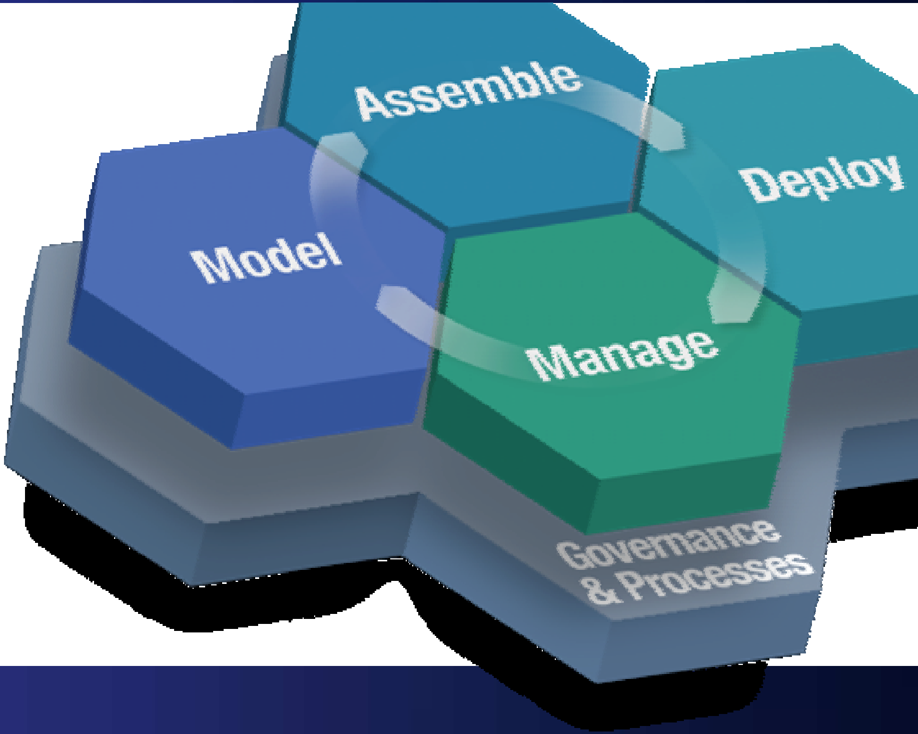
Model and Assemble: Business Driven Development

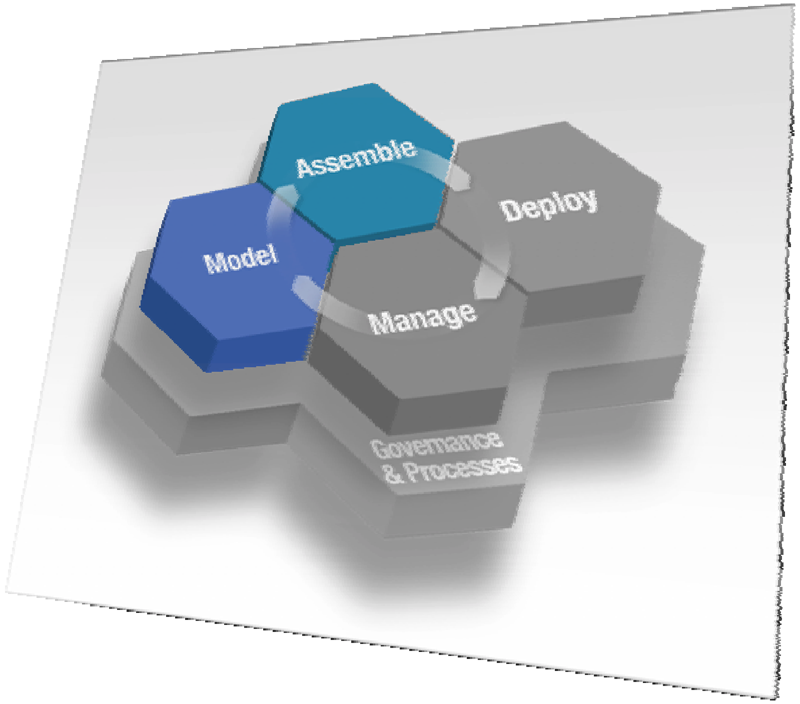
From Business Objectives to Deployed SOA Solutions



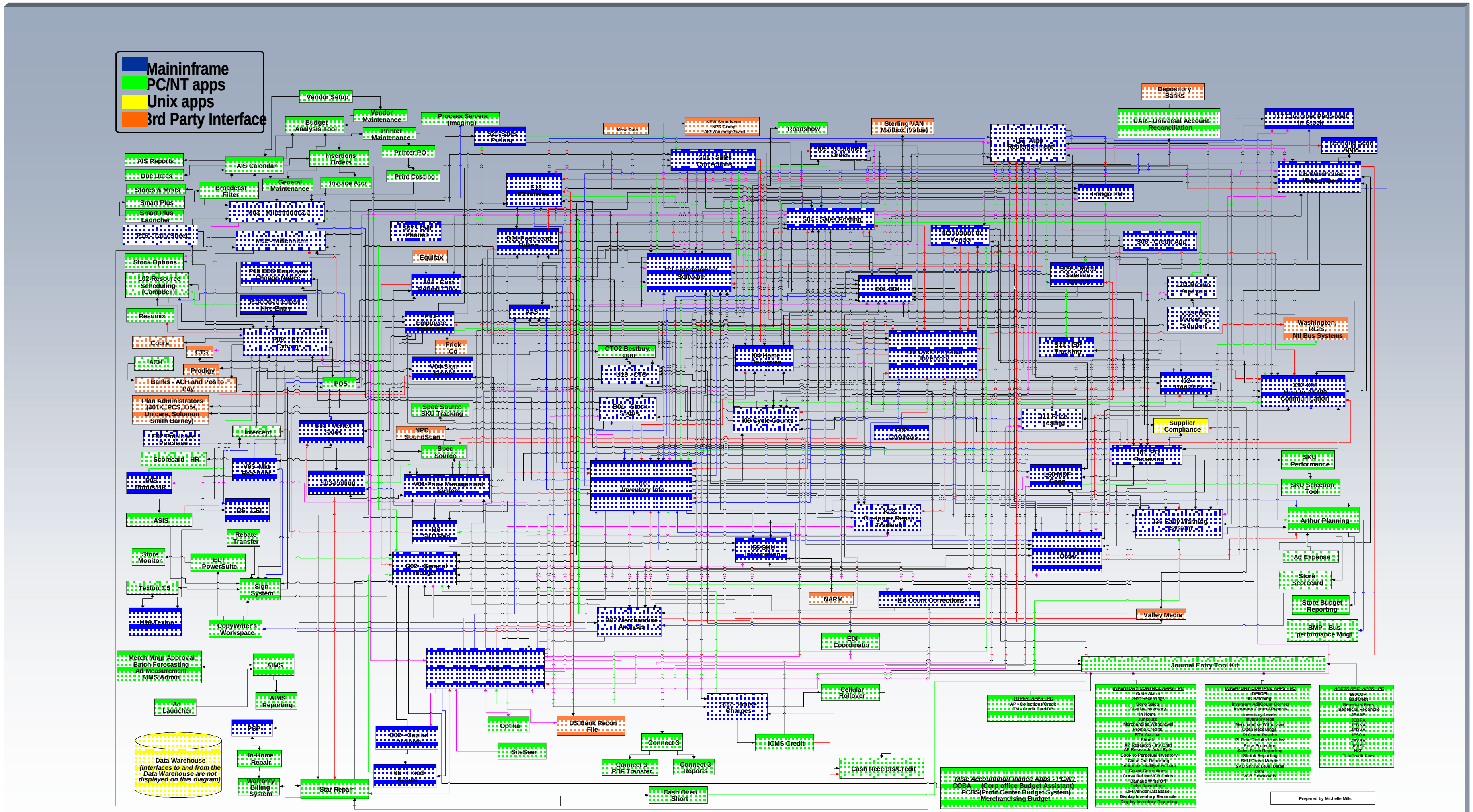
ON DEMAND BUSINESS™

Agenda

- Business Driven Development
  - A development process for deriving solutions from business objectives
- Software Development Platform for BDD and SOA
- Capture Business Goals and Objectives
- Analyze Business Processes to realize Business Goals
- Architect a Services Solution
- Assemble, Deploy & Test
- Summary

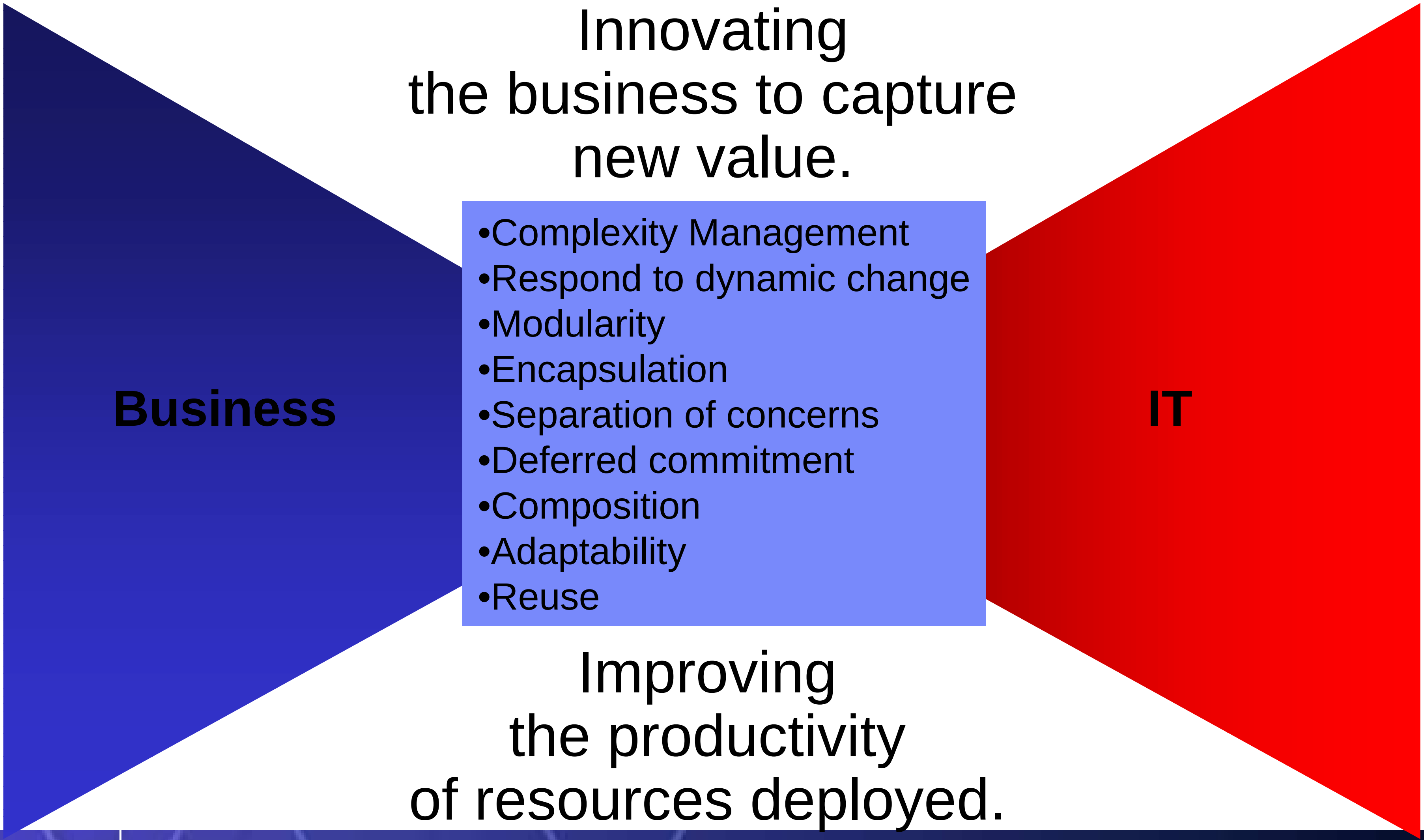


Complexity is Forcing Change


Maininframe
PC/NT apps
Unix apps
3rd Party Interface
Data Warehouse (Interfaces to and from the Data Warehouse are not displayed on this diagram)
Journal Entry Tool Kit
Inventory Info
Prepared by Michelle Mills

Actual Application Architecture for Consumer Electronics Company

The Business and IT have to address similar concerns



Innovating
the business to capture
new value.

- Complexity Management
- Respond to dynamic change
- Modularity
- Encapsulation
- Separation of concerns
- Deferred commitment
- Composition
- Adaptability
- Reuse

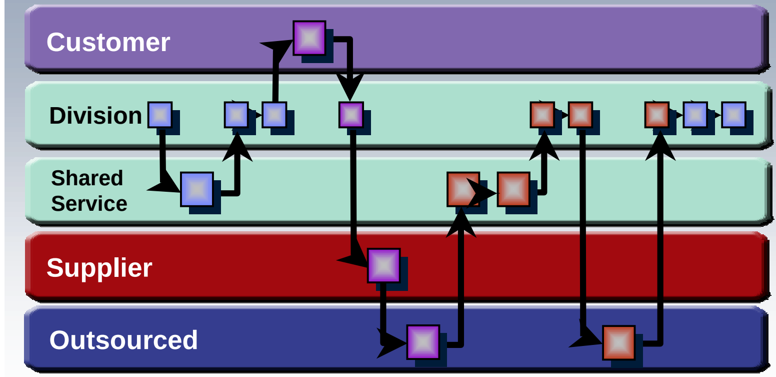
Improving
the productivity
of resources deployed.

Development process can bridge the gap

Pave the Way for Successful Business Innovation

- **Drive development processes and delivered solutions from business goals and objectives**
- Standards (including open source) for interoperability
- Model Driven Architecture (MDA)
- Self-defined, loosely coupled interfaces
- Tools to visualize and integrate existing assets
- Declarative specifications and languages
- Architecture is the key to successful business innovation

Case Study: Procure to Pay Process



What is Business Driven Development?

Business-driven development

An integrated approach to software development that aligns line-of-business, development and operations teams to improve business performance

Development as a business process

- Align Technology and Business priorities
- Improve efficiency and responsiveness
- Create innovative products

Software development becomes a driver of competitive advantage

Three Key Concepts
To Adapt for Business Driven Development

Business Innovation and Optimization

-- Focus on Responsiveness and Optimization

- *A monitoring and management approach that leverages integrated resources to achieve aligned, accountable, and action-oriented business operations*

Model Driven Architecture

-- Focus on Efficiency and Quality

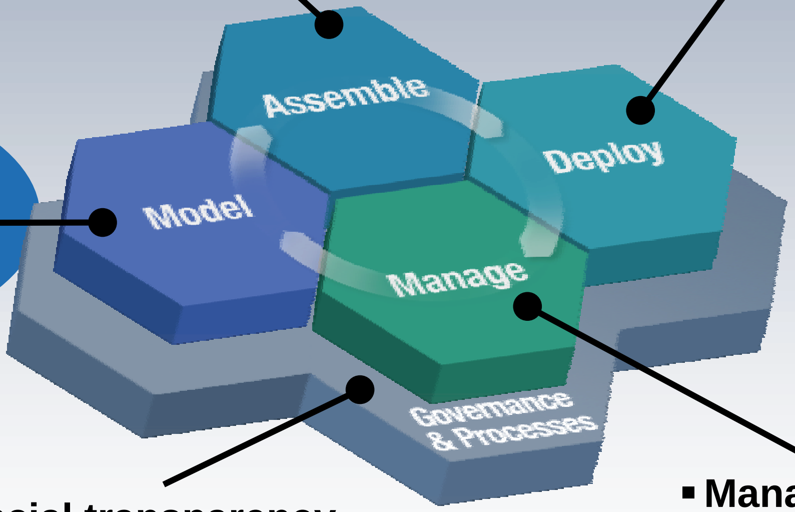
- *A style of enterprise application development and integration based on using automated tools to build system independent models and transform them into efficient implementations.*

Service Oriented Architecture

-- Focus on Flexibility and Reuse

- *An approach for designing and implementing distributed systems that allows a tight correlation between the business model and the IT implementation*

BDD enables business integration, optimization and verification



- Discover
- Construct & Test
- Compose

- Integrate people
- Integrate processes
- Manage and integrate information

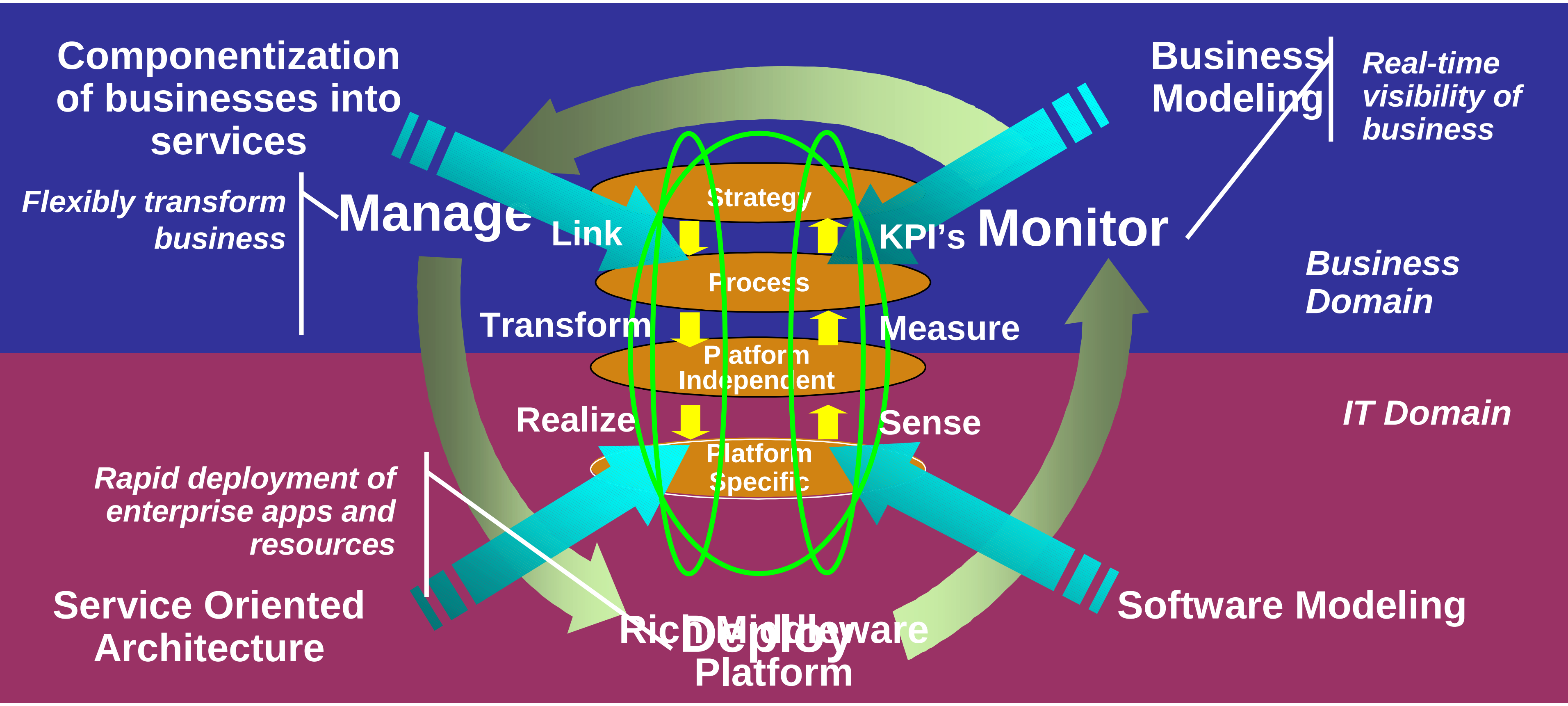
- Gather requirements
- Model & Simulate
- Design

- Financial transparency
- Business/IT alignment
- Process control

- Manage applications & services
- Manage identity & compliance
- Monitor business metrics

The IBM Vision for Business Driven Development

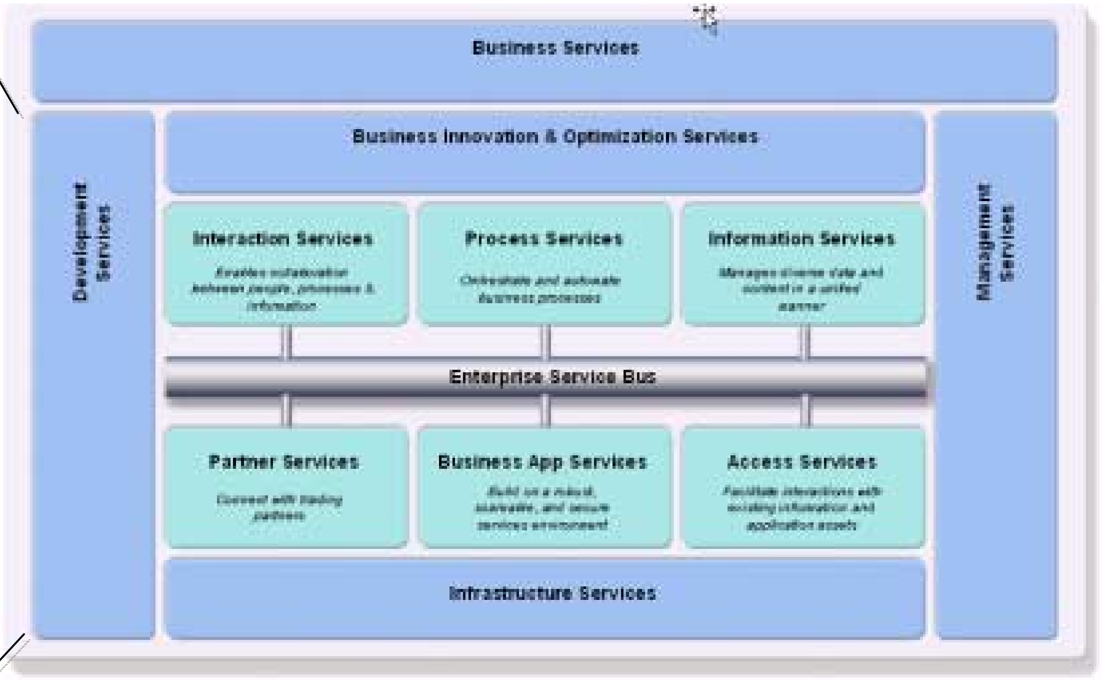
Business applications will be deployed, monitored and managed through the manipulation of multi-level models



