Business Consulting Services

Web Services - A Consultant‘s View
From IT Stategy to IT Architecture

Hans-Peter Hoidn, Timothy Jones,
Jürg Baumann, Oliver Vogel

February 12, 2003

Agenda

Introduction

I. Motivation
II. Real World Situation
III. From Business to Architecture
IV. What an Architecture has to provide
V. From Architecture to running services
VI. Summary: Outstanding Issues

Conclusion

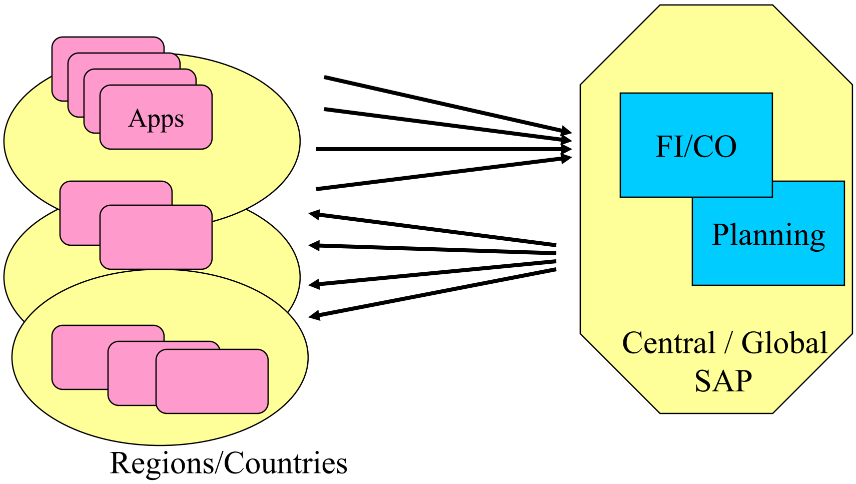
I. Motivation

- Consultant‘s View:
  - Business processes span multiple departments and / or companies and each party has its own IT system
  - Coexistence and collaboration of existing applications and packaged systems (like ERP, CRM, SCM)
  - Integration plays an important role
- Promise Web Services
  - Bridging gaps among heterogeneous platforms
  - Faster Development
  - Easier collaboration of organizations
  - Enabling SOA (Service Oriented Architecture)
- *Let‘s perform a reality check*
  - Actual business needs
  - Capabilities of web services

II. Real World Situation

- IBM carries out large projects with global customers:
  - New business processes must be implemented, e.g. for global operations, supply chain management
  - New governance procedures over heterogeneous business environments must be supported
  - Solutions crossing boundaries of divisions, companies must be provided
- Facets of integration solutions are:
  - Global or regional solutions
  - Regional / country / plant specific software components
  - Autonomous business units
  - Heterogeneous infrastructure (hardware and software)

Business and Application Landscape

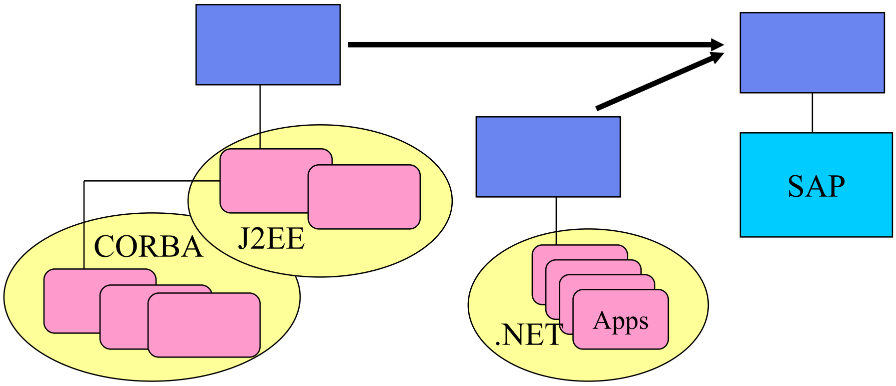


Examples of Application Landscapes

- Examples – driven by globalization:
  - Operational systems in each country feed global FI/CO system, which provides controlling on the holding level
  - Production planning on the corporate layer drives production in local plants
  - Consolidation of planning across various legal entities
  - Central procurement system
- New IT Strategies and their implications :
  - Implementing new business processes for reporting
  - Replacement of local FI/CO systems by a global solution
  - Regional IT support instead of IT support per country
- *The* information system is a system of systems

