Governance for Modeling and Implementing Enterprise Services at SAP

Enterprise SOA Solution Management, SAP AG

Walldorf, April 2007
<table>
<thead>
<tr>
<th>Introduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harmonized Enterprise Model</td>
</tr>
<tr>
<td>PIC Process and Enterprise Services Community Process</td>
</tr>
<tr>
<td>Implementation and Life Cycle</td>
</tr>
<tr>
<td>Summary</td>
</tr>
</tbody>
</table>
Business Process Platform

Exposed as enterprise services

Portal  Rendering  Devices  Office  RFID

SAP xApps
composite applications powered by SAP NetWeaver

Business process platform

Enterprise Services Repository

Home-grown or ISV  SAP  Process components  Subsidiary  Bus. partner

mySAP ERP  mySAP CRM  mySAP SRM  Business object

© SAP AG 2007, Governance for Modeling and Implementing Enterprise Services / 3
Enterprise Services in Business Process Platform

**SAP xApps**
composite applications powered by SAP NetWeaver

**Business process platform**

**Enterprise Services Repository**

**Home-grown or ISV**

**SAP**

**Process components**

**Sub-sidiary**

**Bus. partner**

**Structure of enterprise services**

**Service interface**

**Glue code**

**Service implementation**
An enterprise service is a callable entity that provides business functionality and is published by SAP in the Enterprise Services Repository. Enterprise services are structured according to a harmonized enterprise model based on global data types (GDTs), process components, and business objects. They are well documented, safeguard quality and stability, and are based on open standards.
What Would Consumers of Enterprise Services Expect?

Enterprise services should be

- Easy to discover (have a harmonized classification scheme)
- Easy to understand (business language)
- Easy to invoke (based on open standards)
- Have a stable interface and stable behavior (well-defined life cycle)
Common guidelines and patterns for modeling and implementation of enterprise services make life easier for service consumers.

SAP has already put many product and process standards into place that also apply to enterprise services.

For enterprise services, additional guidelines and patterns exist on different levels:

- Map of process component and business objects (BOs)
- Service interfaces and service operations per BO
- Structure of message types (= signature of service operations)
- Common set of reusable data types
- Transactional behavior at runtime
- Service implementation (business application code)
Introduction

Harmonized Enterprise Model

PIC Process and Enterprise Services Community Process

Implementation and Life Cycle

Summary
Harmonized Enterprise Model

Harmonized enterprise model

Achieved by modeling all enterprise services according to rules for
- Service definition and service cut
- Service signature
- Element definition stored in a common dictionary
  (→ global data types)

The process integration content (PIC) governance process takes care for compliance with the rules
- PIC 0 for service definition and cut
- PIC 1.3 for service signature
- GDT PIC for new elements

- All changes of already shipped enterprise services are reviewed once again to safeguard compliance and interface stability

ARIS is used for modeling of the service definition and service signature
Service Definition

Based on

Business objects
- A business object represents a specific view on well-defined and outlined business content. Business objects are defined free of business functionality redundancies.

Process components
- Process components group business objects. A business object belongs to exactly one process component. Process components describe a part of the value chain. That part is typically executed by one department (in large companies).

Deployment units
- A deployment unit is a group of decoupled process components that can be operated separately.

Service operations
- An operation belongs to exactly one business object. A business object has multiple operations.

Service interfaces
- A service interface is a grouping of operations.
Service Cut Based on Interface Patterns

Service cut based on interface patterns to safeguard efficiency and harmonization, based on standardized, cross-solution approach.

Patterns incorporate transactional communication patterns.

Define rules for grouping operations into interfaces.

Example: Manage <BO> Interface

Define rules for operation granularity.