***Value*: Accurately and reliably capture and translate business intent into IT solutions**

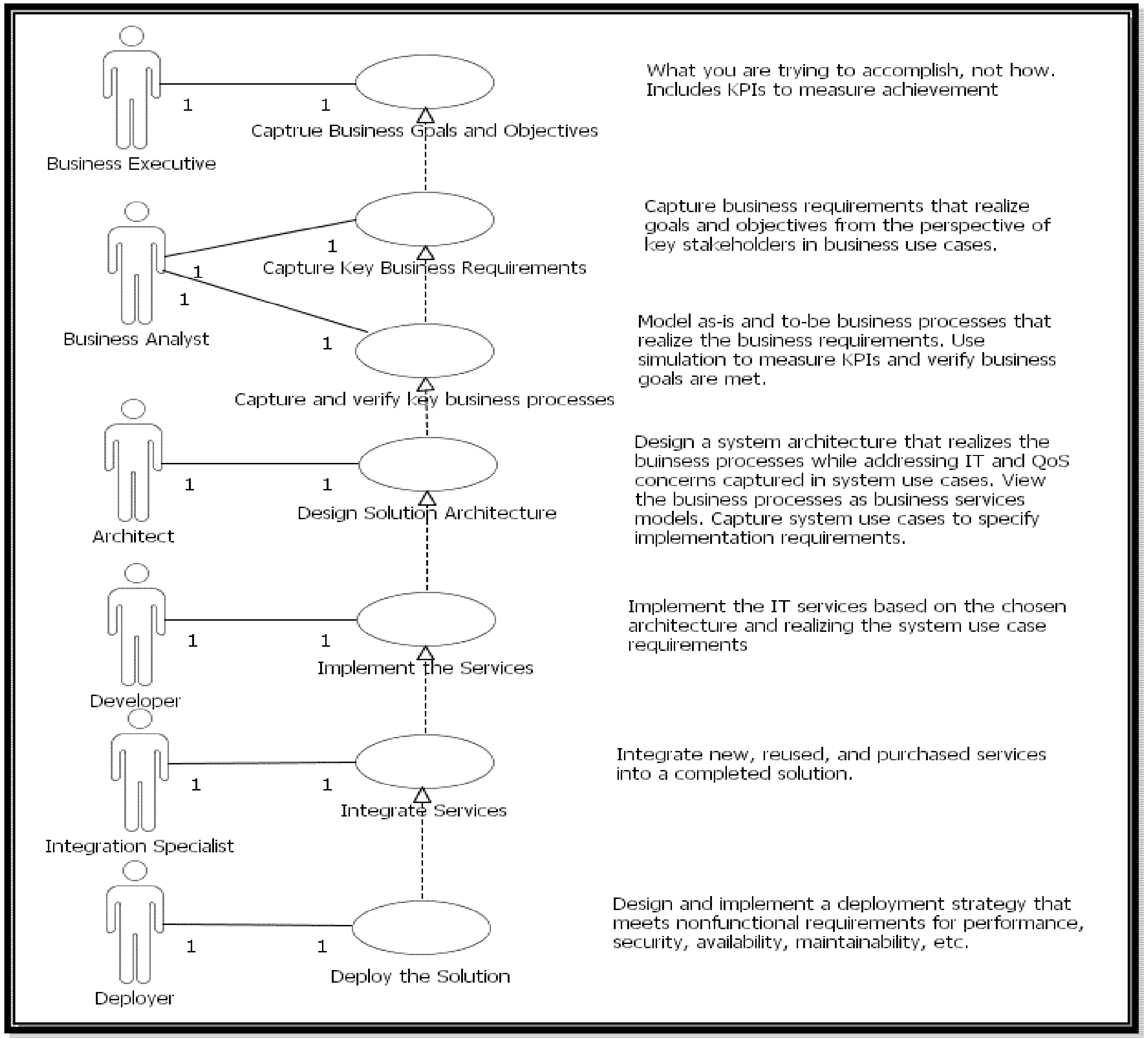
Key Development Phases in Business Driven Development

- **Model**
  - Business Level Modeling
  - Service Oriented Modeling and Design
- **Assemble**
  - Construction of Services (User, Service, Information)
  - Assembly of services (new and existing)
  - Choreography of Services


Business Services
Development Services
Business Innovation & Optimization Services
Management Services
Interaction Services
Enables collaboration between people, processes & information
Process Services
Orchestrate and automate business processes
Information Services
Manages diverse data and content in a unified manner
Enterprise Service Bus
Partner Services
Connect with trading partners
Business App Services
Build on a robust, scaleable, and secure services environment
Access Services
Facilitate interactions with existing information and application assets
Infrastructure Services

- **End result:**
  - A deployed business process and associated services addressing business and IT concerns

BDD consists of Successive Realization



What you are trying to accomplish, not how. Includes KPIs to measure achievement

Capture business requirements that realize goals and objectives from the perspective of key stakeholders in business use cases.

Model as-is and to-be business processes that realize the business requirements. Use simulation to measure KPIs and verify business goals are met.

Design a system architecture that realizes the buinsess processes while addressing IT and QoS concerns captured in system use cases. View the business processes as business services models. Capture system use cases to specify implementation requirements.

Implement the IT services based on the chosen architecture and realizing the system use case requirements

Integrate new, reused, and purchased services into a completed solution.

Design and implement a deployment strategy that meets nonfunctional requirements for performance, security, availability, maintainability, etc.

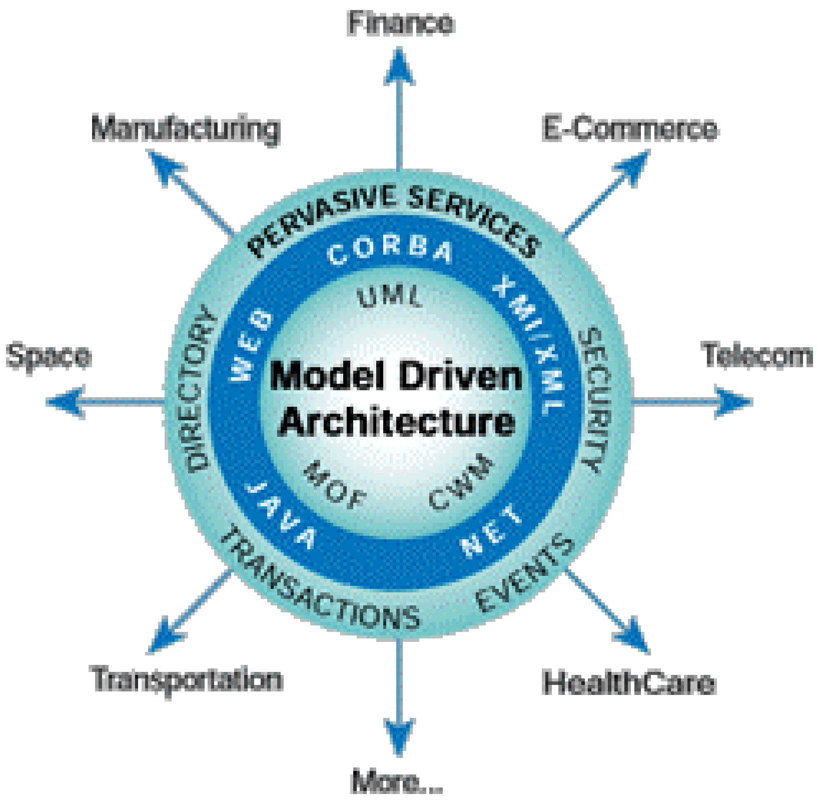
Benefits of Business Driven Development (BDD)

Accelerate development & delivery against *Business Goals*

- More relevant results – *Alignment of Business & IT*
  - Deliver Results to the Business by focusing development on the Business' highest priority projects based on quantifiable data (ReqPro & WBIM Simulation)
- Better results – *Improve Communication b/wn Business & IT*
  - Common language, framework, tools, data, (meta-) models & integrations (WBIM, RSA, ReqPro, WSADIE)
- Accelerated results – *Reduce Development & Deployment Time*
  - Re-use & Governance: Use the tools to leverage work previously done – shared (meta-) models & integrations (WBIM, RSA, ReqPro, WSADIE)
- Lower TCO – *Quickly identify & assess the impact of Business or System components need to Change*
  - Traceability (WBIM, RSA, ReqPro)
  - Don't solve the same problem twice
    - Discover & create re-usable solutions to common problems (patterns) (RSA)
  - Don't recreate common req'ts, business or system components
    - Share &/or Re-use (WBIM, RSA, ReqPro)

MDA : Open standards for Modeling & Development
OMG ™ Model Driven Architecture (MDA)™

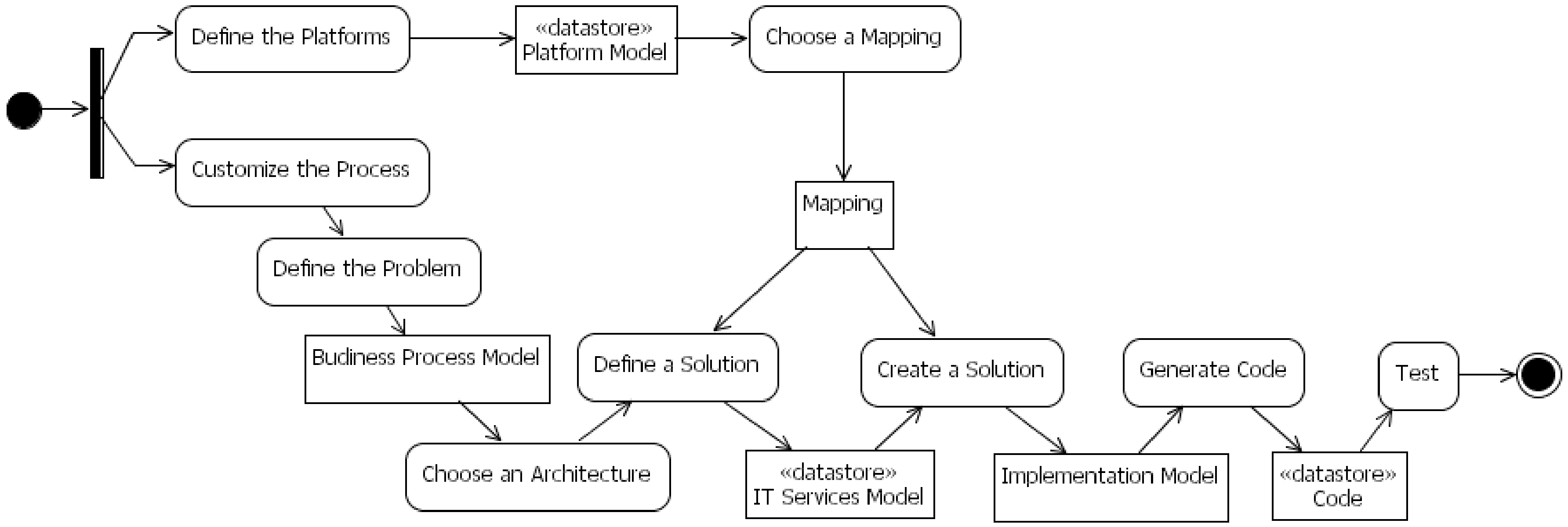
: lopen Of



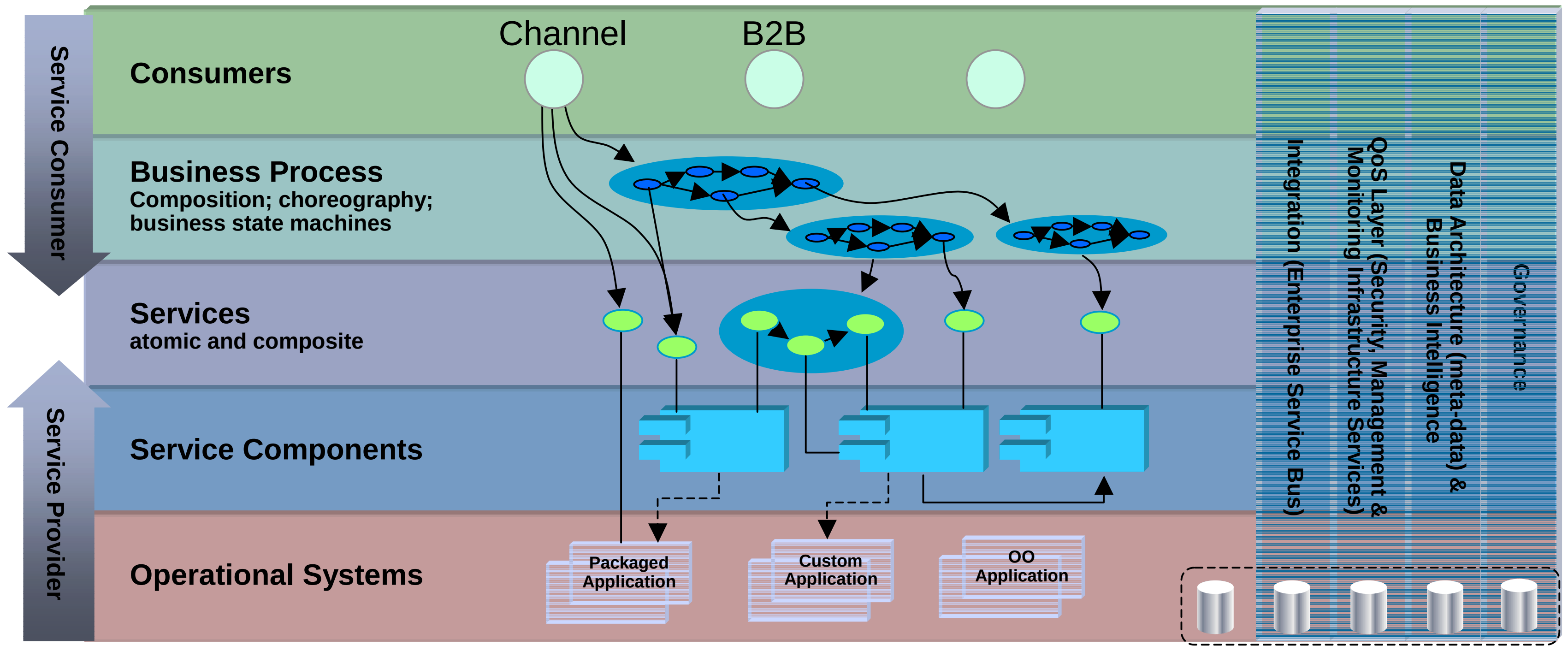
MDA is a Standards Framework & a Open Modeling & Metadata Standards

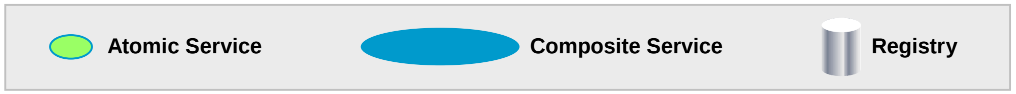
- **An integration of best practices in Modeling, Middleware, Metadata, and Software Architecture**
- **Model Driven (Models at the core - UML, MOF, CWM, BPDM, SBVR, RAS…)**
  - Computation Independent Models (CIM) – Typically Conceptual and Business Models
  - Platform Independent Models (PIM) – Technology or Domain Models – can have logic/computation
  - Platform Specific Models (PSM) - J2EE, .Net, SQL
  - Mappings : PIM<->PIM, PSM<->PSM, PIM<->PSM
  - Applies across the business software life cycle
- **Key Benefits**
  - Improved Productivity for Architects, Designers, Developers and Administrators
  - Lower cost of Application Development and Management
  - Enhanced Portability and Interoperability
  - Business Models and Technologies evolve at their own pace on platform(s) of choice

MDA Process in a Nutshell



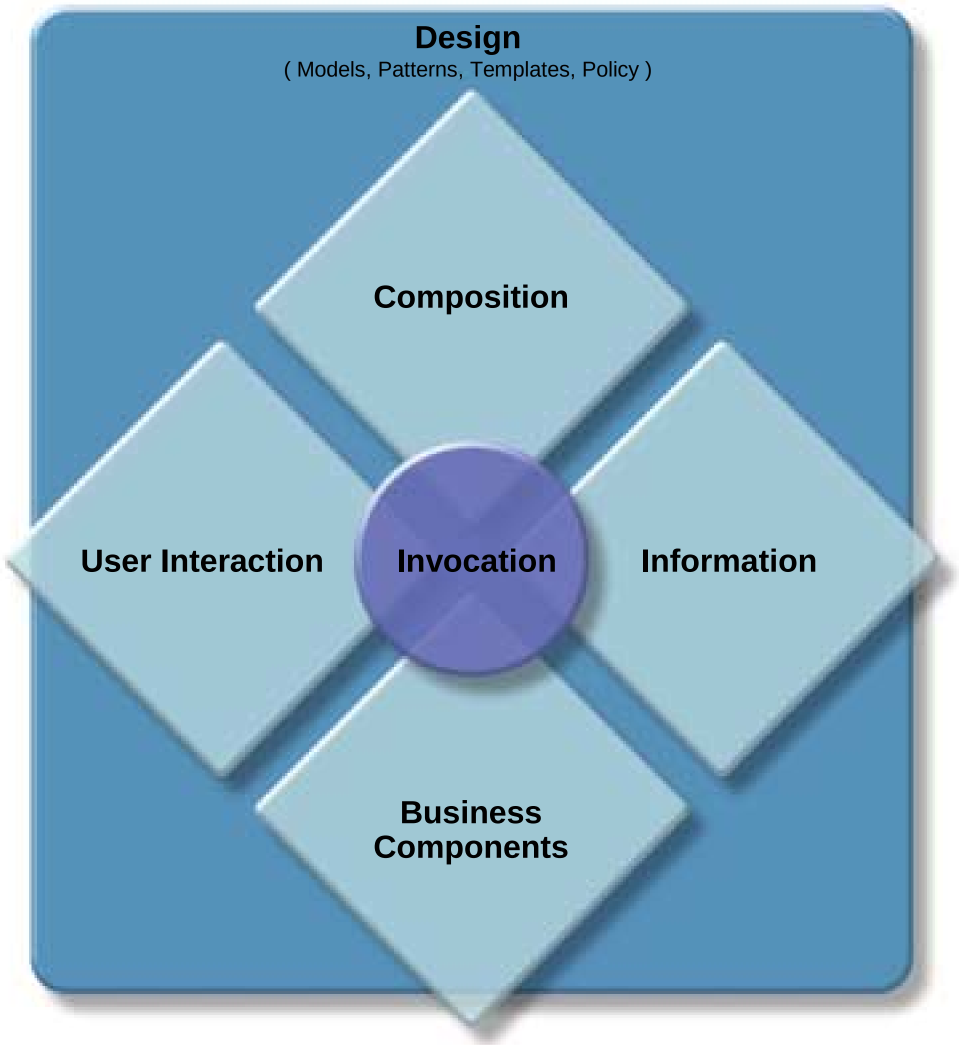
Moving to Services-Oriented Solutions


Service Consumer
Service Provider
Consumers
Channel
B2B
Business Process
Composition; choreography; business state machines
Services
atomic and composite
Service Components
Operational Systems
Packaged Application
Custom Application
OO Application
Integration (Enterprise Service Bus)
QoS Layer (Security, Management & Monitoring Infrastructure Services)
Data Architecture (meta-data) & Business Intelligence
Governance



SOA Programming Model

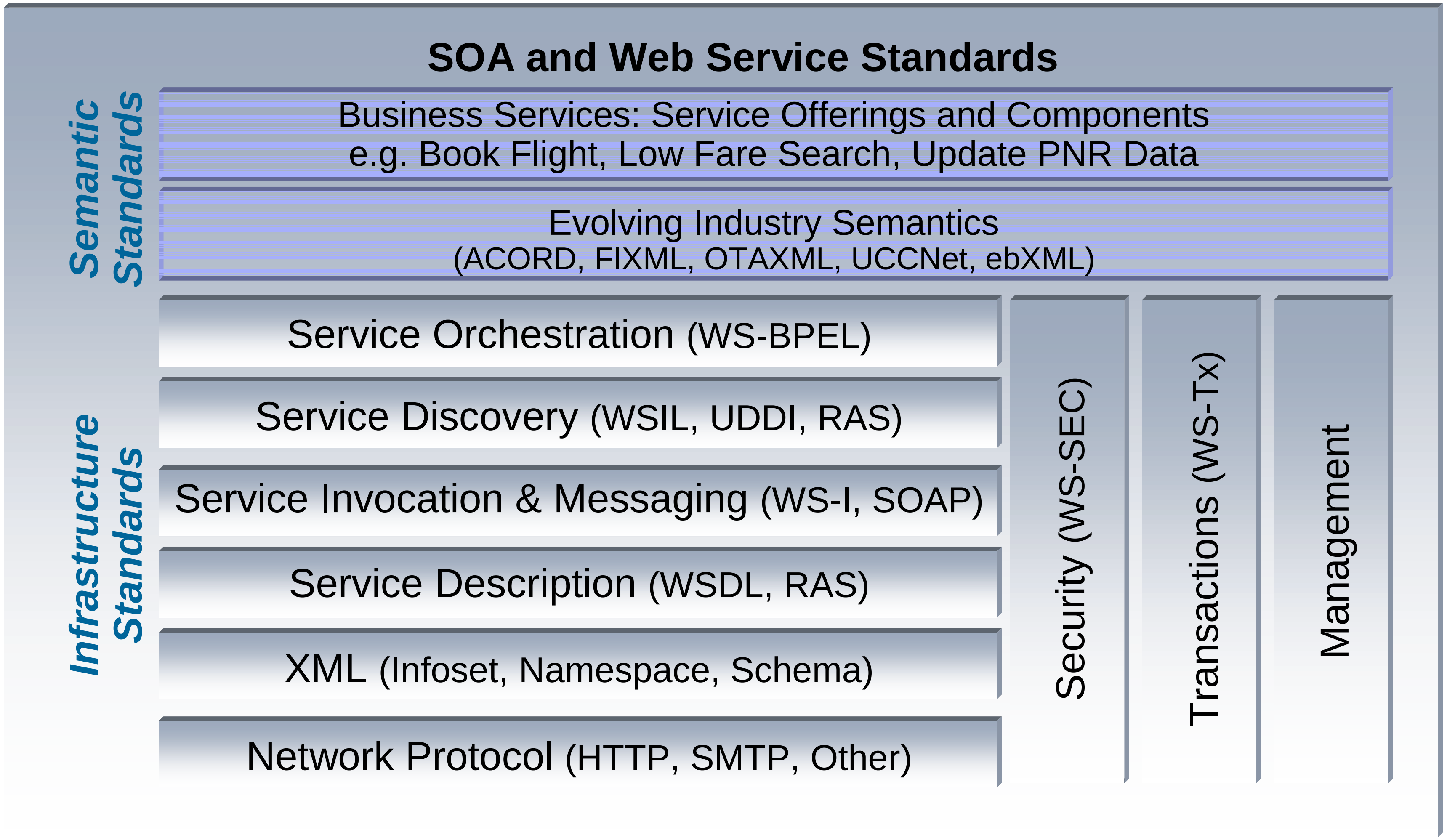
- **User Interaction**
  - Dynamic support for people integration into the business design
- **Business Components**
  - Composable and reusable services
- **Information**
  - Built-in access to service state, disconnected service-data exchange, information composition and transformation
- **Composition of Business-level Applications**
  - Wired assembly of services to form business-level applications, workflows, and business orchestration
- **Invocation**
  - Loosely-coupled call-style and event-driven interconnection of services with built-in support for topology transparency, mediation, and brokering featuring standards-based interoperability
- **Design**
  - Focus on business design modeling, simplification, and role-based collaboration
  - Use of declarative policy to control execution behavior and relationships