Mix of platforms – a closer look

Business Processes spanning heterogeneous environments



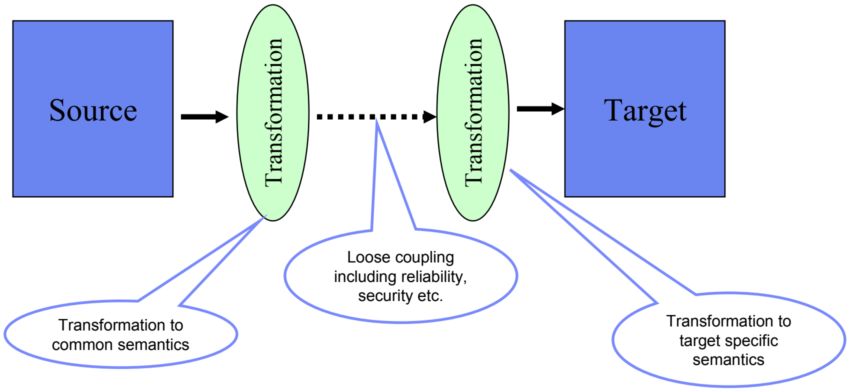
IT Environment - Consequences

- Mix of technologies and paradigms to enable information flow (e.g. from and to ERP systems) among the „Extended Enterprise“
  - Close coupling (synchronic)
  - Loose coupling (Messaging and Web Services)
- Mix of technologies
  - J2EE
  - CORBA
  - .NET
  - WSDL
- Consequences:
  - The view of the business processes drive the use of technology
  - There is the need for a holistic approach

III. From Business to Architecture

- Key information items describing business processes:
  - Business function / service (not necessarily IT supported)
  - Business system (grouping of business functions)
  - Information flow (contains information objects)
  - Representation of business objects differ (e.g. identifiers are different in various systems).
- What do we need is a holistic view
  - of the business and its processes
  - over heterogeneous groups addressing multi-language, multi-culture, multi-company aspects
  - over heterogeneous platforms addressing multi-programming-language, multi-software-platform, multi-hardware aspects

Applications / Components and Information Flow



Transformation to common semantics

Loose coupling including reliability, security etc.

Transformation to target specific semantics

Requirements

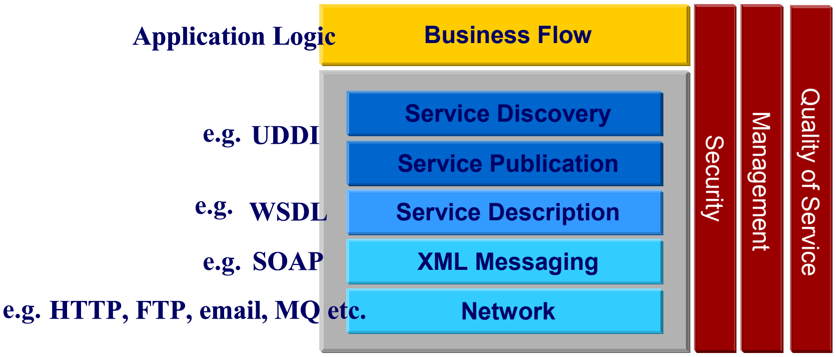
- Holistic view of the business including common semantics of business terms
- An architecture that is independent of company boundaries and managing heterogeneous technology platforms (where platforms themselves will change)
- Coexistence of Components and Web Services, unified view of various kinds of services
- Concepts of transformations embedded in information flow and workflow concepts
- Information technology management requires less dependence on infrastructure

IV. An Architecture has to Provide Unified Views of

- Rules and Procedures
  - to capture business processes with the needed precision
- Semantics and ontologies
  - use the same terminology
- Service definitions
  - definitions of components and interfaces
  - access methods
- Separation of concerns
  - Tiers (logical distribution of functionality)
  - Layers (separation of logic - addressing different abstraction layers)

An Architecture has to cover various abstraction layers

A service architecture may consist of the following layers:



Application Logic

Business Flow

e.g. UDDI

Service Discovery

Service Publication

e.g. WSDL

Service Description

e.g. SOAP

XML Messaging

e.g. HTTP, FTP, email, MQ etc.