Define proposals and rules for naming and definition.
Transactional Communication Patterns

Sender

Query
Response

Request
Confirmation

Notification

Information

Receiver

Asynchronous or synchronous

Database update

Synchronous only

© SAP AG 2007, Governance for Modeling and Implementing Enterprise Services / 12
Model Overview: Business Object Map

Deployment unit

Process component (drill down in detailed model)

Business object
Model Drilldown: Process Component → Enterprise Service
Modeling of business objects is used to derive service signatures to safeguard

- Harmonization of message structures across different service operations for the same business object
  → All messages for one business object are based on the same structure, which is the structure of the business object

Each service operation utilizes messages for communication
**Global Data Types: The Basic Elements of Service Signatures**

- **Global data type** = reusable elements stored in a common dictionary for enterprise services
  - **Message data type** = GDT + log
    - Suffix: message
    - Reuse by projection
    - Part of SAP global (data type)
  - **Message type** = signature for service operation
    - Part of SAP Global (message type)

- **Service operation**
SAP established the PIC governance process for the creation and definition of enterprise services.

The entire enterprise service definition process consists of:
- PIC 0: service definition and cut
- PIC 1: raw design of interfaces based on business object model
- GDT – PIC: harmonized definition of relevant data types
- PIC 3: final interface definition including element structure with all details
- Service implementation in ES Repository and back end
Outside-In: Enterprise Services Community Process

Collaborative Process

Roadmap-Driven Track
SAP’s plans to service-enable mySAP Business Suite

Stakeholder-Driven Track
Customers / partners deciding how SOA drives business value

DECIDE

Confirmation of SAP plans
Prioritize, considering dependencies

Long List
use cases

Select
quarterly

Short List

Necessary but not obvious

Committed Outcome

DEFINE collaboratively
SAP Architects

DEVELOP

DEPLOY on SAP ERP 2005

DECLARE

COMMUNITY Definition Groups

ES Bundles

THE BEST-RUN BUSINESSES RUN SAP™
Introduction
Harmonized Enterprise Model
PIC Process and Enterprise Services Community Process
Implementation and Life Cycle
Summary
Guidelines for Service Implementation

Enterprise services should technically behave in the same manner concerning the following features

- **Transactional behavior**
  - Enterprise services are **stateless**: each service call is an atomic transaction
  - COMMIT logic: all services perform database updates in local update task
  - Reliable and exactly once execution of service calls safeguarded

- **Error-handling**
  - Errors are always returned in the same structure using the GDT log as a mandatory element of response and confirmation messages
Enterprise services are built for a high level of reuse inside and outside of SAP software

→ Keeping released services stable is key

The level of stability needs to be known and common for all enterprise services

However, the following interests need to be balanced

- **Protect** our customers’ and partners’ investments in integration scenarios and composite applications that leverage enterprise services

→ Safeguarding Stability

Released service operations can be changed only in a compatible manner.
Incompatible changes require that new service operations are introduced, and the old ones are deprecated.
Which Changes Are Compatible – Two Rules of Thumb

A change to a service operation is (backward) compatible if any application consuming the unchanged operation can consume the changed operation and retain its own functionality without having to be adjusted in any way.

A change to a service operation is (backward) compatible if under the same conditions, the original and changed operations return the same output when receiving the same input, with the changed operation’s output projected to the original operation’s output structure.
A change must be at least *syntactically compatible* to be compatible

→ SAP-accepted rule

A service operation change is syntactically compatible if **only optional elements or attributes are added** to an element of complex type in input or output message type.

All other changes are considered syntactically incompatible.
Service Operation Deprecation and Maintenance

Service op. Hugo

Service operation Hugo

Release N-1

Release N

Release N+1

Release N+2

Compatible change

Service op. Hugo

Service operation Hugo_V1

Released

Deprecated: maintenance

Revoked: throws exception

Removed: out of maintenance

Release N-1

Release N

Release N+1

Release N+2

Incompatible change

SAP AG 2007, Governance for Modeling and Implementing Enterprise Services / 25
Governance is crucial for the success of enterprise SOA to provide a consistent set of enterprise services following common guidelines and patterns.

How do we take care that SAP enterprise services adhere to common guidelines?

- **Modeling and implementation guidelines** are mandatory for every (new) service development.
- **Special review process** has been installed with several quality gates (reviews by PIC council).

And in addition

- **Manual and automated tests** are executed regularly for every service operation to safeguard functional correctness.