SOA Programming Model

- **JavaServer Faces**
  - Standard way to construct user interfaces for web applications, portlet applications, etc.
  - MVC based Architecture
- **Service Component Architecture (SCA)**
  - Component services programming model which provides a consistent framework for assembling solutions
  - Jointly developed/endorsed by IBM, BEA, IONA, Oracle, SAP, and Sybase
  - Apache Open Source Incubator Project
    - http://incubator.apache.org/tuscany/
- **Service Data Objects (SDO)**
  - Uniform (technology independent) way to represent data
  - Provides *Single abstraction* (common API) across JDBC ResultSet, JCA Record, XML DOM, JAXB, Entity EJB, CMI (for MQ messages), and so on
  - Co-developed by IBM and BEA
- **Business Process Execution Language (WS-BPEL)**
  - Standard way to choreograph business processes
  - Standardization through OASIS

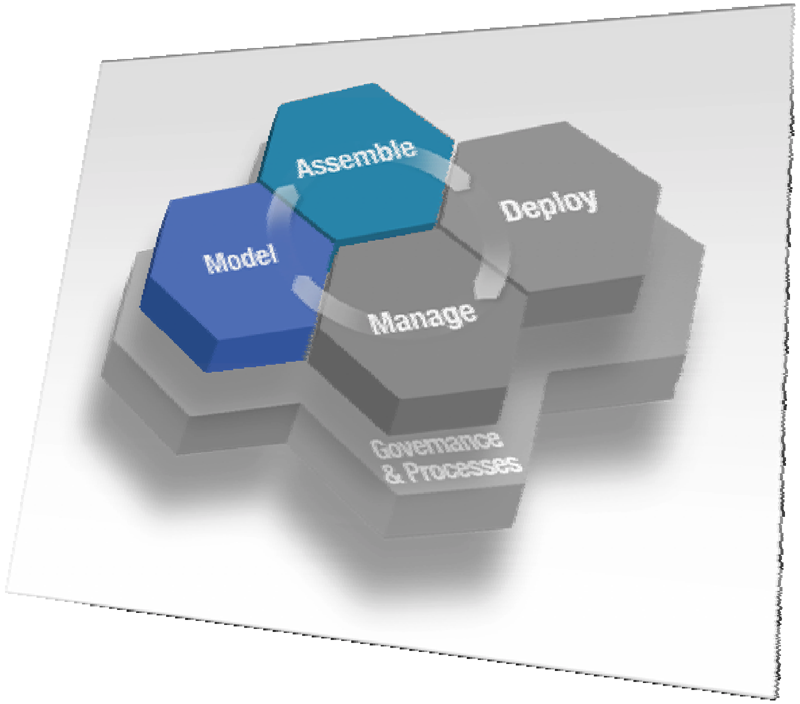
Key Standards and Technologies
Used in Business Driven Development



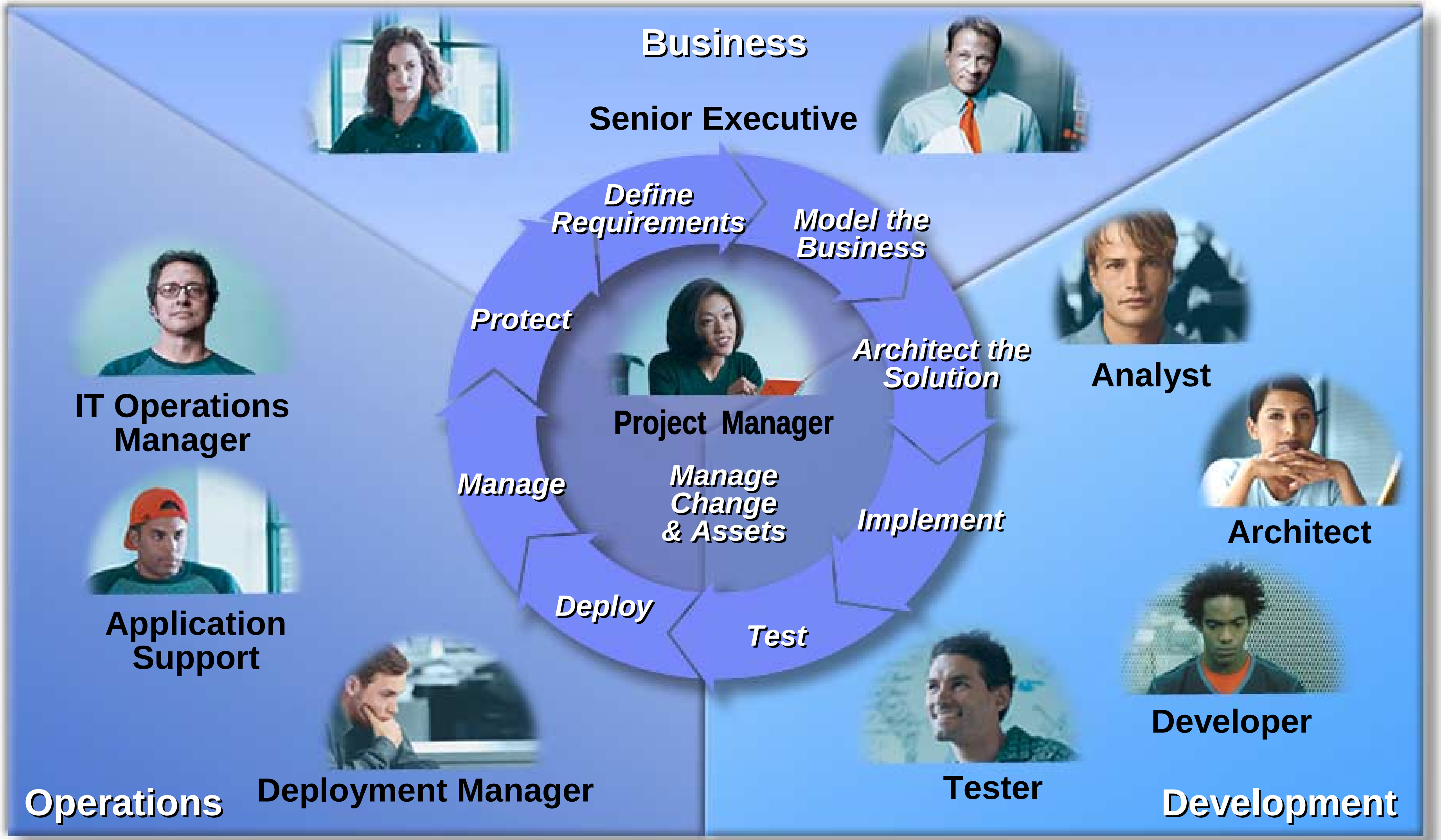
SOA and Web Service Standards
Semantic Standards
Business Services: Service Offerings and Components
e.g. Book Flight, Low Fare Search, Update PNR Data
Evolving Industry Semantics
(ACORD, FIXML, OTAXML, UCCNet, ebXML)
Infrastructure Standards
Service Orchestration (WS-BPEL)
Service Discovery (WSIL, UDDI, RAS)
Service Invocation & Messaging (WS-I, SOAP)
Service Description (WSDL, RAS)
XML (Infoset, Namespace, Schema)
Network Protocol (HTTP, SMTP, Other)
Security (WS-SEC)
Transactions (WS-Tx)
Management

Agenda

- Business Driven Development
  - A development process for deriving solutions from business objectives
- Software Development Platform for BDD and SOA
- Capture Business Goals and Objectives
- Analyze Business Processes to realize Business Goals
- Architect a Services Solution
- Assemble, Deploy & Test
- Summary



The Business Driven Development lifecycle



The IBM Rational Software Development Platform

Analyst

Model the business & identify the business services

Architect

Design the services architecture

Analyze assets for reuse

Developer

Construct the services

Assemble & deploy the composite application

Tester

Test the individual services & composite application

Deployment Manager

Provision, configure, tune and troubleshoot composite applications

Project Manager

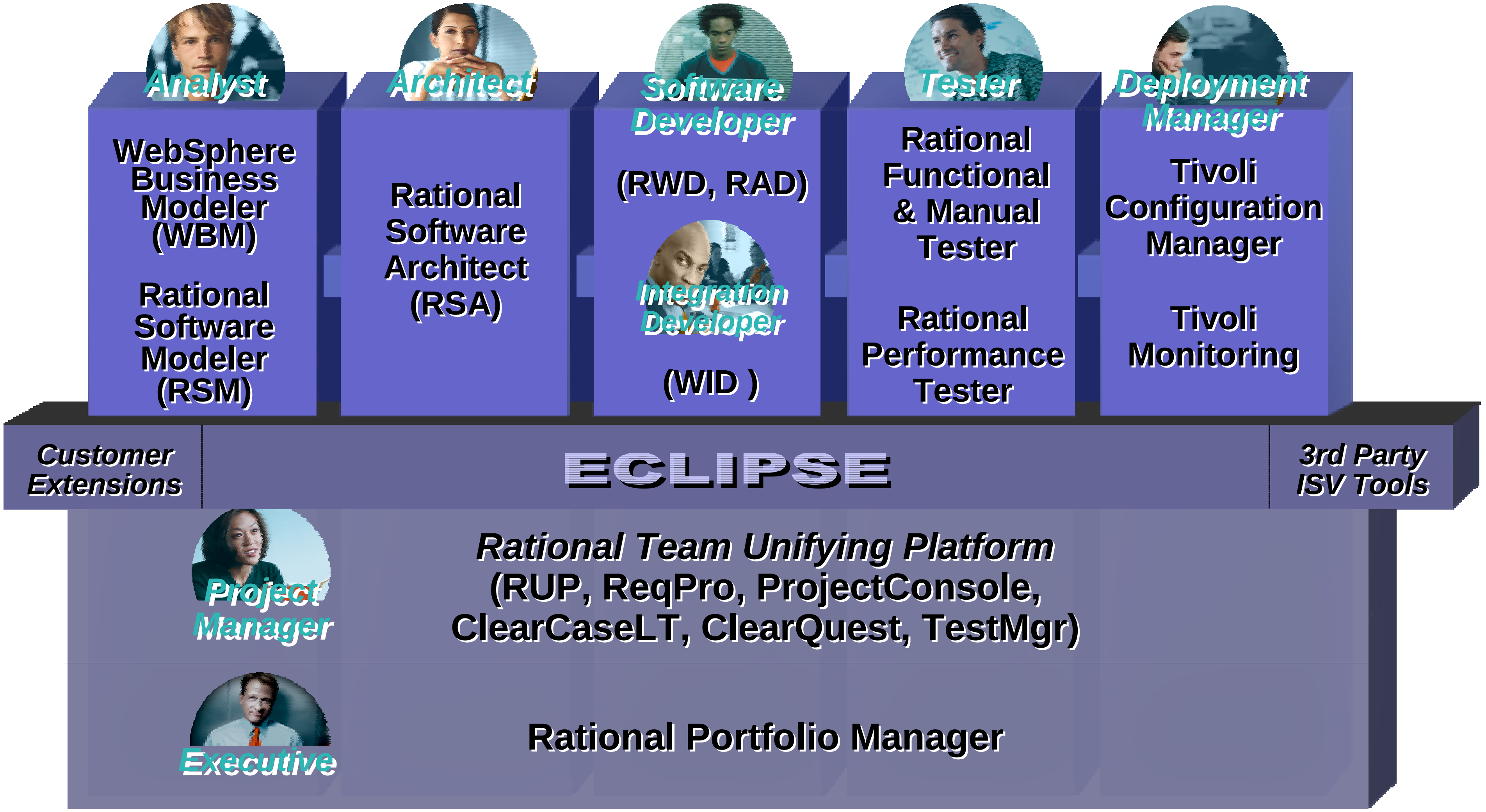
- Follow a service-oriented process
- Manage requirements
- Manage SOA assets
- Manage quality

IT Executive

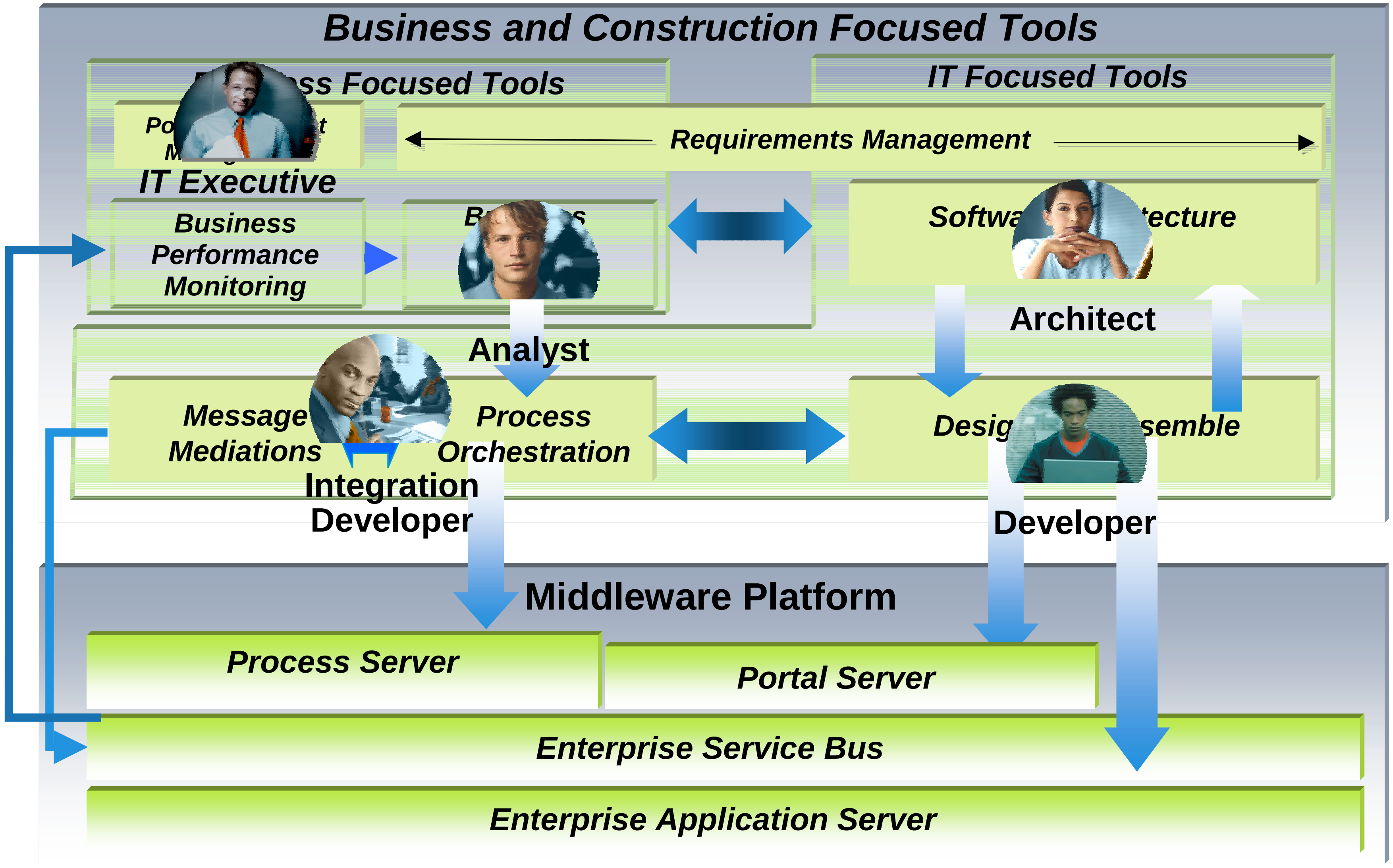
- Align business strategy with IT execution
- Govern SOA – align, execute and control investments

The IBM Software Development Platform

A complete, open, modular, and proven solution


Analyst
WebSphere Business Modeler (WBM)
Rational Software Modeler (RSM)
Architect
Rational Software Architect (RSA)
Software Developer
(RWD, RAD)
Integration Developer
(WID)
Tester
Rational Functional & Manual Tester
Rational Performance Tester
Deployment Manager
Tivoli Configuration Manager
Tivoli Monitoring
Customer Extensions
ECLIPSE
3rd Party ISV Tools
Project Manager
Rational Team Unifying Platform
(RUP, ReqPro, ProjectConsole, ClearCaseLT, ClearQuest, TestMgr)
Executive
Rational Portfolio Manager

Business Driven Development Scenario

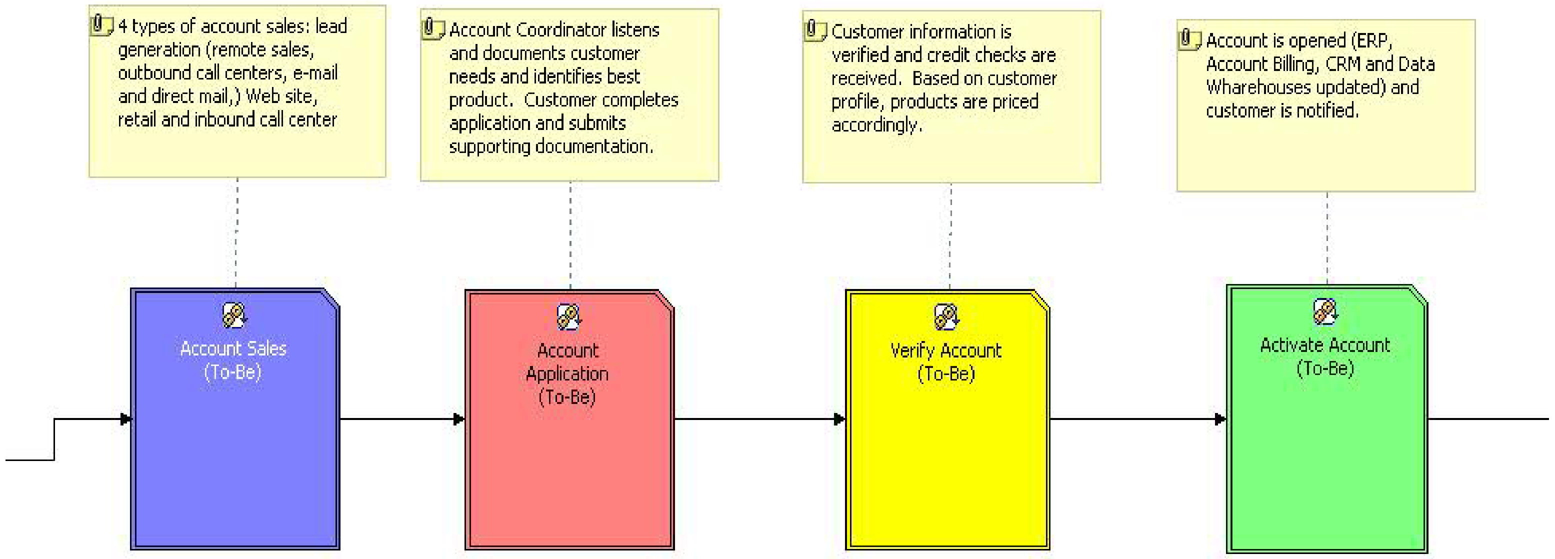


Example: JK Enterprises – Account Management Project

- Organization Description
  - Multiple Market Segments – Mass Market, SMB, Corporate
  - Multiple Channels – Web, Brick & Mortar, Remote Sales, Back Office
  - Struggling to grow uniformly at low cost and risk
- IT topology:
  - Mix of build vs. buy – SAP, Siebel, CICS, Batch
  - Legacy Applications – core accounting, order entry
  - Multiple Platforms - Windows, AIX, iSeries, zSeries
  - Heterogeneous topology
  - Very little reuse of components and skills
- Numerous Targets for BPO but no methodology and way forward
- *This scenario is based on work for real customers*

JK Enterprises Account Open Process Overview

- Account Verification is one of four steps in the whole business process
- Use best practices to build out the solution
  - Use Rational RequisitePro to capture business goals and objectives
  - Use WebSphere Business Modeler to describe the business process
  - Use RSA to implement new services
  - Use WebSphere Integration Developer to integrate services


4 types of account sales: lead generation (remote sales, outbound call centers, e-mail and direct mail,) Web site, retail and inbound call center
Account Coordinator listens and documents customer needs and identifies best product. Customer completes application and submits supporting documentation.
Customer information is verified and credit checks are received. Based on customer profile, products are priced accordingly.
Account is opened (ERP, Account Billing, CRM and Data Wharehouses updated) and customer is notified.
Account Sales (To-Be)
Account Application (To-Be)
Verify Account (To-Be)
Activate Account (To-Be)

JK Enterprises Challenge #1 – Account Open

- **Account Sales:**
  - Latency – customers want to use their accounts right away
  - Incomplete Status available during setup process – can’t answer customer questions
  - Inconsistent response on completion or when issues arise
  - Lack of collaborative materials to ease pre-sale (chat, co-browsing, difficult to ‘find’ information)
  - Limited sales channels
- **Account Application:**
  - Complex application forms
  - Different applications for different products
  - Errors due to re-keying of information from paper application to screen
  - Lack of single customer view
- **Account Verification:**
  - Lack of digital imaging
  - Lack of single customer view
  - Declining too many customer requests
  - Lack of dynamic, rules-based pricing
  - Credit checks inconsistent, expensive, and time consuming
- **Account Activation:**
  - Prolonged process due to lack of digital signature
  - Manually entering information in various internal systems (CRM, ERP, etc.)