Network

Security

Management

Quality of Service

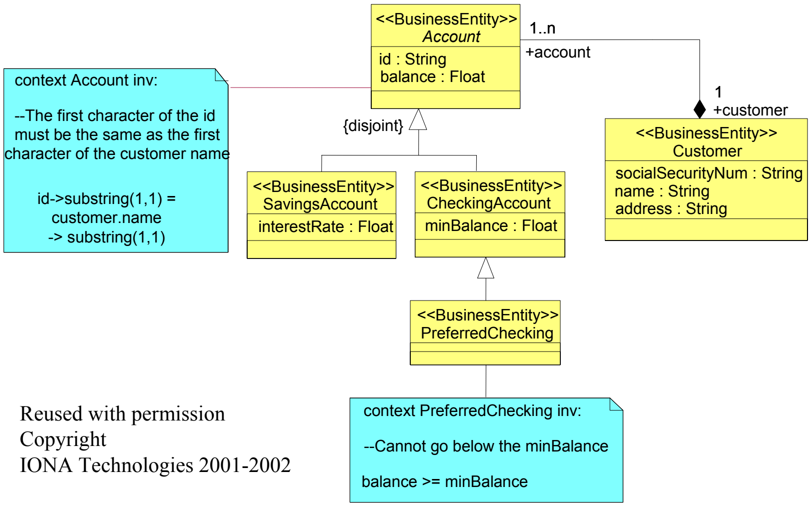
Elements of a Solution – Semantics and Ontologies

- *ebXML*: Standardization of business messages
  However, ebXML emphasizes the Web Services approach, but the semantics are valid for a larger scope
  - Holistic view
- Different view points of „*Services*“
  (SOA – Service Oriented Architecture)
  - „Business Services“ (e.g. enroll customer)
  - „Technical services“ (e.g. add_customer)
- *Ontology* is a classification of terms
  - Common vocabulary
  - Includes relationships between terms
  - Enabling expression of domain-specific knowledge
  - Is a pre-requisite for business-wide standardization

Elements of a Solution - Specifications

- UML (Unified Modeling Language)
  - Means to define rigorous specifications that are precise and sufficiently complete
  - Modeling of component view
- Metamodels and Profiles
  - *Business Process, Business Rules, and Information Flow*: Providing specifications for the execution and interoperation of business processes
  - *Components and Interfaces*: Providing specifications for the definition of collaboration (including composition and decomposition of components)

Example Precise Business Information Model



<<BusinessEntity>>
Account
id : String
balance : Float

1..n
+account

1
+customer

<<BusinessEntity>>
Customer
socialSecurityNum : String
name : String
address : String

{disjoint}

<<BusinessEntity>>
SavingsAccount
interestRate : Float

<<BusinessEntity>>
CheckingAccount
minBalance : Float

<<BusinessEntity>>
PreferredChecking

context Account inv:

--The first character of the id must be the same as the first character of the customer name

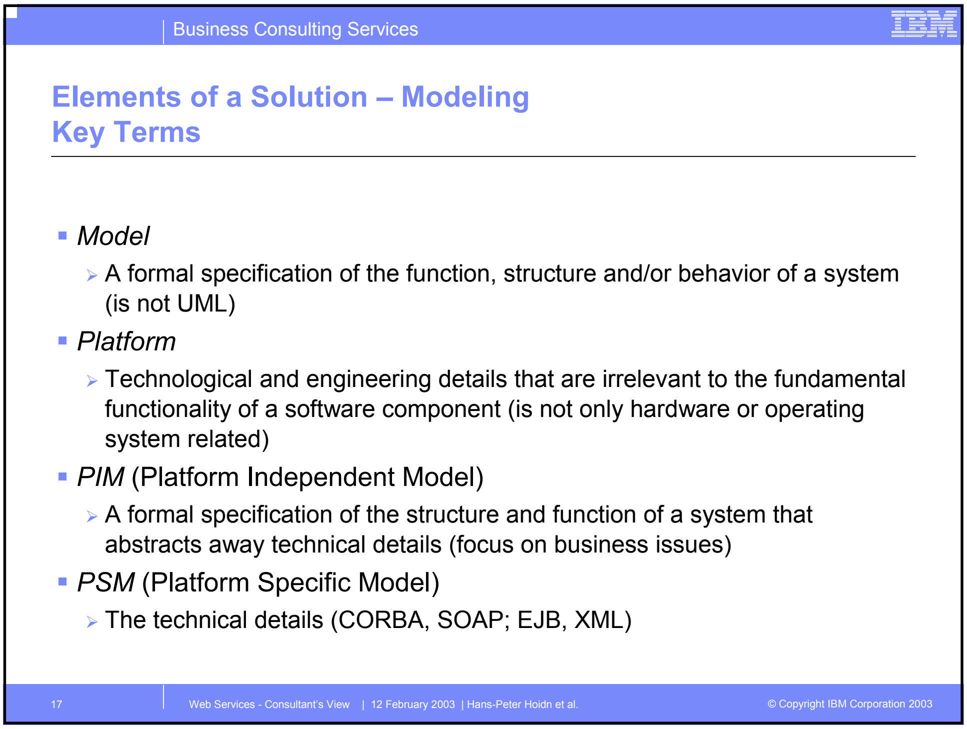
id->substring(1,1) =
customer.name
-> substring(1,1)

context PreferredChecking inv:

