Platinum subscription – A CBDI Forum subscription provides an enterprise or individual with access to the CBDI-SAE Knowledgebase for SOA delivering ongoing practice research, guidance materials, eLearning, tools, templates and other resources.

Everware-CBDI Services

At Everware-CBDI we enable large enterprises and governments to become more agile by modernizing their business systems. We have repeatable processes, resources, tools and knowledge-based products that enable enterprises to transform their current applications in an efficient, low risk manner, into an optimized service-based solutions portfolio that supports continuous, rapid and low cost evolution. Our consulting services range from providing practices and independent governance to architecture development, solution delivery and service engineering.

Contact

To find out more, and to discuss your requirements visit www.everware-cbdi.com or call

USA and Americas: 703-246-0000 or 888-383-7927 (USA)

Europe, Middle East, Africa, Asia, and Australasia: Telephone: +353 (0)28 38073 (Ireland)

www.everware-cbdi.com

IMPORTANT NOTICE: The information available in CBDI publications and services, irrespective of delivery channel or media is given in good faith and is believed to be reliable. Everware-CBDI Inc. expressly excludes any representation or warranty (express or implied) about the suitability of materials for any particular purpose and excludes to the fullest extent possible any liability in contract, tort or howsoever for implementation of, or reliance upon, the information provided. All trademarks and copyrights are recognized and acknowledged. The CBDI Journal may be distributed internally within customer enterprises that have current corporate subscriptions. Otherwise CBDI Journals may not be copied or distributed without written permission from Everware-CBDI.