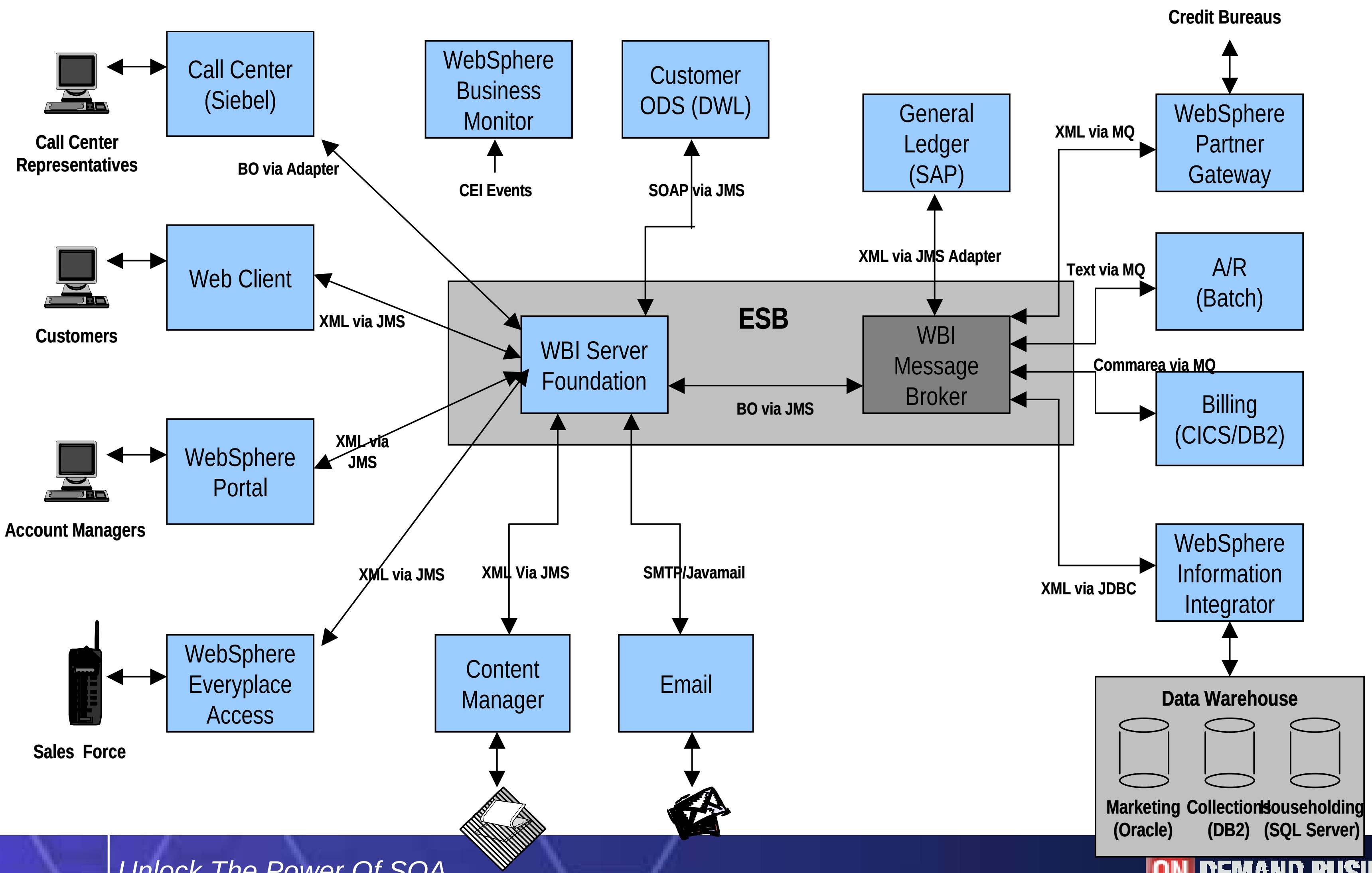
Account Open – Business Goals

- Strategic Initiative - Optimize Account Setup
  - Reduce Cost and Latency – grow efficiently and profitably
  - Improve sales and customer service through increased speed and responsiveness
  - Enhance productivity through reductions in total cost of ownership (TCO)
  - Increase the value and reach of services and products
  - Reduce risk and increase consistency via rules based Business Process Management

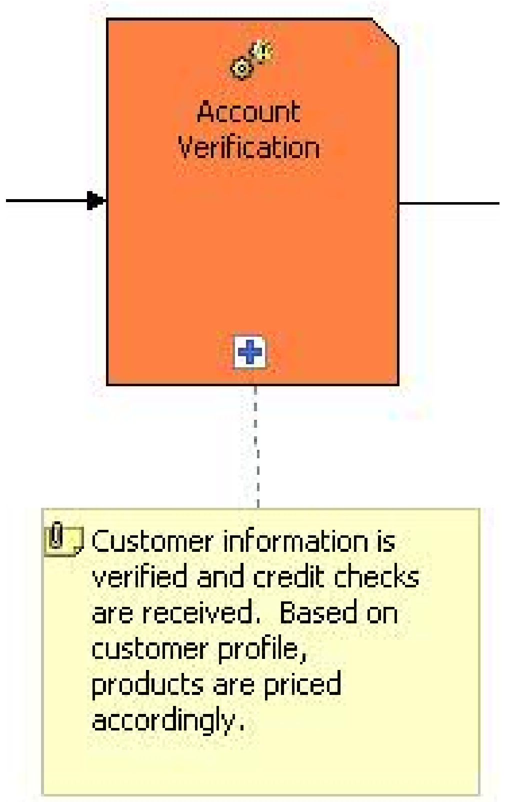
Account Open – IT Implementation

- Today
  - 200+ Developers
  - 20+ different development tools
  - No end-end methodology
  - Lacking governance model
  - About to embark on major development supporting BPO
- Tomorrow
  - Supporting methodology with artifacts and contracts defined
  - Supported and integrated tools
  - Role based
  - Business participates

Account Open – IT Solution Topology



Account Open – Account Verification Process

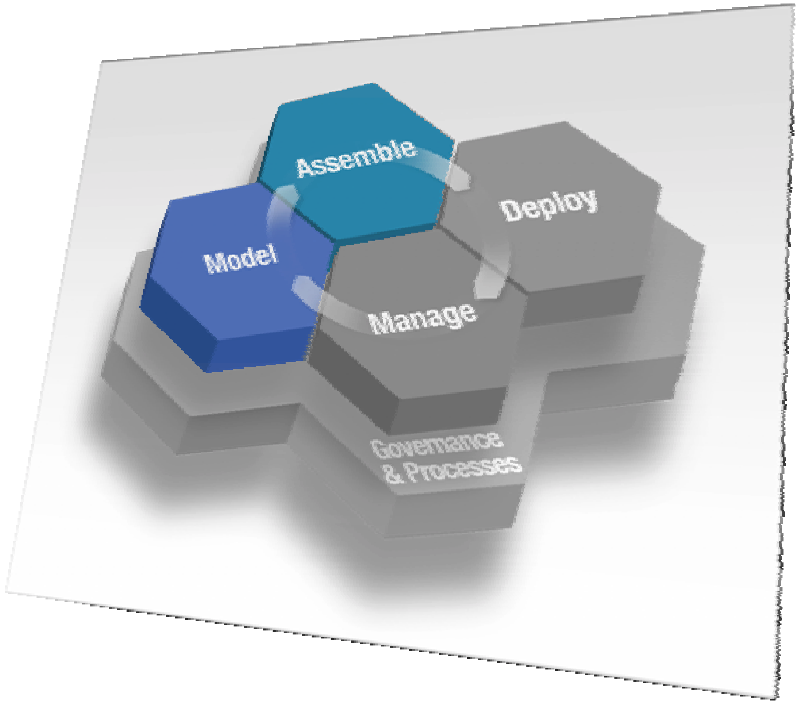


Customer information is verified and credit checks are received. Based on customer profile, products are priced accordingly.

- Customer information is verified
- Retrieve Credit Report if necessary
- If credit score is “bad”, additional documentation is required
- Based on customer profile (and credit score), pricing plan is calculated
- Manual application review, depending on score
- Generate decline or proceed to Account Activation step

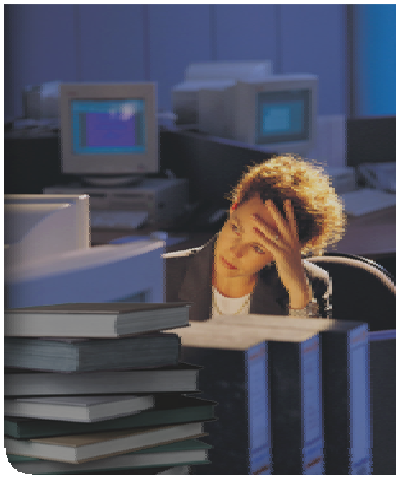
Agenda

- Business Driven Development
  - A development process for deriving solutions from business objectives
- Software Development Platform for BDD and SOA
- Capture Business Goals and Objectives
- Analyze Business Processes to realize Business Goals
- Architect a Services Solution
- Assemble, Deploy & Test
- Summary



Capture Business Goals and Objectives

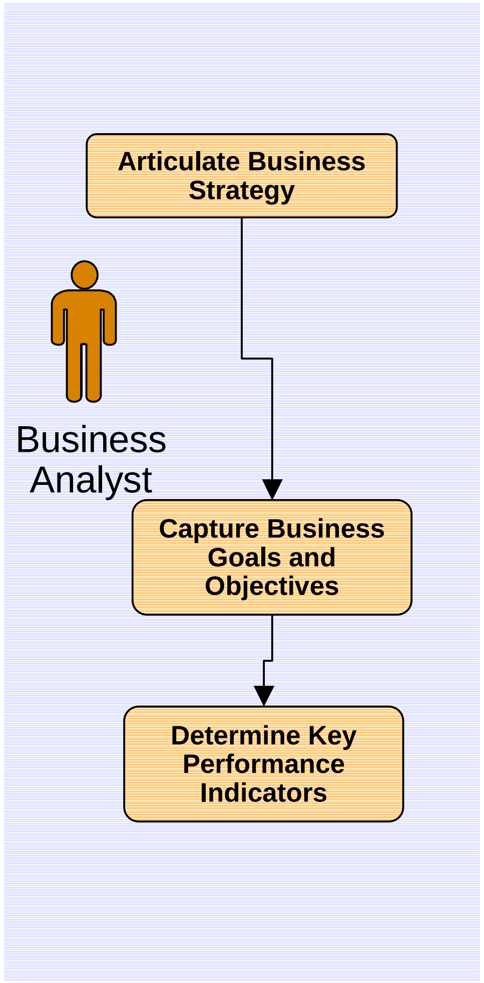
- **Understanding and using requirements**
  - **Know what you want to accomplish and how to measure success**
  - Centrally manage requirements located in many documents, charts, and models
  - Provide context for business value
  - Make available to designers, developers and testers
- **Organizing and reporting on requirements**
  - Assign priority, risk and level of effort
  - Support different requirement types
- **Managing changes to requirements**
  - Maintain record of relationships and origin
  - Communicate changes in a timely manner
  - Assess and understand change impact

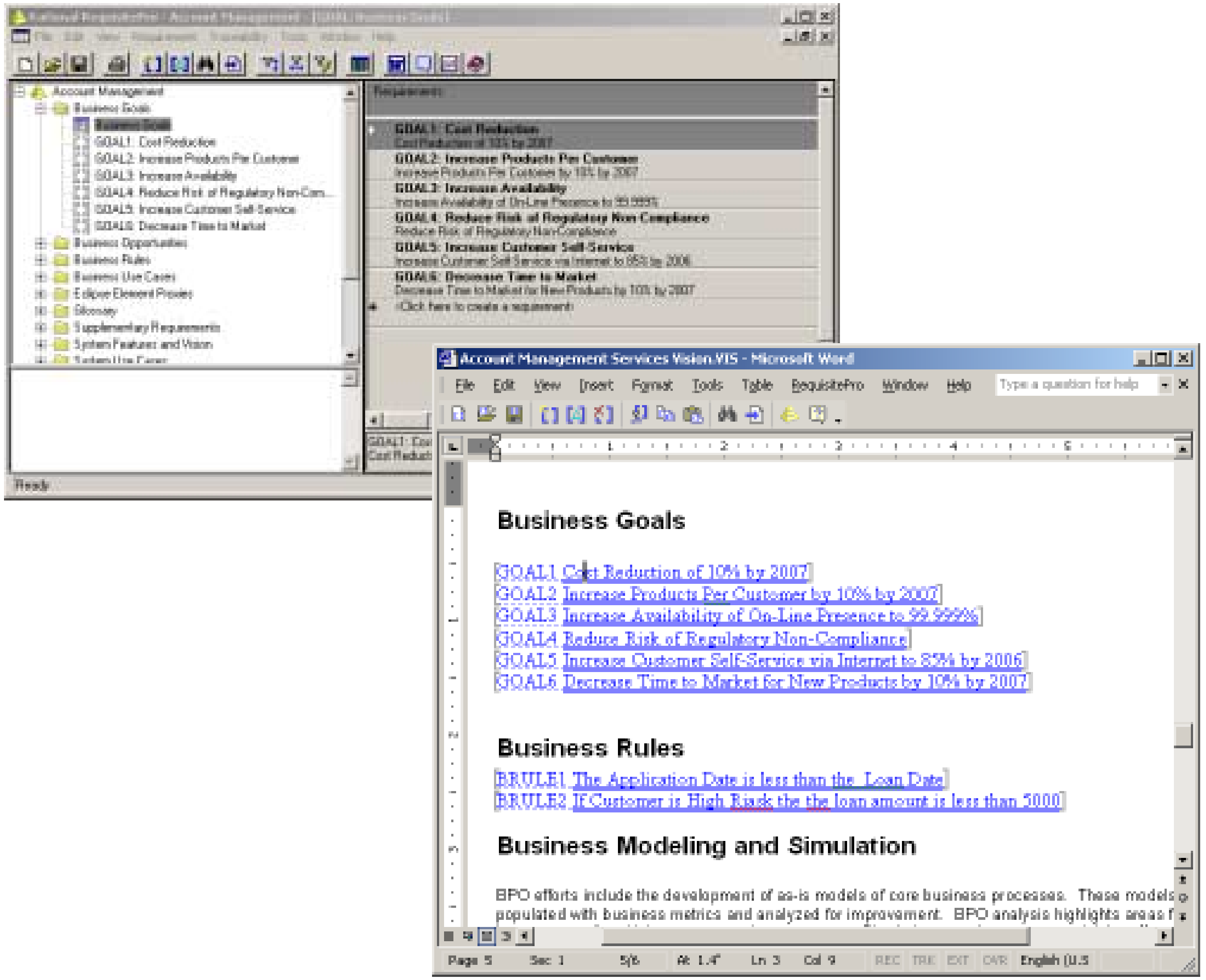


Unsolved RM Challenges ➔ Software Rework ➔ Cost, Delays, Quality Issues

Capture Business Goals and Objectives and KPIs

Rational RequisitePro


Articulate Business Strategy
Business Analyst
Capture Business Goals and Objectives
Determine Key Performance Indicators



GOAL1: Cost Reduction
GOAL2: Increase Products Per Customer
GOAL3: Increase Availability
GOAL4: Reduce Risk of Regulatory Non-Com...
GOAL5: Increase Customer Self-Service
GOAL6: Decrease Time to Market

Requirements
GOAL1: Cost Reduction Cost Reduction of 10% by 2007
GOAL2: Increase Products Per Customer Increase Products Per Customer by 10% by 2007
GOAL3: Increase Availability Increase Availability of On-Line Presence to 99.999%
GOAL4: Reduce Risk of Regulatory Non Compliance Reduce Risk of Regulatory Non-Compliance
GOAL5: Increase Customer Self-Service Increase Customer Self-Service via Internet to 85% by 2006
GOAL6: Decrease Time to Market Decrease Time to Market for New Products by 10% by 2007
<Click here to create a requirement>

Business Goals

GOAL1 Cost Reduction of 10% by 2007
GOAL2 Increase Products Per Customer by 10% by 2007
GOAL3 Increase Availability of On-Line Presence to 99.999%
GOAL4 Reduce Risk of Regulatory Non-Compliance
GOAL5 Increase Customer Self-Service via Internet to 85% by 2006
GOAL6 Decrease Time to Market for New Products by 10% by 2007

Business Rules

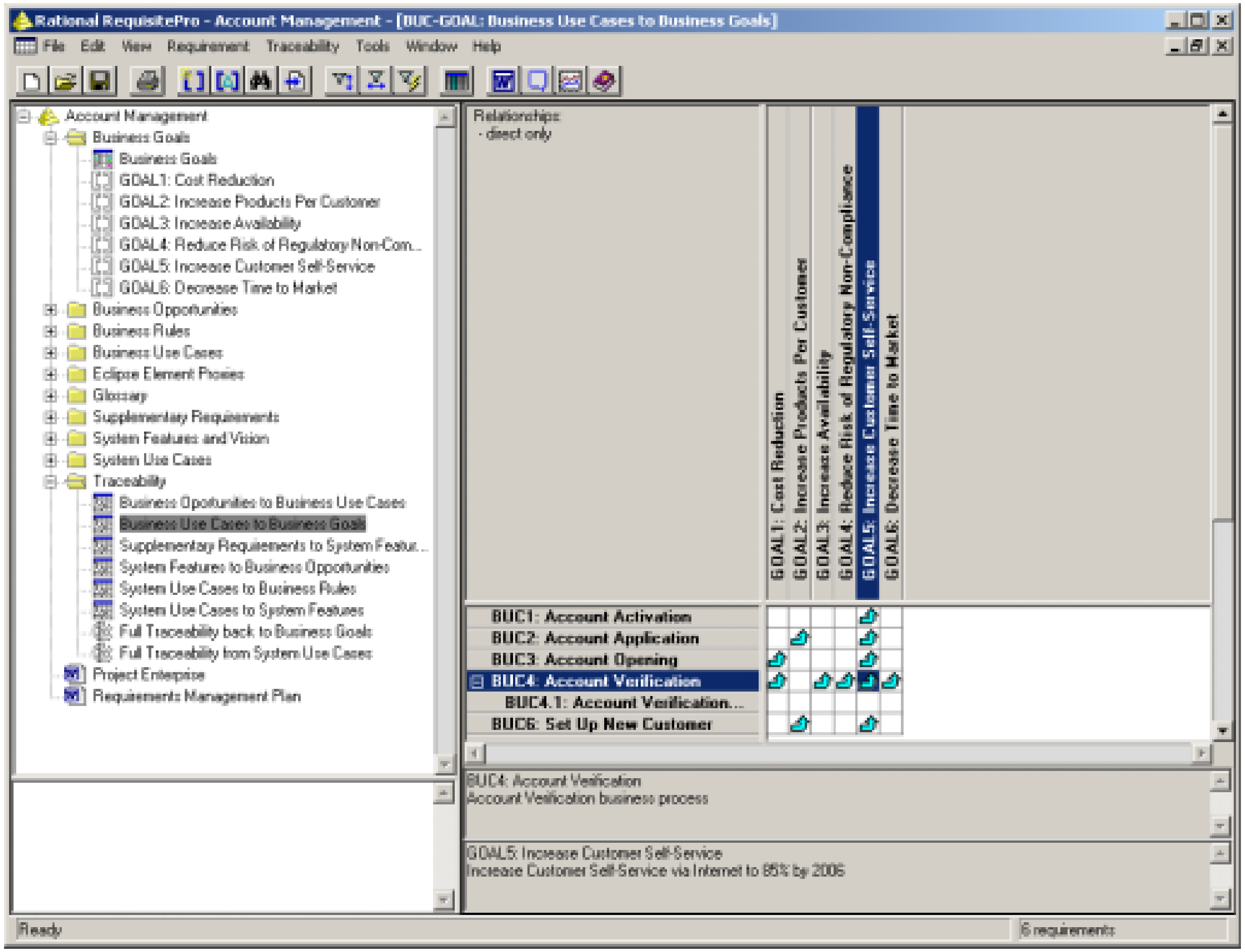
BRULE1 The Application Date is less than the Loan Date
BRULE2 If Customer is High Risk the the loan amount is less than 5000

Business Modeling and Simulation

BPO efforts include the development of as-is models of core business processes. These models populated with business metrics and analyzed for improvement. BPO analysis highlights areas f

Page 5 Sec 1 5/6 At 1.4" Ln 3 Col 9 REC TRK EXT OVR English (U.S

Trace Business Use Cases to Business Goals they Realize



Rational RequisitePro - Account Management - [BUC-GOAL: Business Use Cases to Business Goals]

File Edit View Requirement Traceability Tools Window Help

- Account Management
  - Business Goals
    - Business Goals
    - GOAL1: Cost Reduction
    - GOAL2: Increase Products Per Customer
    - GOAL3: Increase Availability
    - GOAL4: Reduce Risk of Regulatory Non-Com...
    - GOAL5: Increase Customer Self-Service
    - GOAL6: Decrease Time to Market
  - Business Opportunities
  - Business Rules
  - Business Use Cases
  - Eclipse Element Proxies
  - Glossary
  - Supplementary Requirements
  - System Features and Vision
  - System Use Cases
  - Traceability
    - Business Opportunities to Business Use Cases
    - Business Use Cases to Business Goals
    - Supplementary Requirements to System Featur...
    - System Features to Business Opportunities
    - System Use Cases to Business Rules
    - System Use Cases to System Features
    - Full Traceability back to Business Goals
    - Full Traceability from System Use Cases
  - Project Enterprise
  - Requirements Management Plan

Relationships:
- direct only

	GOAL1: Cost Reduction	GOAL2: Increase Products Per Customer	GOAL3: Increase Availability	GOAL4: Reduce Risk of Regulatory Non-Compliance	GOAL5: Increase Customer Self-Service	GOAL6: Decrease Time to Market
BUC1: Account Activation					✓	
BUC2: Account Application		✓			✓	
BUC3: Account Opening	✓				✓	
BUC4: Account Verification	✓		✓	✓	✓	✓
BUC4.1: Account Verification...						
BUC6: Set Up New Customer		✓			✓	

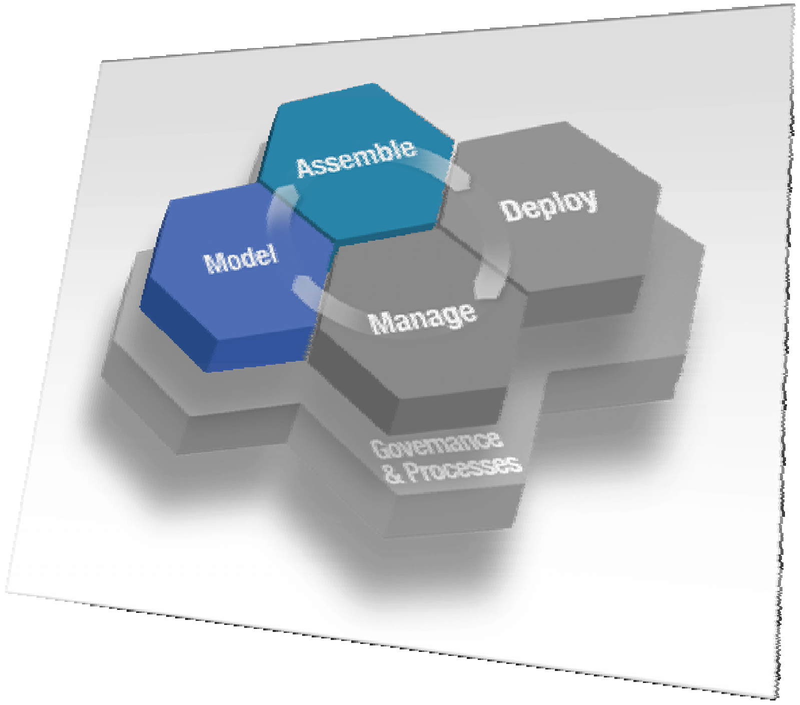
BUC4: Account Verification
Account Verification business process

GOAL5: Increase Customer Self-Service
Increase Customer Self-Service via Internet to 85% by 2006

Ready | 6 requirements

Agenda

- Business Driven Development
  - A development process for deriving solutions from business objectives
- Software Development Platform for BDD and SOA
- Capture Business Goals and Objectives
- Analyze Business Processes to realize Business Goals
- Architect a Services Solution
- Assemble, Deploy & Test
- Summary



Analyze Business Processes to realize Business Goals

The project manager Pam selected a proposal for improving the Account Application and Verification process as one to pursue to achieve the business objectives. Pam reviewed the available resources, and assigned the work to the Business Analyst Bob.