--Cannot go below the minBalance

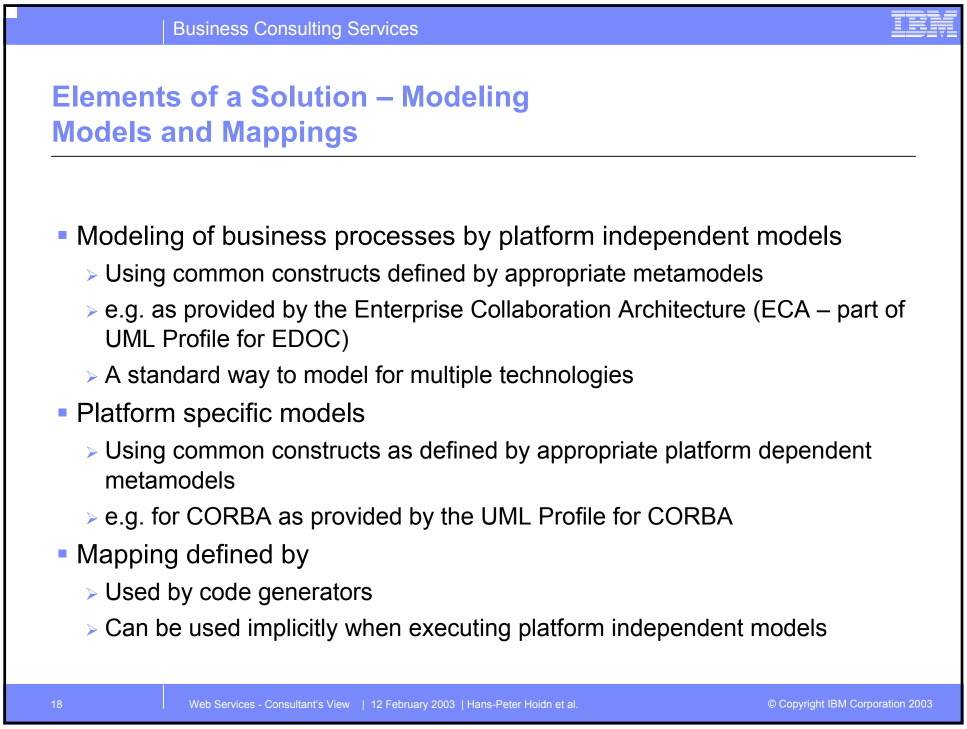
balance >= minBalance

Reused with permission
Copyright
IONA Technologies 2001-2002



Elements of a Solution – Modeling Key Terms

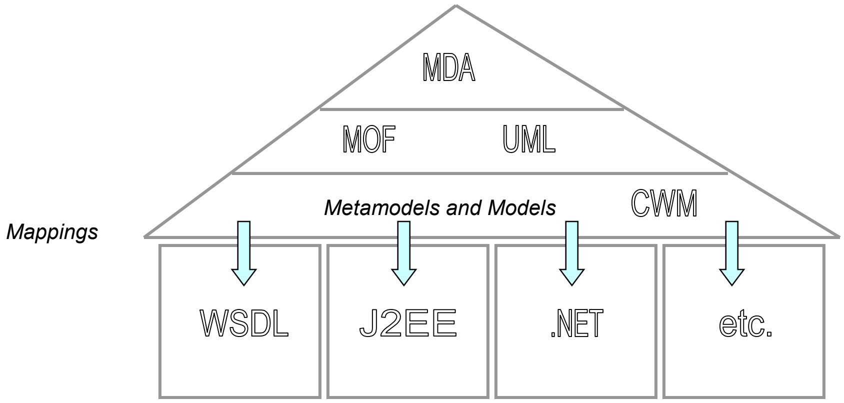
- *Model*
  - A formal specification of the function, structure and/or behavior of a system (is not UML)
- *Platform*
  - Technological and engineering details that are irrelevant to the fundamental functionality of a software component (is not only hardware or operating system related)
- *PIM* (Platform Independent Model)
  - A formal specification of the structure and function of a system that abstracts away technical details (focus on business issues)
- *PSM* (Platform Specific Model)
  - The technical details (CORBA, SOAP; EJB, XML)



Elements of a Solution – Modeling Models and Mappings

- Modeling of business processes by platform independent models
  - Using common constructs defined by appropriate metamodels
  - e.g. as provided by the Enterprise Collaboration Architecture (ECA – part of UML Profile for EDOC)
  - A standard way to model for multiple technologies
- Platform specific models
  - Using common constructs as defined by appropriate platform dependent metamodels
  - e.g. for CORBA as provided by the UML Profile for CORBA
- Mapping defined by
  - Used by code generators
  - Can be used implicitly when executing platform independent models

Levels of Abstraction


MDA

MOF UML

Metamodels and Models CWM

Mappings

WSDL J2EE .NET etc.