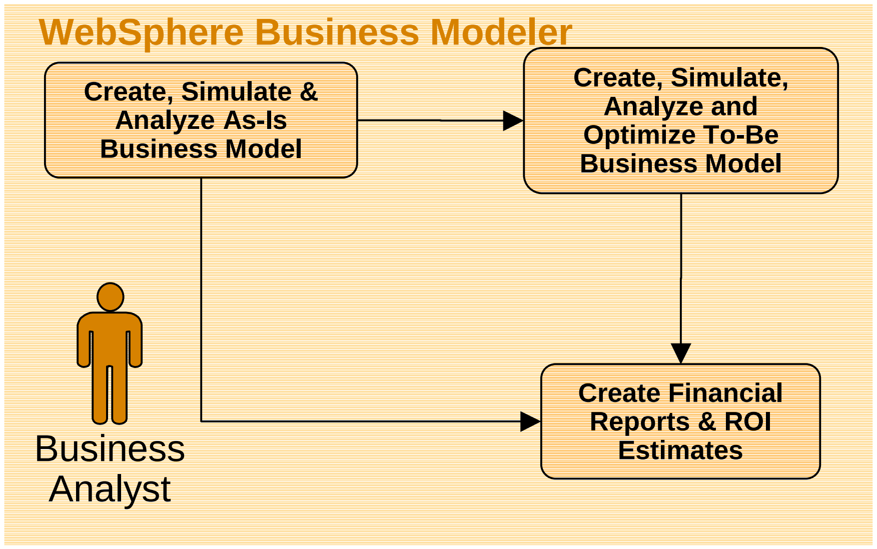
Bob assesses the service automation opportunities within JK Enterprise's existing Account Application and Account Verification process. Bob runs a simulation of the process using data provided by the BPO team. He finds that human-based tasks are the most expensive and the decline case is the most expensive of all – occurring nearly 25% of the time. Given the business goals and objectives that drove the work item request, Bob adds an automated step to the application review process to eliminate the manual review of many paper-based documents, as well as a new credit assessment category to steer a higher percentage of applications through the automated approval process (thereby reducing the number of decline cases).

Bob runs a simulation of the TO-BE process and compares the data from the AS-IS and TO-BE business processes. He sees dramatic improvements and identifies these as new business opportunities. Anticipating that these new business opportunities will be acted upon, Bob makes the modeling project and artifacts available for subsequent consumption.

Since generating declines results in the most cost for this business process, Bob will create a Key Performance Measure which will be used to track what percentage of requests are declined and ensure the rate of declines meets the business objectives.

- Review the process enhancement request work item
- Review the business goals and objectives that drives the process enhancement request
- Model the AS-IS process model and discovers service automation opportunities
- Create the TO-BE process model and ascertains the impact of these service automation opportunities
- Define Key Performance Indicators and other Business Measures.
- Creates new business opportunities
- Indicate the process enhancement work item is complete

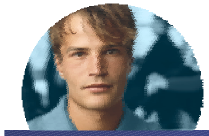
Model Business Processes to realize goals and objectives


WebSphere Business Modeler
Create, Simulate & Analyze As-Is Business Model
Create, Simulate, Analyze and Optimize To-Be Business Model
Create Financial Reports & ROI Estimates
Business Analyst

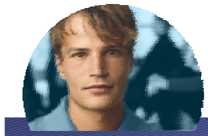
- Formally describe current business
- Discover and design key business processes
- Capture business data items exchanged between processes & tasks
- Assign tasks to roles that are responsible for their performance
- Determine and allocate required resources
- Model the business organization & roles organizational units can play
- Determination of any other process/tasks (services) that must be provided by others
- Verify business operational requirements

Model the business Operational Requirements

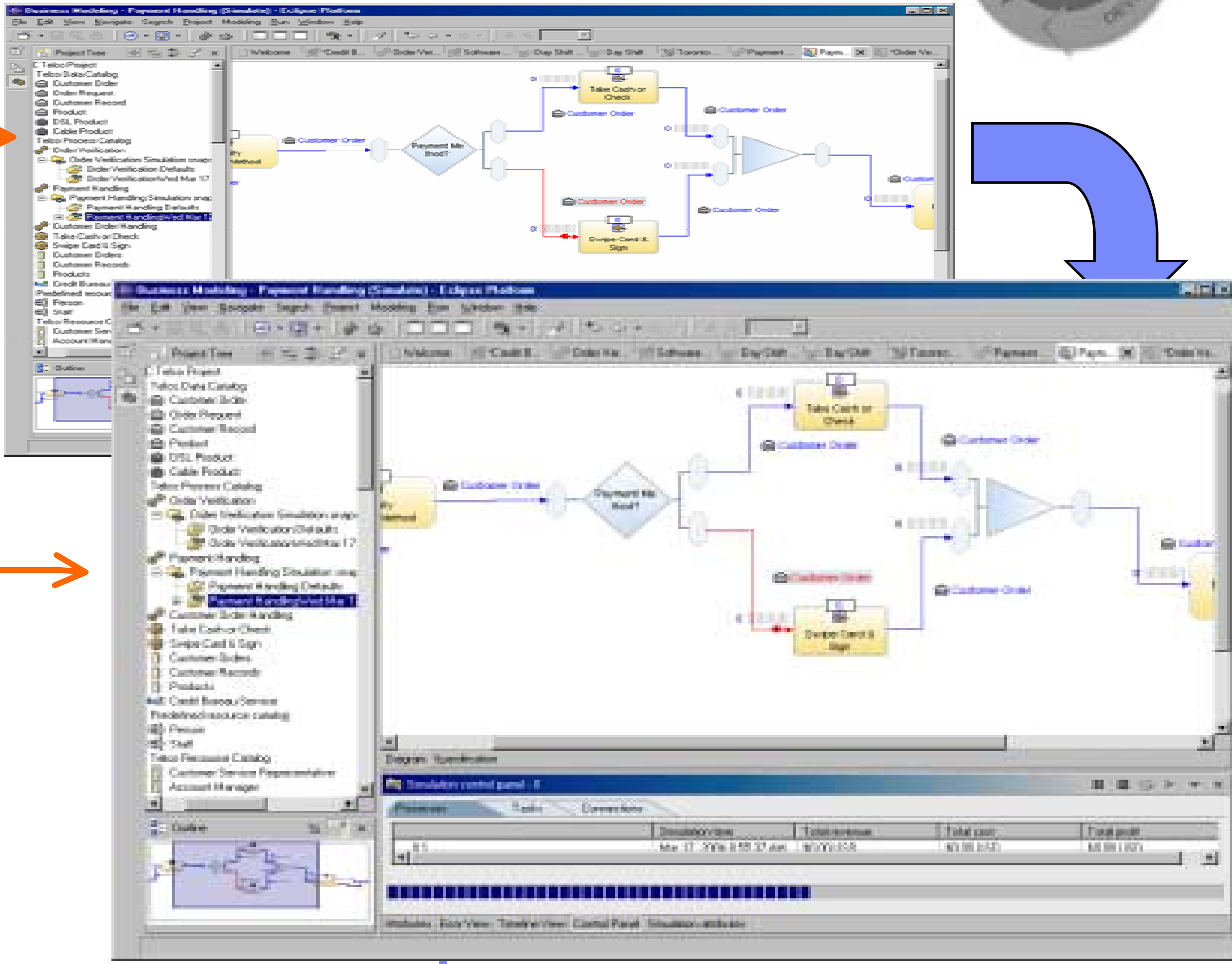
Business processes that realize goals and objectives



Analyst models "as is" business process and explores alternative "to be" business processes



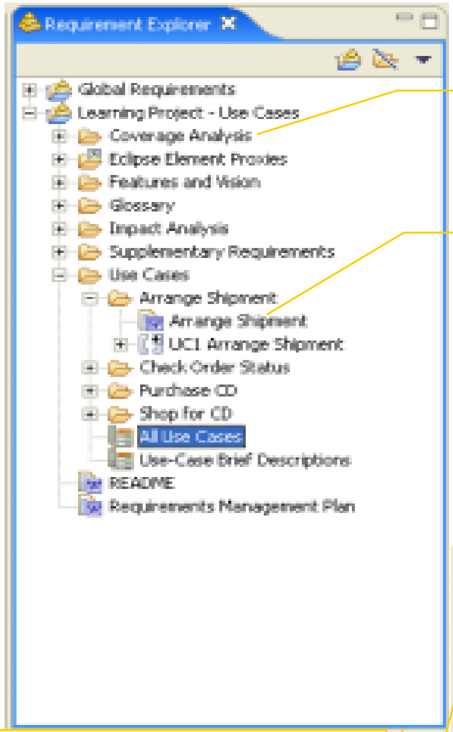
Verify "to be" business processes through simulation



IBM WebSphere Business Modeler

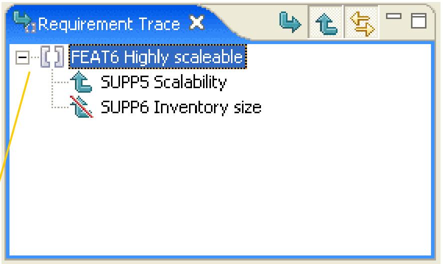
Requirements Management

- Create Business Vision Documents
- Create Business Use Case Specifications
- Define/Document Business Rule, Business Goal Requirements
- Define detailed system requirements (use cases and supplementary requirements)
- Trace enterprise requirements to business processes and service implementations



Requirements Explorer for viewing requirements

Create requirements and documents

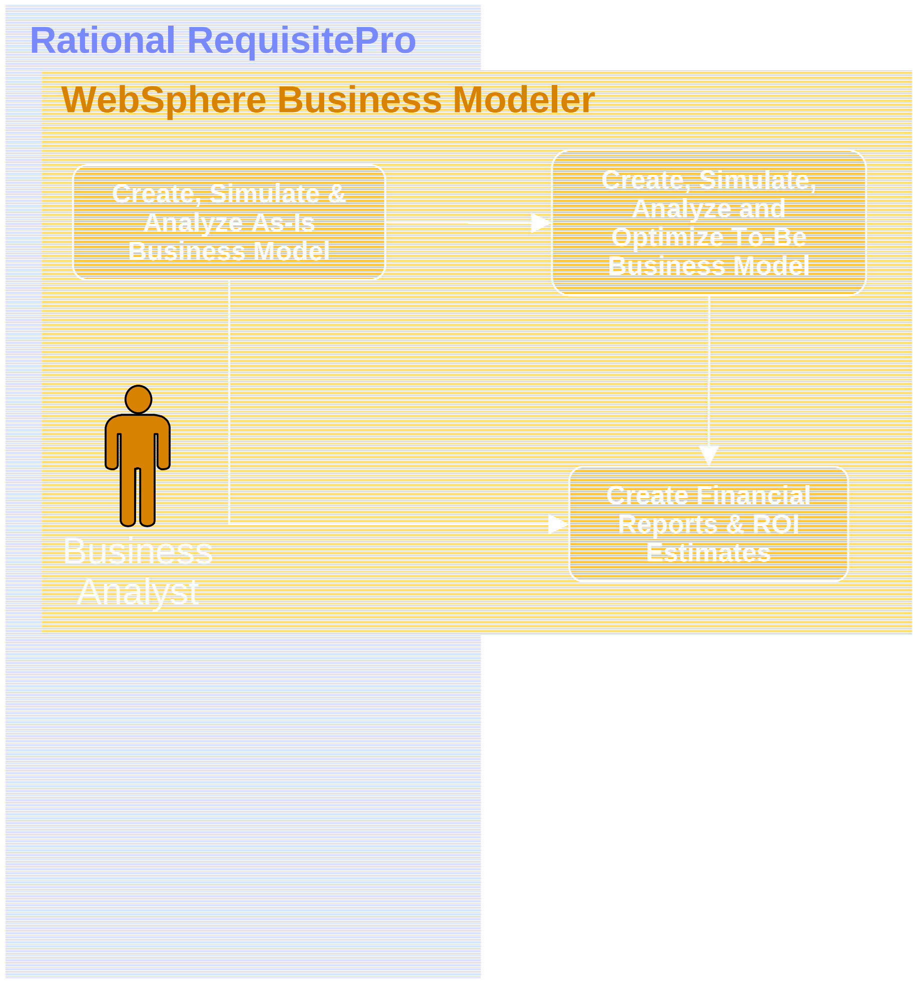


View requirements traceability from the perspective of either "trace-to" or "trace-from"

Customer Benefit:

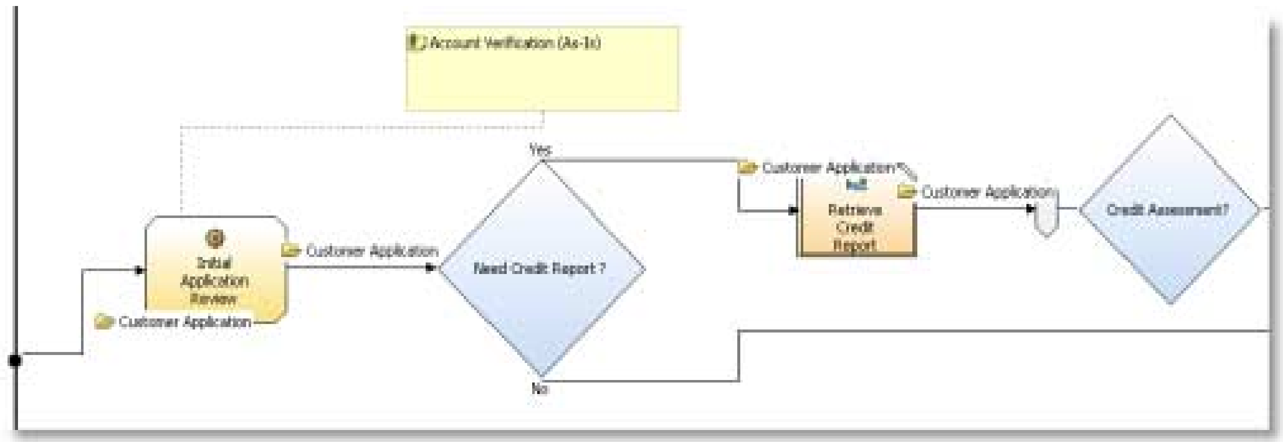
- Document and Capture Business Requirements
- Capture traceability relationships between elements in the application

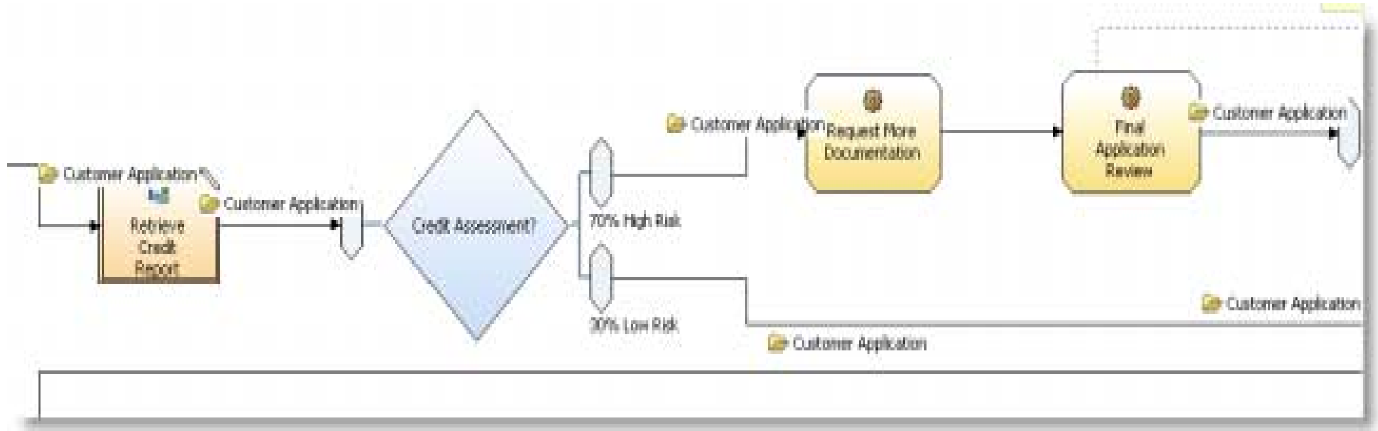
Analyze Business Processes to realize goals and objectives


Rational RequisitePro
WebSphere Business Modeler
Create, Simulate & Analyze As-Is Business Model
Create, Simulate, Analyze and Optimize To-Be Business Model
Create Financial Reports & ROI Estimates
Business Analyst

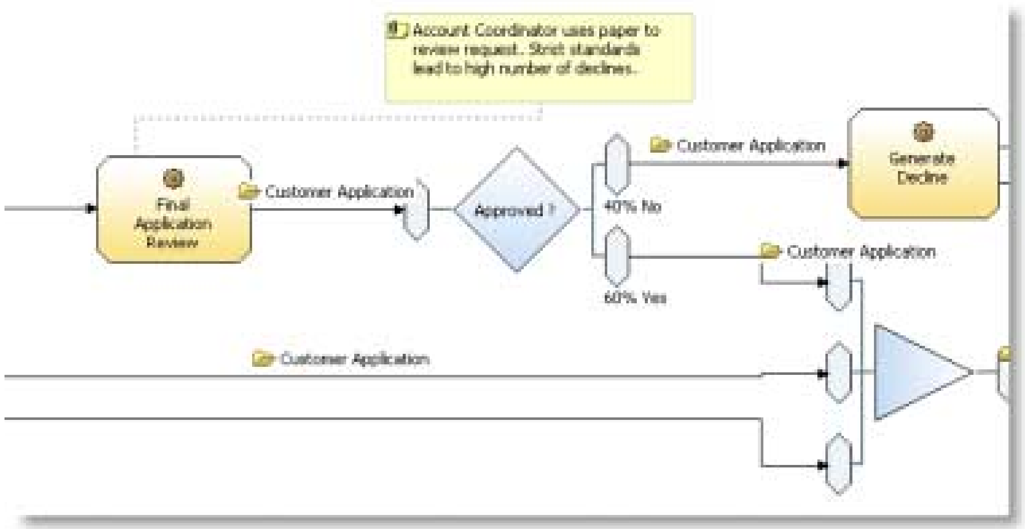
- Discover and design key business processes
- Capture business data items exchanged between processes & tasks
- Assign tasks to roles that are responsible for their performance
- Determine and allocate required resources
- Model the business organization & roles organizational units can play
- Assign resource costs, location, and availability
- Determination of any other process or tasks (services) that must be provided by others

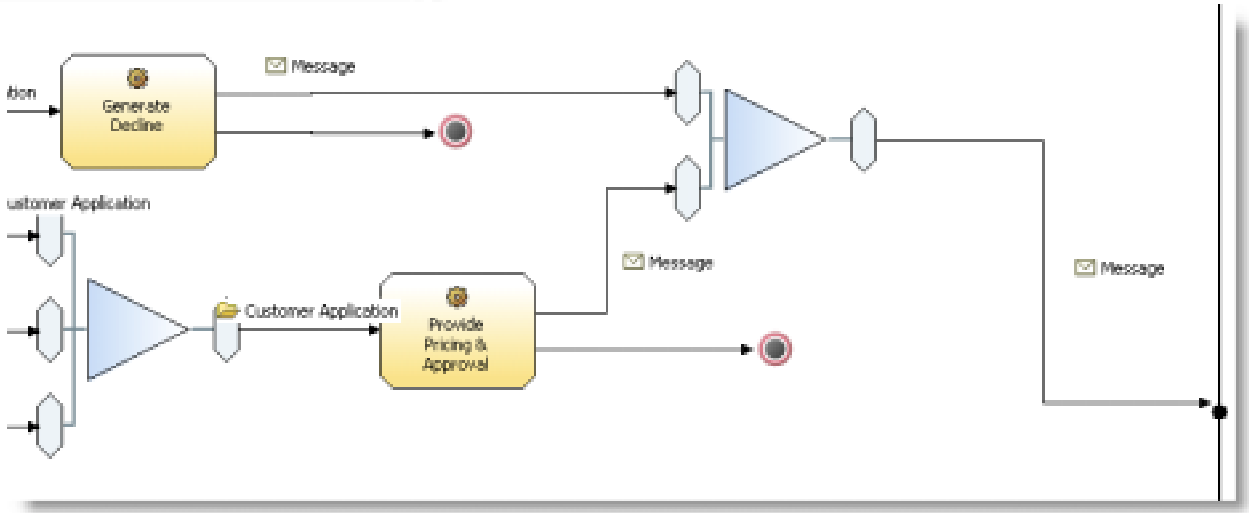
Examine existing scenarios for as-is business processes



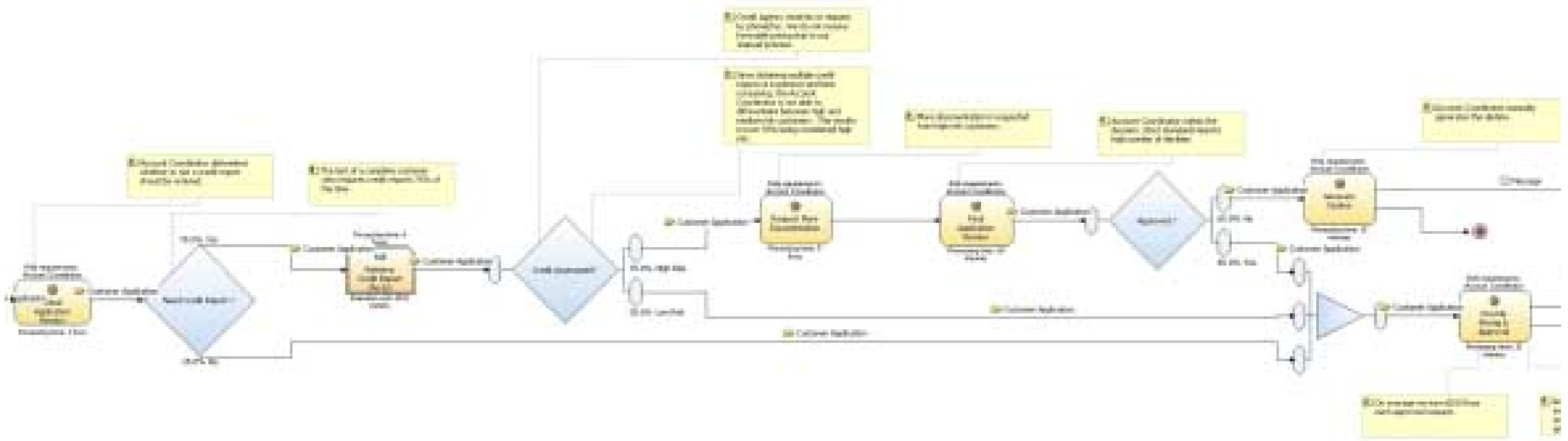


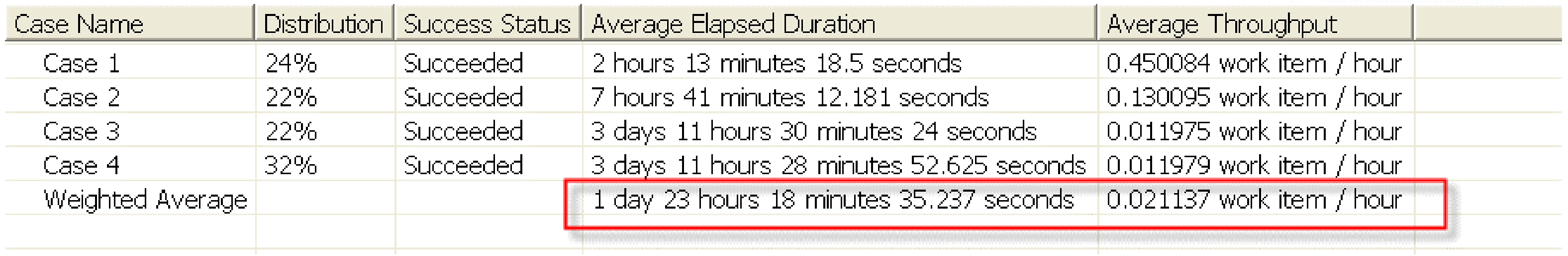
Account Verification (As-Is) Business Process (cont)



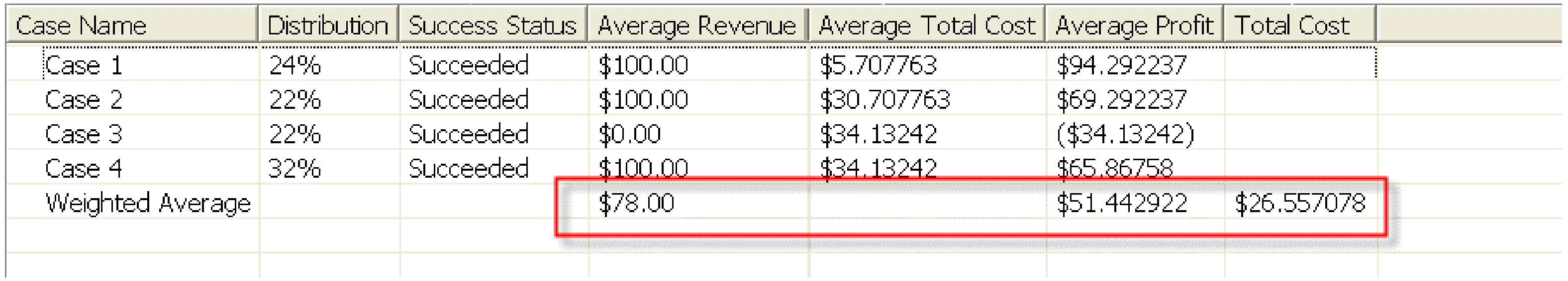


Simulate as-is Business Process





Case Name	Distribution	Success Status	Average Elapsed Duration	Average Throughput
Case 1	24%	Succeeded	2 hours 13 minutes 18.5 seconds	0.450084 work item / hour
Case 2	22%	Succeeded	7 hours 41 minutes 12.181 seconds	0.130095 work item / hour
Case 3	22%	Succeeded	3 days 11 hours 30 minutes 24 seconds	0.011975 work item / hour
Case 4	32%	Succeeded	3 days 11 hours 28 minutes 52.625 seconds	0.011979 work item / hour
Weighted Average			1 day 23 hours 18 minutes 35.237 seconds	0.021137 work item / hour

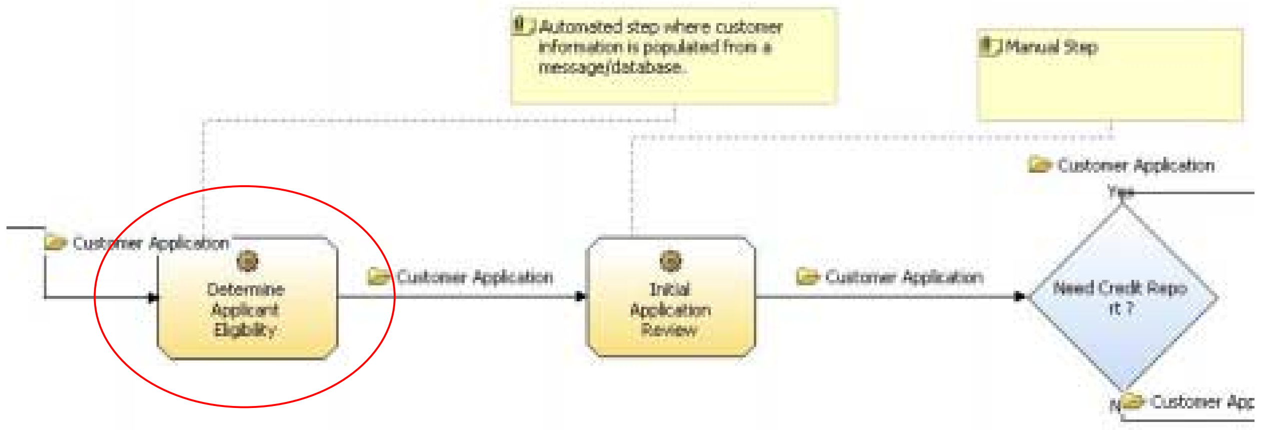


Case Name	Distribution	Success Status	Average Revenue	Average Total Cost	Average Profit	Total Cost
Case 1	24%	Succeeded	$100.00	$5.707763	$94.292237	
Case 2	22%	Succeeded	$100.00	$30.707763	$69.292237	
Case 3	22%	Succeeded	$0.00	$34.13242	($34.13242)	
Case 4	32%	Succeeded	$100.00	$34.13242	$65.86758	
Weighted Average			$78.00		$51.442922	$26.557078

Opportunities for Improvement

- Lack of single customer view and business rules for credit process result in our ordering more credit reports than we need
- Manual process for retrieving credit reports is highly inconsistent, expensive and time consuming
- We decline too many customer requests, resulting in unhappy potential customers and dissatisfied Sales representatives
- Although the rules for pricing and account are fairly straightforward the values change frequently. Since the procedure is manual it is difficult to implement the changes quickly
- Average Elapsed Duration: 1 day 23 minutes
- Weighted Average Profit: $51.44

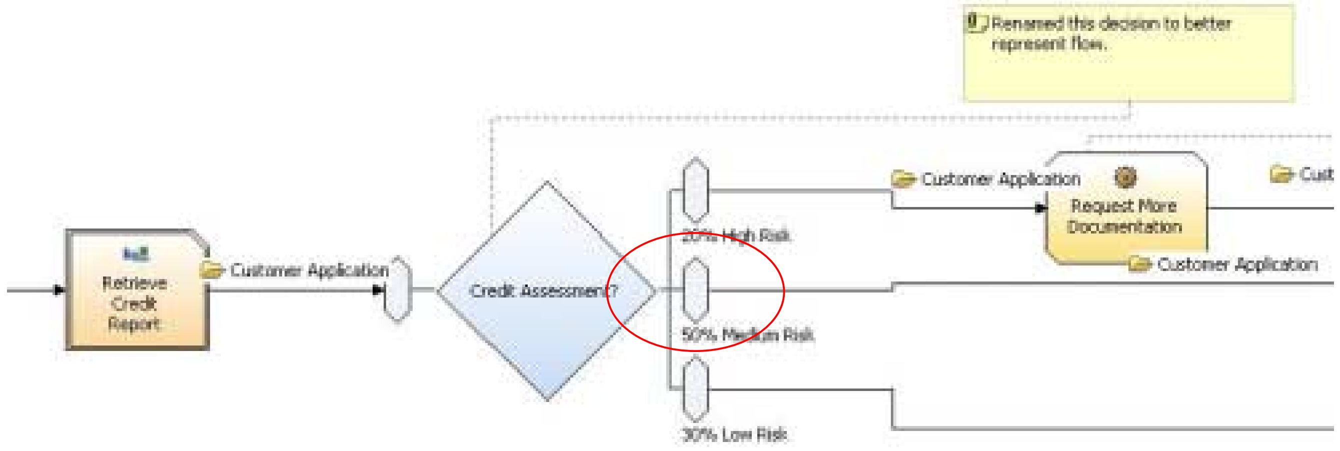
Re-engineer business processes to realize goals


Automated step where customer information is populated from a message/database.
Manual Step
Customer Application
Yes
Customer Application
Determine Applicant Eligibility
Customer Application
Initial Application Review
Customer Application
Need Credit Report ?
No
Customer App

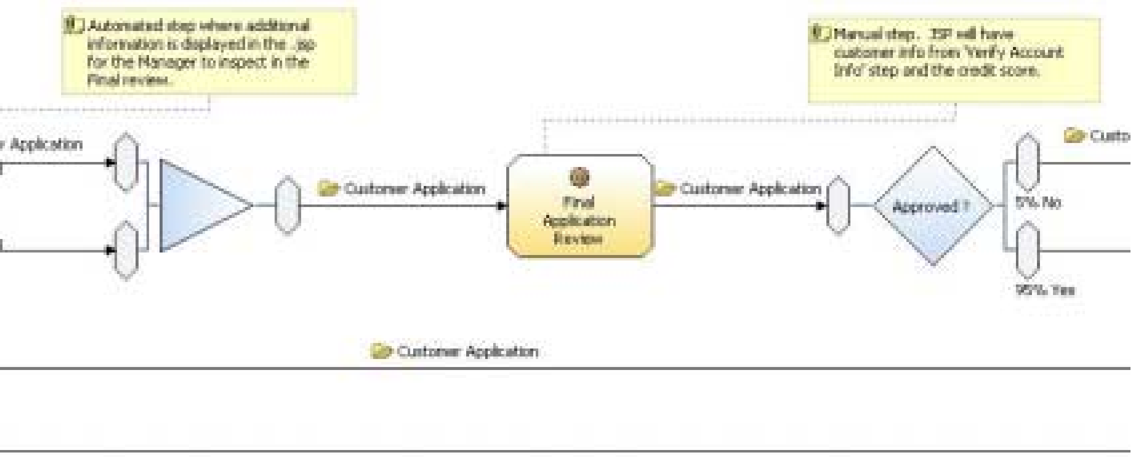
Shorter Process Duration

More Automation

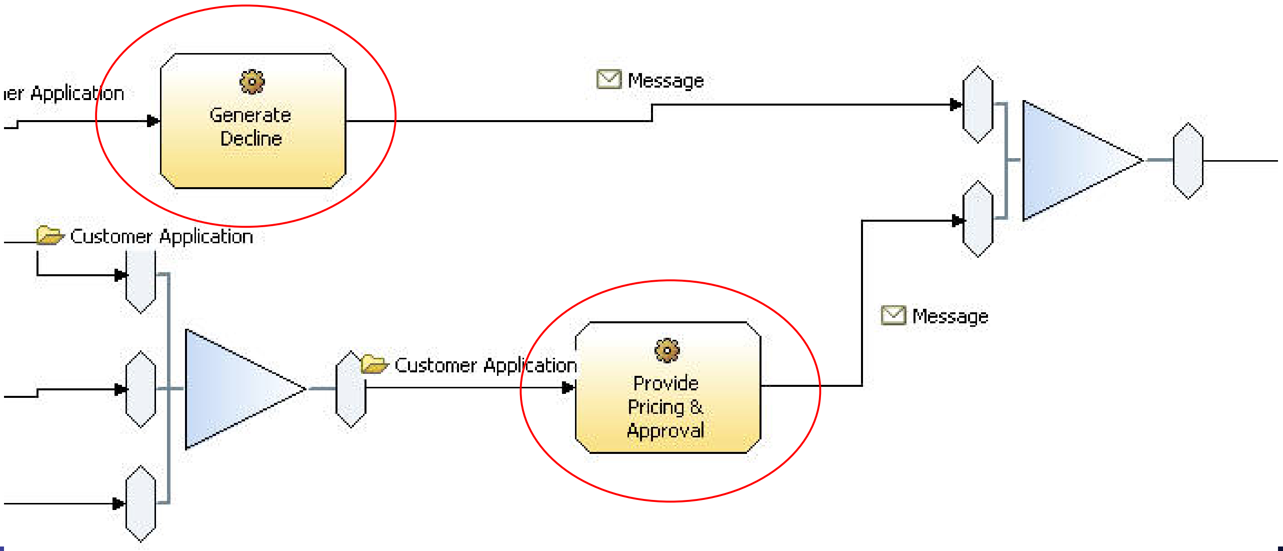
Lower Process Cost



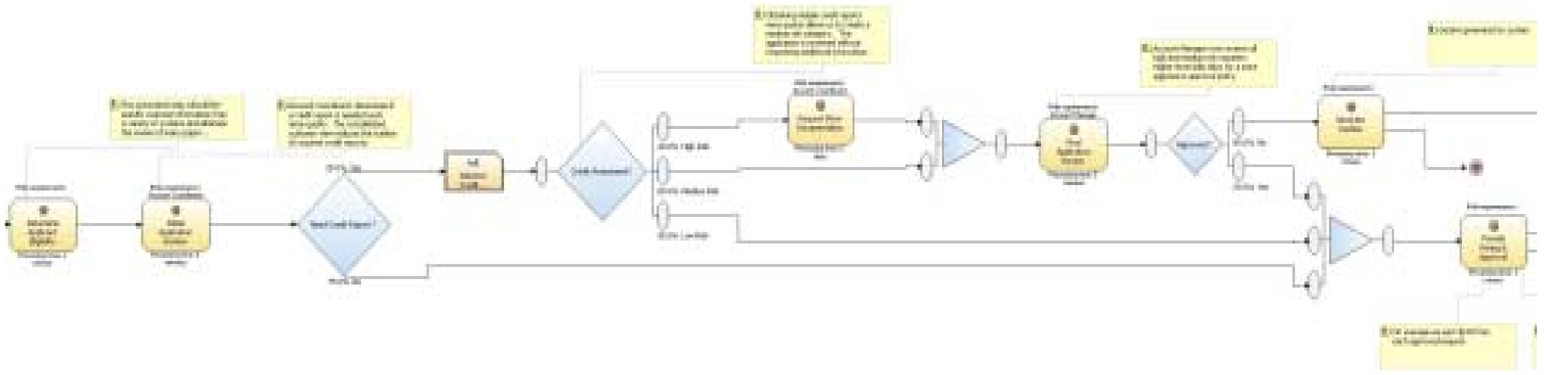
Account Verification (To-Be) Business Process (cont)

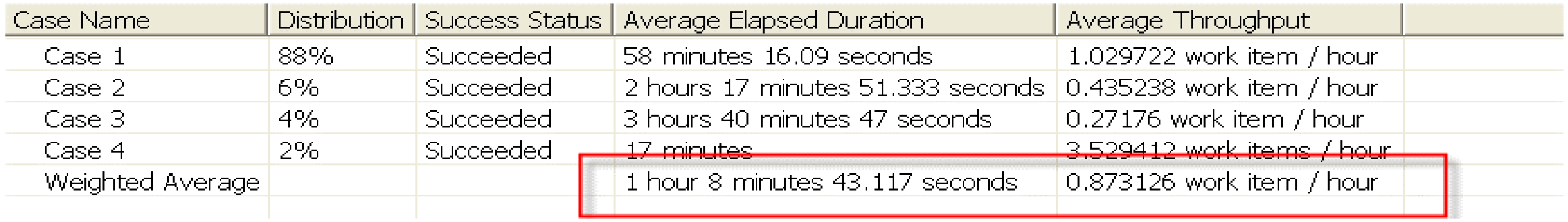

Automated step where additional information is displayed in the .jsp for the Manager to inspect in the Final review.
Manual step. ISP will have customer info from 'Verify Account Info' step and the credit score.
r Application
Customer Application
Final Application Review
Customer Application
Approved ?
Custo
5% No
95% Yes
Customer Application

These are the business operational requirements

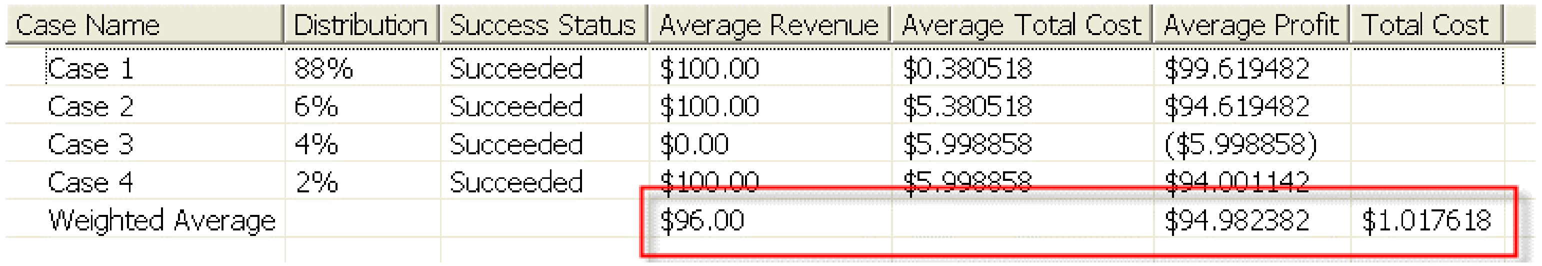


Refine and simulate to-be Business Processes



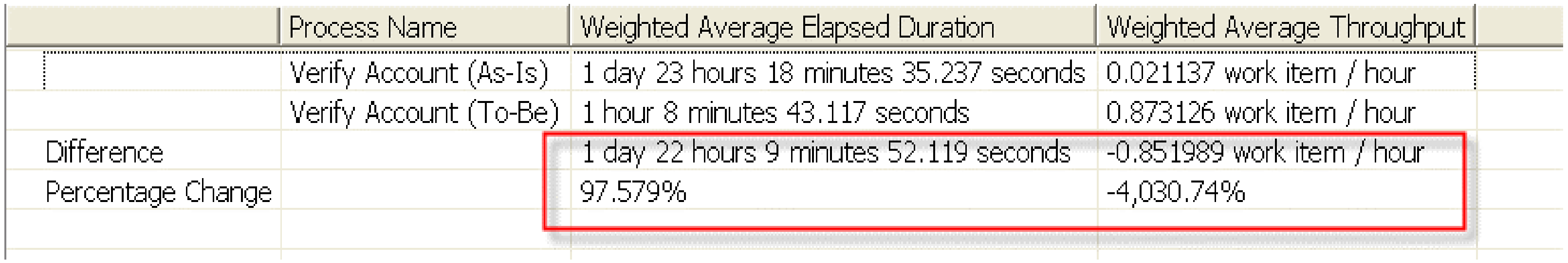


Case Name	Distribution	Success Status	Average Elapsed Duration	Average Throughput
Case 1	88%	Succeeded	58 minutes 16.09 seconds	1.029722 work item / hour
Case 2	6%	Succeeded	2 hours 17 minutes 51.333 seconds	0.435238 work item / hour
Case 3	4%	Succeeded	3 hours 40 minutes 47 seconds	0.27176 work item / hour
Case 4	2%	Succeeded	17 minutes	3.529412 work items / hour
Weighted Average			1 hour 8 minutes 43.117 seconds	0.873126 work item / hour