The MDA Promise – Bridging the Gaps

- *Horizontally* – collaboration and tiers:
  - Business process execution and interoperation (some more standards needed)
  - Flow of information
  - Data transformations
  - Transactions
- *Vertically* – separation of logic:
  - Mappings to various platforms – from PIM to PSM (including Web services)
  - Generation of definitions and code (including WSDL descriptions)
  - Execution of models

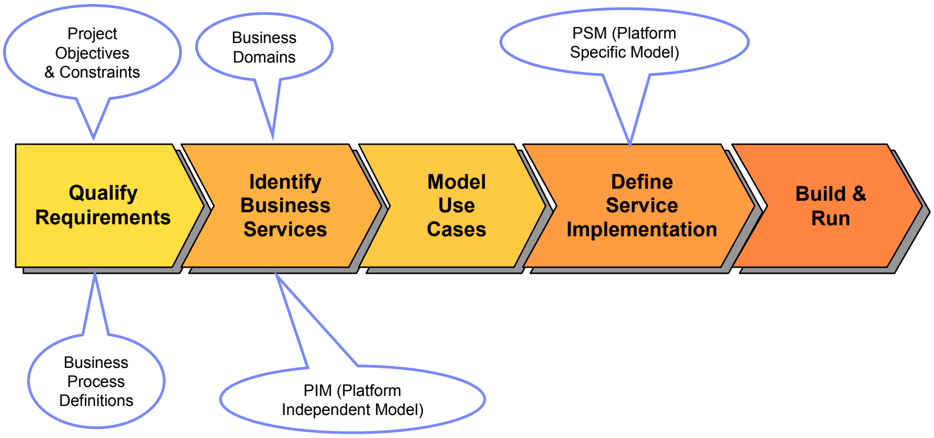
The MDA Promise – Holistic View

- Isolation of information and processing logic from technology specifics
  - Consistent view of the business processes and information flow
  - Enabling crisply layered and tiered architectures
  - Using a “normal form” of interaction (“collaboration”)
- Mappings of technology independent models to platform specifics
  - Common modeling rules
  - Common definition for code generation
- Help to integrate the mix of today, an architecture framework to support the unexpected
  - Integration of assets
  - Flexibility

Web Services in this Picture

- Web Services
  - Support an application-to-application integration that is loosely-coupled over the web
  - Support information flow among heterogeneous platforms
  - Are based on well-defined technical standards (e.g. XML, SOAP, WSDL)
- MDA provides
  - Separation of concerns (vertically and horizontally by various abstraction levels)
  - Leveraging the technology mechanisms for business purposes
  - Positioning of Web Services within holistic view
- Support of Web Services design

V. From Architecture to Running Services


Project Objectives & Constraints
Business Domains
PSM (Platform Specific Model)
Qualify Requirements
Identify Business Services
Model Use Cases
Define Service Implementation
Build & Run
Business Process Definitions
PIM (Platform Independent Model)

From Architecture to Running Services (cont.)

- Qualify Requirements: addresses business process requirements, scope, constraints
- Identify business services from business domains definitions and PIMs (Platform Independent Models)
- Model use cases based on business process definitions
- Define Service Implementation using PSM (Platform Specific Models), producing (technical) service definitions (e.g. WSDL, IDL, etc…), Design level class / interaction diagrams
- Build & Run the defined services
- Software development with iterations

VI. Summary and Outstanding Issues

- The influence of packaged applications:
  - Exposure of critical services
  - Support for Web Services
- Web Services is only a part of the picture:
  - Seamless integration of various technologies is required
  - We must still consider existing integration platforms
- Improvements of Web Services are required:
  - Security
  - Transaction management

Conclusion

A holistic view of an enterprises business services is required.

Web Services promises to be technical enabler of seamless integration at the application layer.

Contact

- Dr. Hans-Peter Hoidn
  IBM Business Consulting Services
  Affolternstr. 56
  +41 1 / 630 36 31
  8050 Zürich
  hans-peter.hoidn@ch.ibm.com