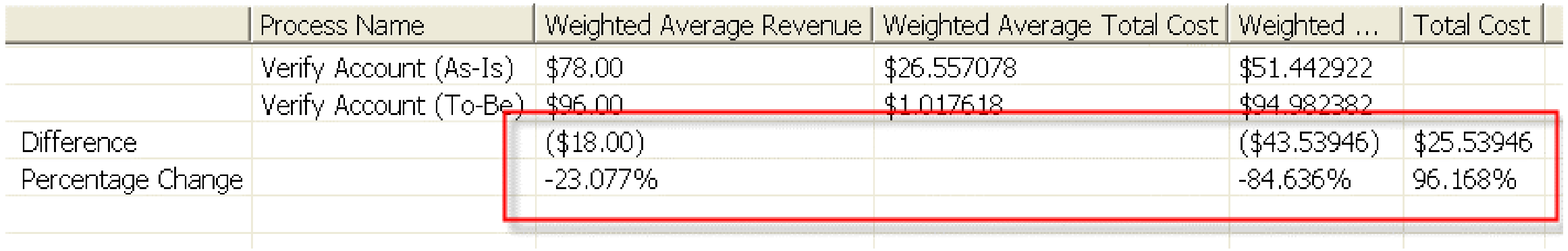


Case Name	Distribution	Success Status	Average Revenue	Average Total Cost	Average Profit	Total Cost
Case 1	88%	Succeeded	$100.00	$0.380518	$99.619482	
Case 2	6%	Succeeded	$100.00	$5.380518	$94.619482	
Case 3	4%	Succeeded	$0.00	$5.998858	($5.998858)	
Case 4	2%	Succeeded	$100.00	$5.998858	$94.001142	
Weighted Average			$96.00		$94.982382	$1.017618

Compare results and see if objects are met



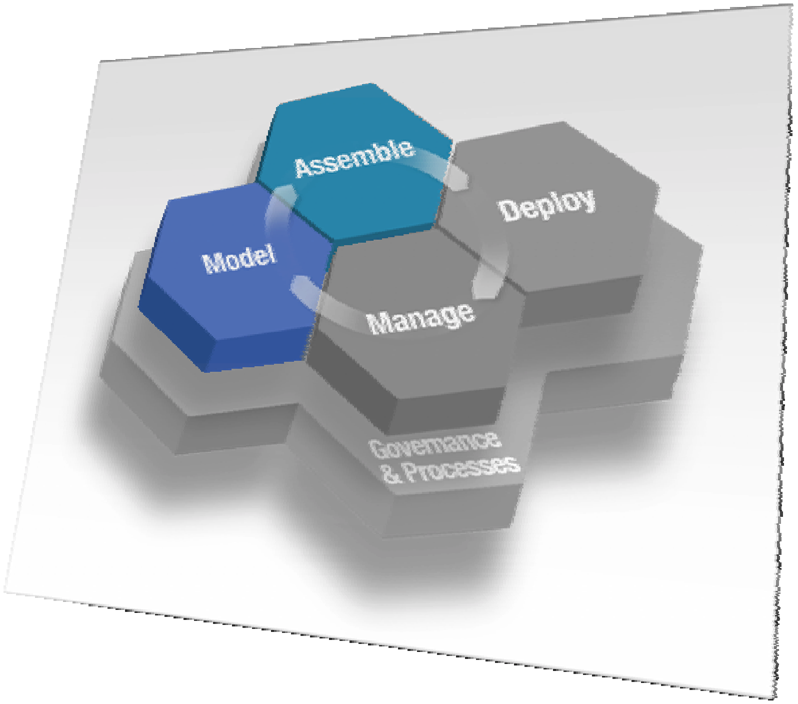
	Process Name	Weighted Average Elapsed Duration	Weighted Average Throughput
	Verify Account (As-Is)	1 day 23 hours 18 minutes 35.237 seconds	0.021137 work item / hour
	Verify Account (To-Be)	1 hour 8 minutes 43.117 seconds	0.873126 work item / hour
Difference		1 day 22 hours 9 minutes 52.119 seconds	-0.851989 work item / hour
Percentage Change		97.579%	-4,030.74%



	Process Name	Weighted Average Revenue	Weighted Average Total Cost	Weighted ...	Total Cost
	Verify Account (As-Is)	$78.00	$26.557078	$51.442922	
	Verify Account (To-Be)	$96.00	$1.017618	$94.982382	
Difference		($18.00)		($43.53946)	$25.53946
Percentage Change		-23.077%		-84.636%	96.168%

Agenda

- Business Driven Development
  - A development process for deriving solutions from business objectives
- Software Development Platform for BDD and SOA
- Capture Business Goals and Objectives
- Analyze Business Processes to realize Business Goals
- Architect a Services Solution
- Assemble, Deploy & Test
- Summary



Architect a Services Solution

Business Analyst Bob modeled just enough detail to capture and verify business operational requirements necessary to realize the business objectives. This initial model is concerned only with the business and often has insufficient detailed for an IT solution.

Once the business requirements are known, Architect Al can elaborates the business processes to ensure the business requirements are sufficiently specified that services design can be created. Al is now ready to begin architecting a services solution to realize the business requirements in a manner that addresses the IT concerns. Al has chosen SOA as an architectural style for developing a solution that he thinks will meet the business and IT functional and non-functional requirements. He designs the services components, their provided and required interfaces, the service data exchanged, and develops activity models that specify the implementation details of some of the key services. In the process, Al has also developed a domain model that captures domain knowledge necessary to design the service implementations. Al uses a set of transformations to generate the initial solution implementation based on the chosen architectural patterns. This initial implementation is handed off to the developer to be completed, deployed, and tested.

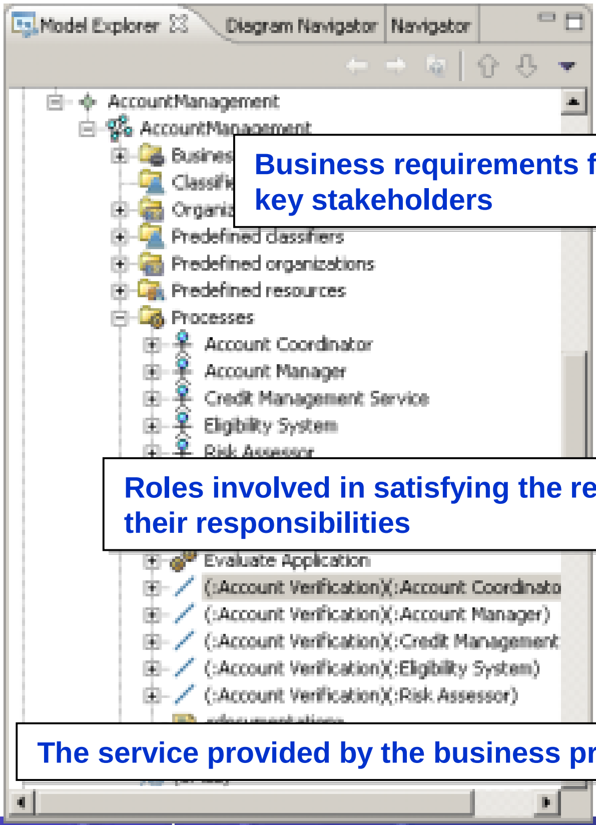
Al must look across a number of possibly complex processes with overlapping roles that may need to be significantly refactored for an effective, maintainable services solution. Al must also deal with IT concerns such as availability, distribution, integrity, security and persistence as well as the nonfunctional characteristics of an acceptable solution. Al may:

Some things an Architect might to:

- Partition the solution by business functional areas in order to increase cohesion and reduce coupling
- Fulfill a number of related contracts with a single component.
- Realize a complex process with a number of choreographed components
- Use strict vs. loose contract fulfillment
- Create components that provide services derived from refactoring and decoupling tasks in processes.
- Break up single tasks into multiple tasks
- Combine sequences of tasks into a new service operation.
- Refactor interfaces by either combining highly related operations or separating out unrelated services into separate interfaces
- Refactor services provided by roles that cross process boundaries.
- Use a state machine to model the realization of a business process.
- Refactor to deal with commonality and variability to support better reuse. Introduce templates and template instantiations for commonly recurring business services that need to be customizable.

Explore Business Requirements

The business analyst Bob focused on the sequences of tasks required to meet a business objectives. Bob finds a process model the best way to document these steps. Al wants to take a more services view of the business requirements in anticipation of using a service oriented solution. Al examines Bob's business processes by exploring them through a business services model view that organizes the processes and tasks by the roles that perform them. Al examines each of the collaborations representing the business processes in order to begin laying out a set of components that realize the business functional and nonfunctional requirements. At this point, Al may discover missing information that is easier to see from the business services view than it was from the business process view of the business requirements.



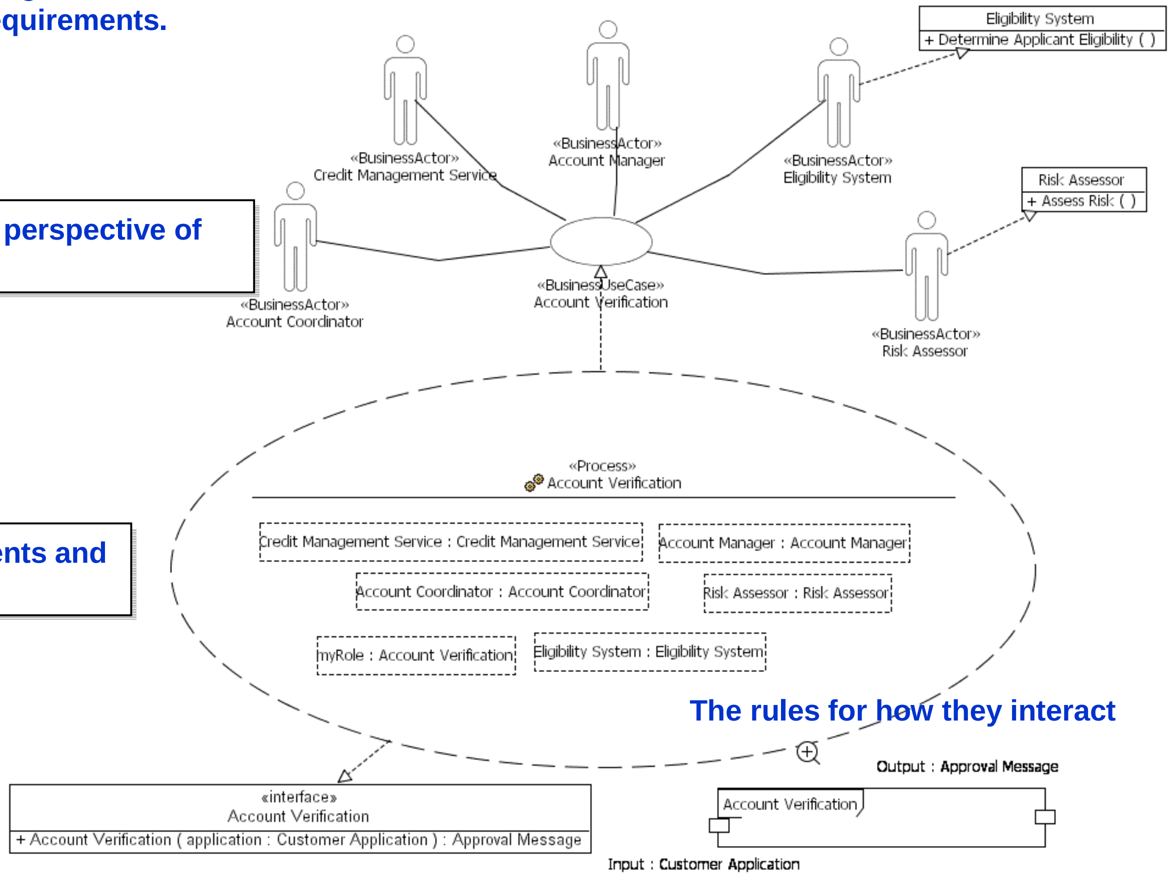
Model Explorer | Diagram Navigator | Navigator

- AccountManagement
  - AccountManagement
    - Busines
    - Classifie
    - Organiz
    - Predefined classifiers
    - Predefined organizations
    - Predefined resources
    - Processes
      - Account Coordinator
      - Account Manager
      - Credit Management Service
      - Eligibility System
      - Risk Assessor
      - Evaluate Application
      - (:Account Verification)(:Account Coordinato
      - (:Account Verification)(:Account Manager)
      - (:Account Verification)(:Credit Management
      - (:Account Verification)(:Eligibility System)
      - (:Account Verification)(:Risk Assessor)

Business requirements from the perspective of key stakeholders

Roles involved in satisfying the requirements and their responsibilities

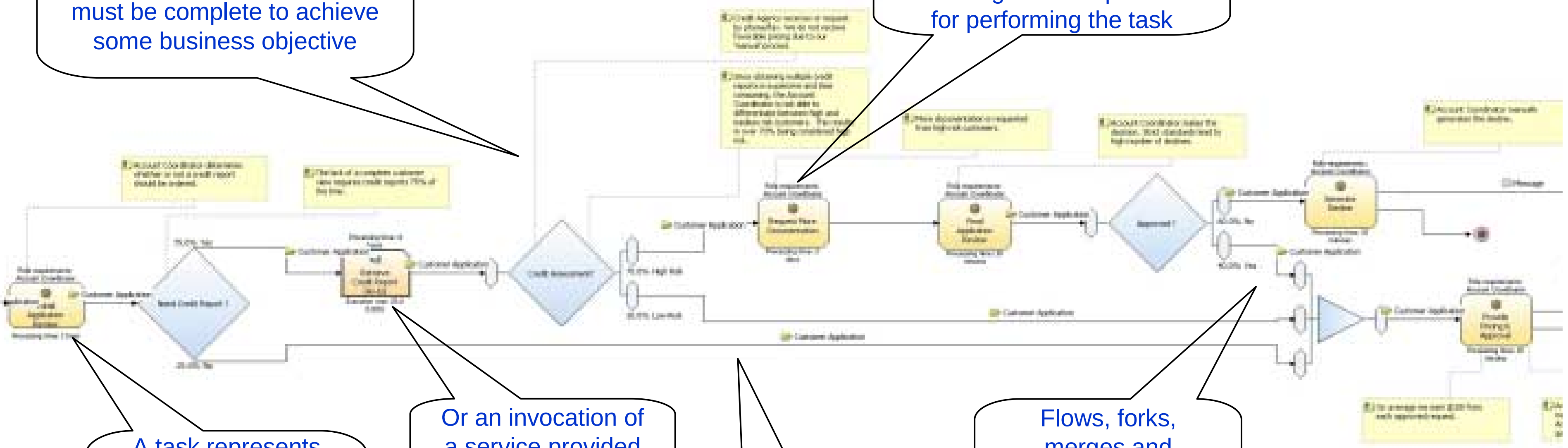
The service provided by the business process



Eligibility System
+ Determine Applicant Eligibility ( )
«BusinessActor» Credit Management Service
«BusinessActor» Account Manager
«BusinessActor» Eligibility System
Risk Assessor
+ Assess Risk ( )
«BusinessActor» Account Coordinator
«BusinessUseCase» Account Verification
«BusinessActor» Risk Assessor
«Process» Account Verification
Credit Management Service : Credit Management Service
Account Manager : Account Manager
Account Coordinator : Account Coordinator
Risk Assessor : Risk Assessor
myRole : Account Verification
Eligibility System : Eligibility System
Output : Approval Message
«interface» Account Verification
+ Account Verification (application : Customer Application) : Approval Message
Account Verification
Input : Customer Application

The rules for how they interact

A closer look at business processes



A business process specifies a sequence of tasks that must be complete to achieve some business objective

Tasks can require resources for completion including roles responsible for performing the task

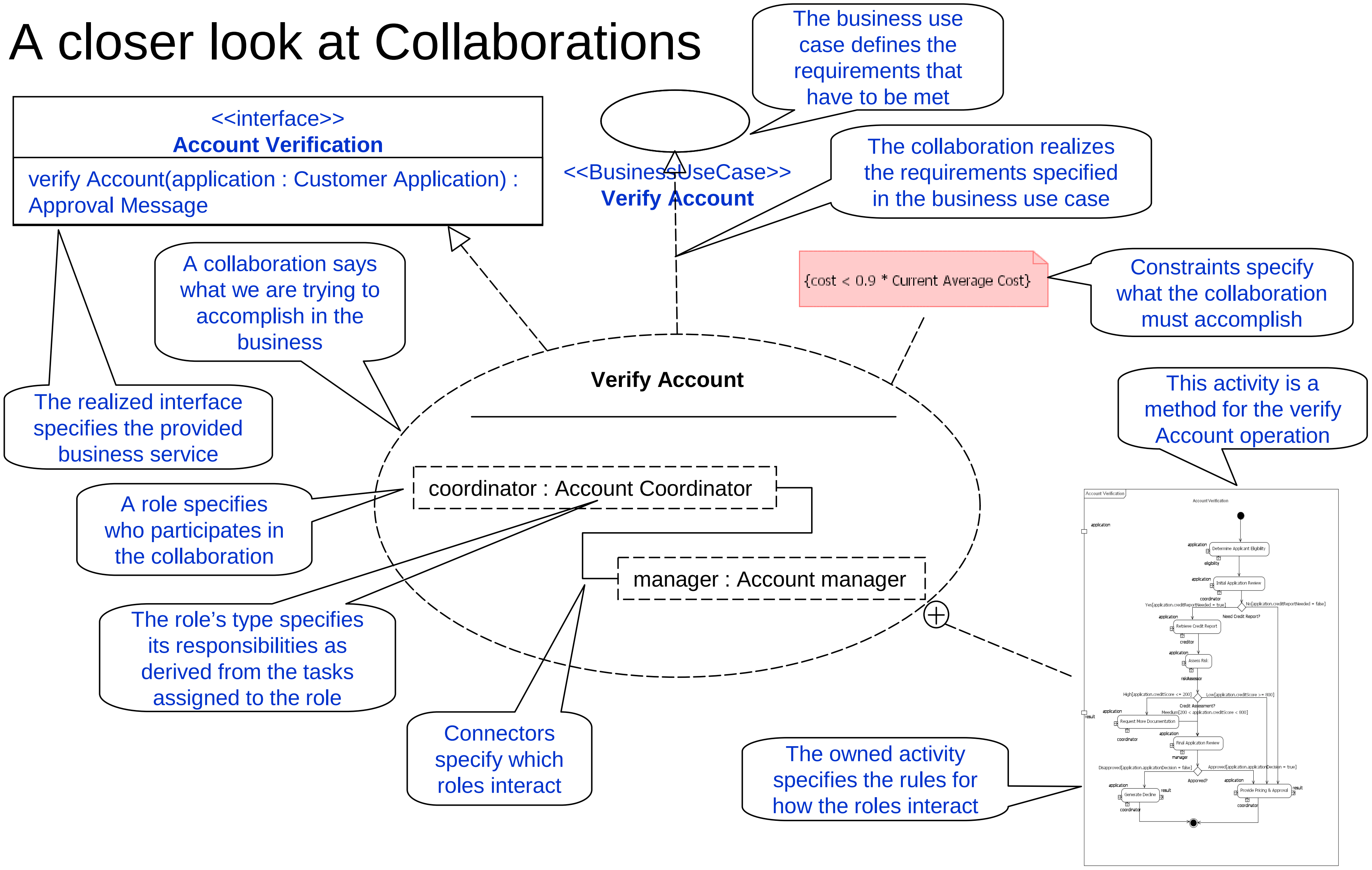
A task represents something that is done in a business process. It can be an atomic action

Or an invocation of a service provided by others

Or the invocation of another process

Flows, forks, merges and decisions control task sequencing

A closer look at Collaborations



<<interface>>
Account Verification
verify Account(application : Customer Application) : Approval Message
<<BusinessUseCase>>
Verify Account
{cost < 0.9 * Current Average Cost}
Verify Account
coordinator : Account Coordinator
manager : Account manager
Account Verification
Account Verification
application
Determine Applicant Eligibility
eligbility
Initial Application Review
coordinator
Yes[application.creditReportNeeded = true]
No[application.creditReportNeeded = false]
Need Credit Report?
Retrieve Credit Report
creditor
Assess Risk
riskAssessor
High[application.creditScore <= 200]
Low[application.creditScore >= 800]
Credit Assessment?
Meedium[200 < application.creditScore < 800]
result
Request More Documentation
Final Application Review
manager
Disapproved[application.applicationDecision = false]
Approved[application.applicationDecision = true]
Apporved?
Generate Decline
Provide Pricing & Approval

The business use case defines the requirements that have to be met

The collaboration realizes the requirements specified in the business use case

Constraints specify what the collaboration must accomplish

A collaboration says what we are trying to accomplish in the business

The realized interface specifies the provided business service

This activity is a method for the verify Account operation

A role specifies who participates in the collaboration

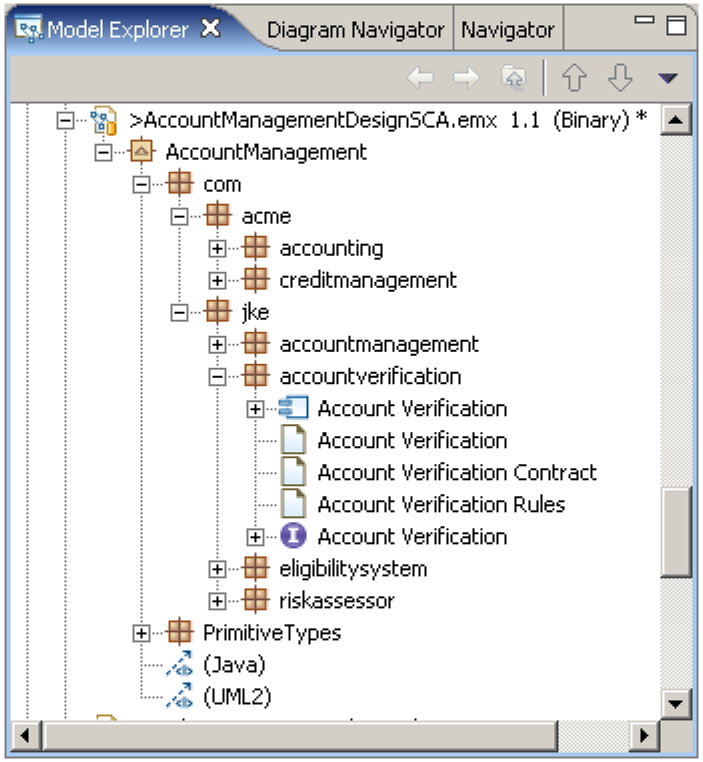
The role's type specifies its responsibilities as derived from the tasks assigned to the role

Connectors specify which roles interact

The owned activity specifies the rules for how the roles interact

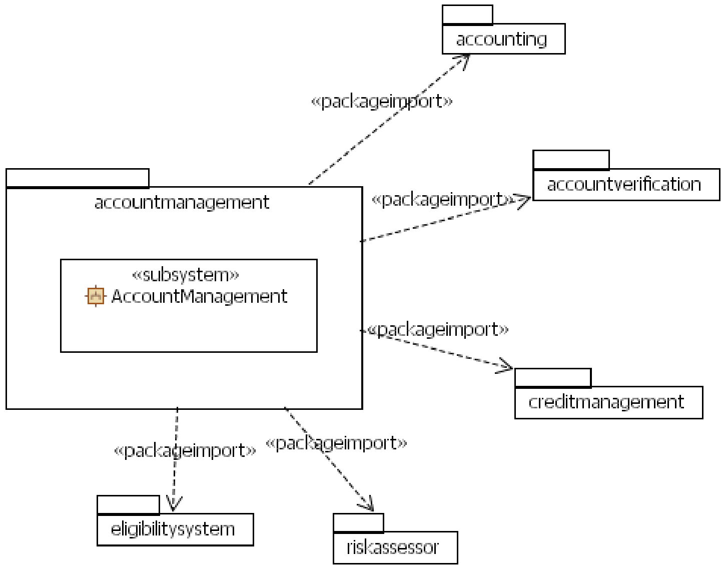
Select the Service Realization Architecture

AI decides that to meet the nonfunctional and IT requirements captured so far in the business services model, a SOA would be the best architectural style to use. He uses the AccountManagementDesign model he created to view the business services contracts to capture the design of the services components used in the solution. AI indicates a SOA architecture will be used by applying the Software Services Profile to the AccountManagementDesign model. This extends the model to support SOA design. AI may also instantiate a template model that creates the intial package structures and empty diagrams that JK Enterperises has standardized on for services development. Such a template model might contain standard packages for the functional areas in JK Enterprises and contain shortcuts to other commonly used models. AI instantiates the templates for the new business functional area he is designing.



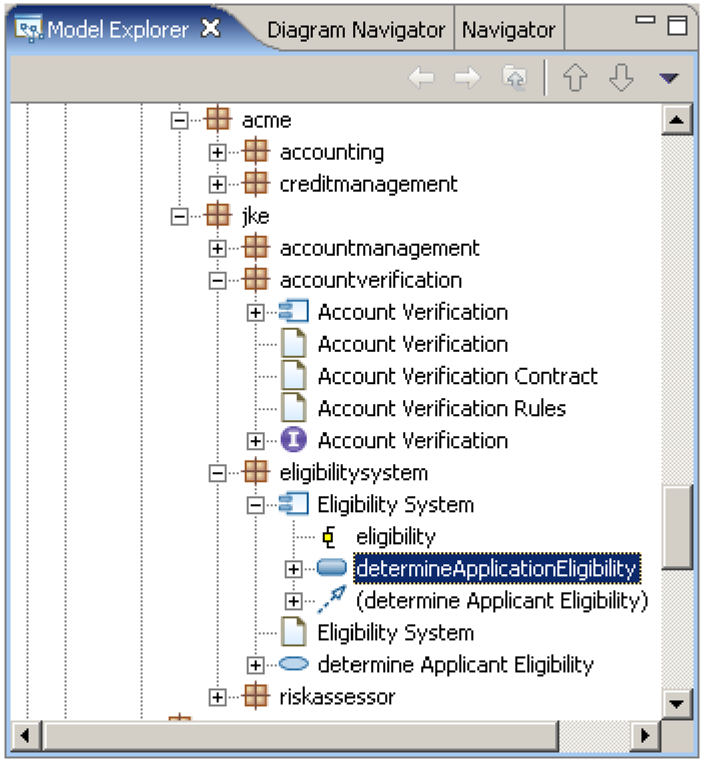
```
>AccountManagementDesignSCA.emx  1.1  (Binary) *
  AccountManagement
    com
      acme
        accounting
        creditmanagement
      jke
        accountmanagement
        accountverification
          Account Verification
          Account Verification
          Account Verification Contract
          Account Verification Rules
          Account Verification
        eligibilitysystem
        riskassessor
    PrimitiveTypes
    (Java)
    (UML2)
```

Start with separating concerns and packaging to manage complexity and facilitate reuse

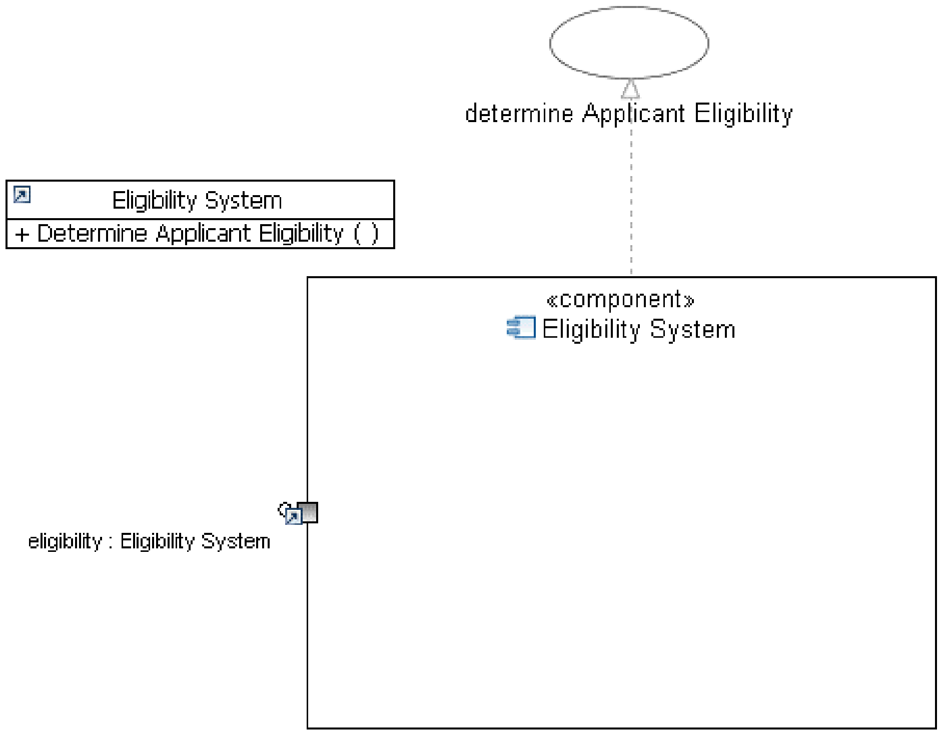


Create The EligibilitySystem Component

Al sees from the Account Verification business contract that he needs to provide components capable of playing a number of roles in order to fulfill the contract. Al has noticed that coordinating and managing accounts has a lot of variability so these required interfaces are obtained through ports that can be wired to different providers in different situations requiring account verification in JK Enterprise applications. However, the eligibility and risk assessment capabilities are internal to JK Enterprises and are stable. So Al decides he does not need to externalize these parts in order to allow them to be connected to different concrete service providers. Al also knows there's an existing component he can easily reuse for accessing risk.



```
acme
  accounting
  creditmanagement
jke
  accountmanagement
  accountverification
    Account Verification
    Account Verification
    Account Verification Contract
    Account Verification Rules
    Account Verification
  eligibilitysystem
    Eligibility System
      eligibility
      determineApplicationEligibility
      (determine Applicant Eligibility)
    Eligibility System
    determine Applicant Eligibility
  riskassessor
```



determine Applicant Eligibility
Eligibility System
+ Determine Applicant Eligibility ( )
«component»
Eligibility System
eligibility : Eligibility System

Develop the Business Domain Model

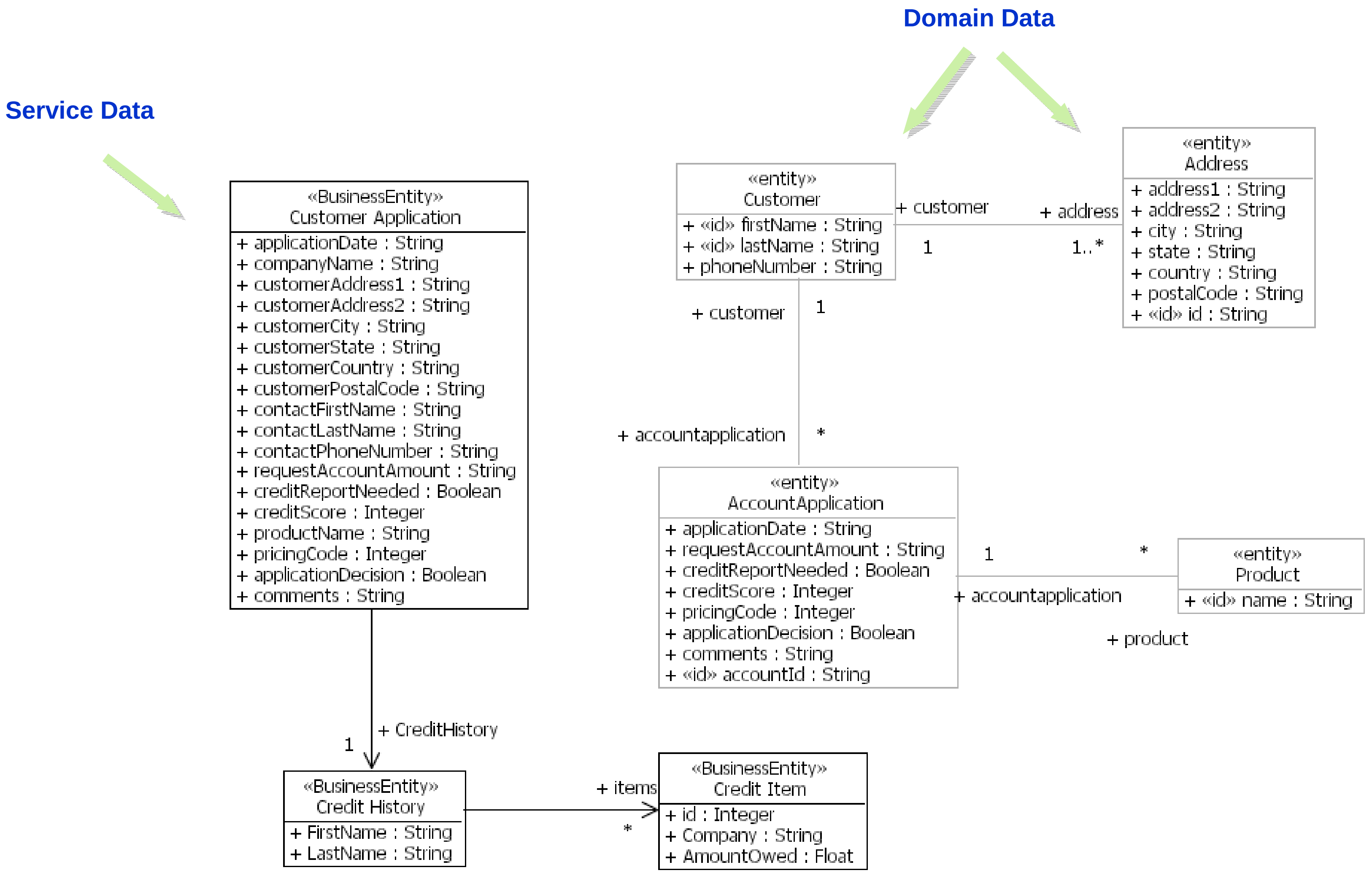
In the process of creating the activity realizing the provided services operations, Al realizes he need some persistent business domain information. He needs information about customers, projects, accounts, addresses, and credit history in order to verify the account request. Some of this information is already available in existing databases that can be directly referenced in the services model, other information can be mined and adapted from existing data stores, but can't be used directly. And finally, some information is new and Al models it directly.

Al opens a model browser in an existing JK Enterprises database to access account information. These databases are displayed as logically related classes in Al's view since he is modeling the services implementations at that level of abstraction. Al use the RDA tools to adapt the customer database to classes that represent the data he needs for customers and addresses. He then creates a new persistent entity class called Project to contain project data that is currently not maintained by any JK Enterprises database. This information can be used as data stores in the account verification activity as needed in order to complete the service implementation. The service implementation uses this domain data in order to update fields in the customer application service data.

The domain model not only describes the data structure, but may also provide operations and constraints that ensure data integrity and provide low-level business behavior that can be reused to implement many services. This business behavior is not expected to be exposed as services outside JK Enterprises, and because of its use in many services, it must be highly secure, accessed quickly, and have ensured integrity. Al decides as part of the system architecture how business capabilities are realized in order to meet all requirements.

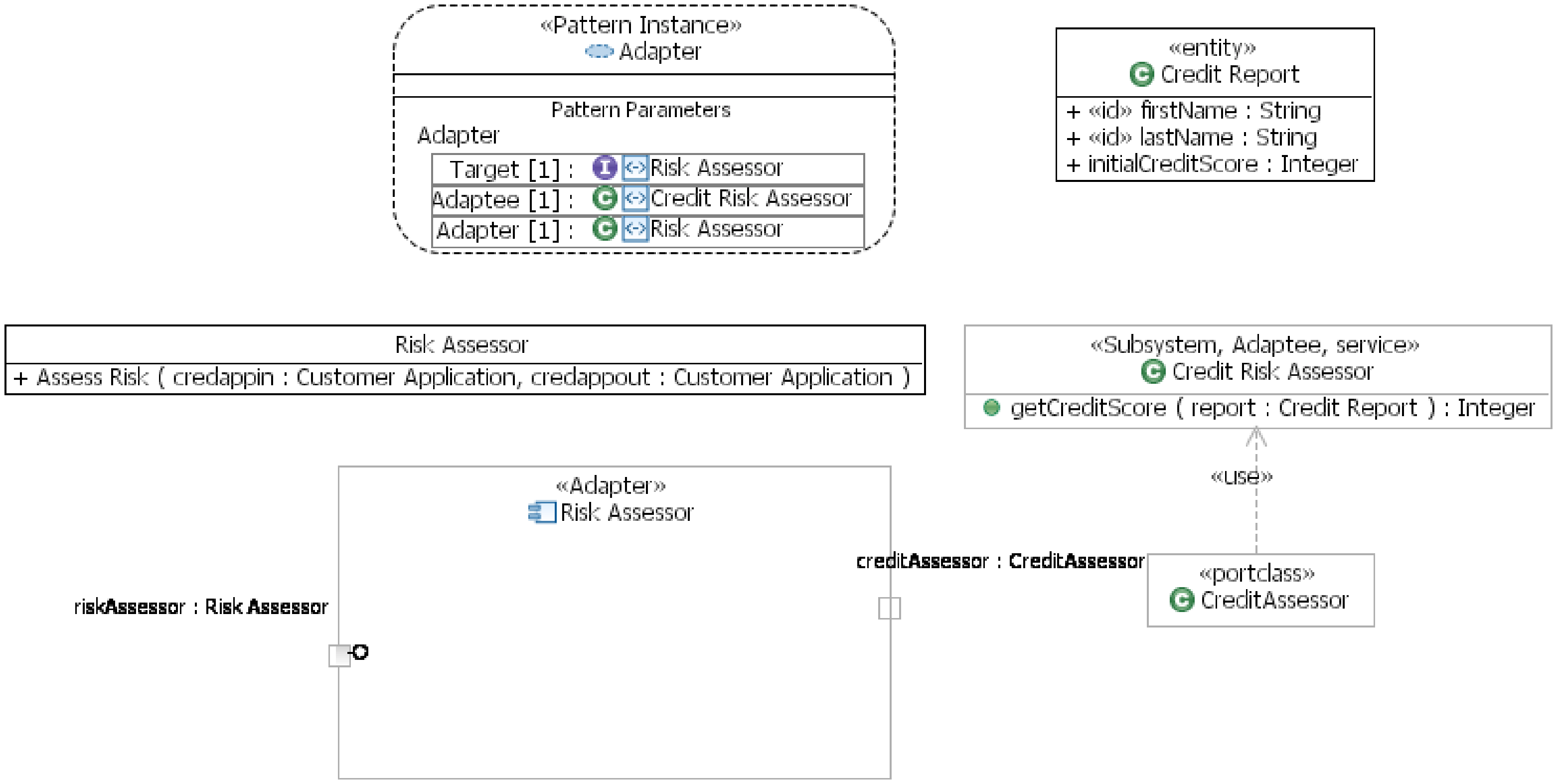
Al may decide later that some domain functions need to be exposed as reusable services. He can do this by selecting the domain class and operations and use the development tools to automatically create s component that exposes that business domain function as an external service.

JK Enterprises Domain Model



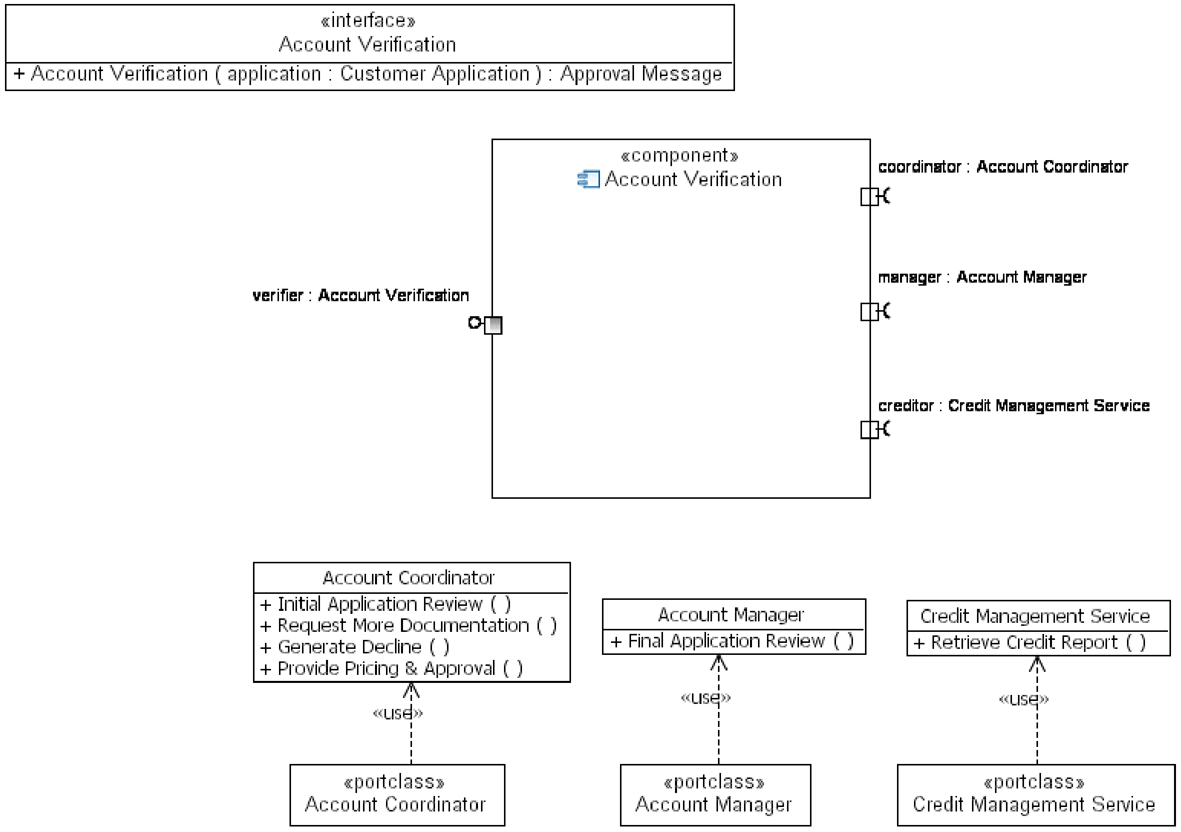
Find and reuse an existing asset

Al remembers something about a existing component that provides services for calculating credit score. He realizes this is all he really needs to implement the Access Risk task. But he decides to adapt the existing service provider to provide exactly what the business analyst wanted. This is because he's been told that there may be additional risk assessment requirements in the future, and he does not want introduce unnecessary maintenance problems by replacing the Assess Risk task with an existing service.


«Pattern Instance»
Adapter
Pattern Parameters
Adapter
Target [1] : Risk Assessor
Adaptee [1] : Credit Risk Assessor
Adapter [1] : Risk Assessor
«entity»
Credit Report
+ «id» firstName : String
+ «id» lastName : String
+ initialCreditScore : Integer
Risk Assessor
+ Assess Risk (credappin : Customer Application, credappout : Customer Application)
«Subsystem, Adaptee, service»
Credit Risk Assessor
getCreditScore (report : Credit Report) : Integer
«use»
«Adapter»
Risk Assessor
creditAssessor : CreditAssessor
«portclass»
CreditAssessor
riskAssessor : Risk Assessor

Create a component for Account Verification

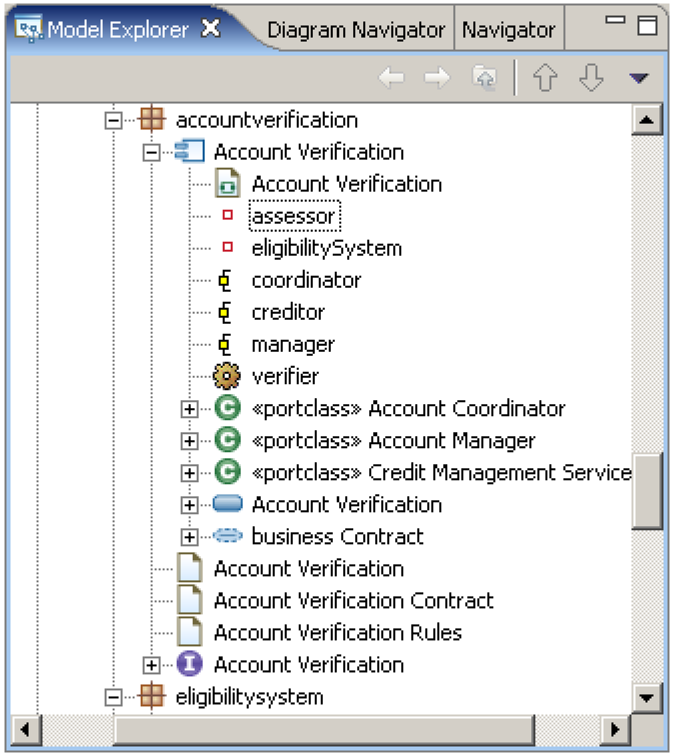
Next Al develops a SOA realization of the Account Verification business process. He creates an Account Verification component which has three ports for interfacing with clients, as well as the coordinator, manager, and creditor service providers. Al indicates the component provides the Account Verification service interface through his verifier port. This interface contains the Account Verification service operation which takes a Customer Application as its input and returns an Approval Message. Al also indicates that the coordinator port requires the Account Coordinator interface, the manager port requires the Account Manager interface, and the creditor port requires the Credit Management Service interface. This completes the external view of the component.



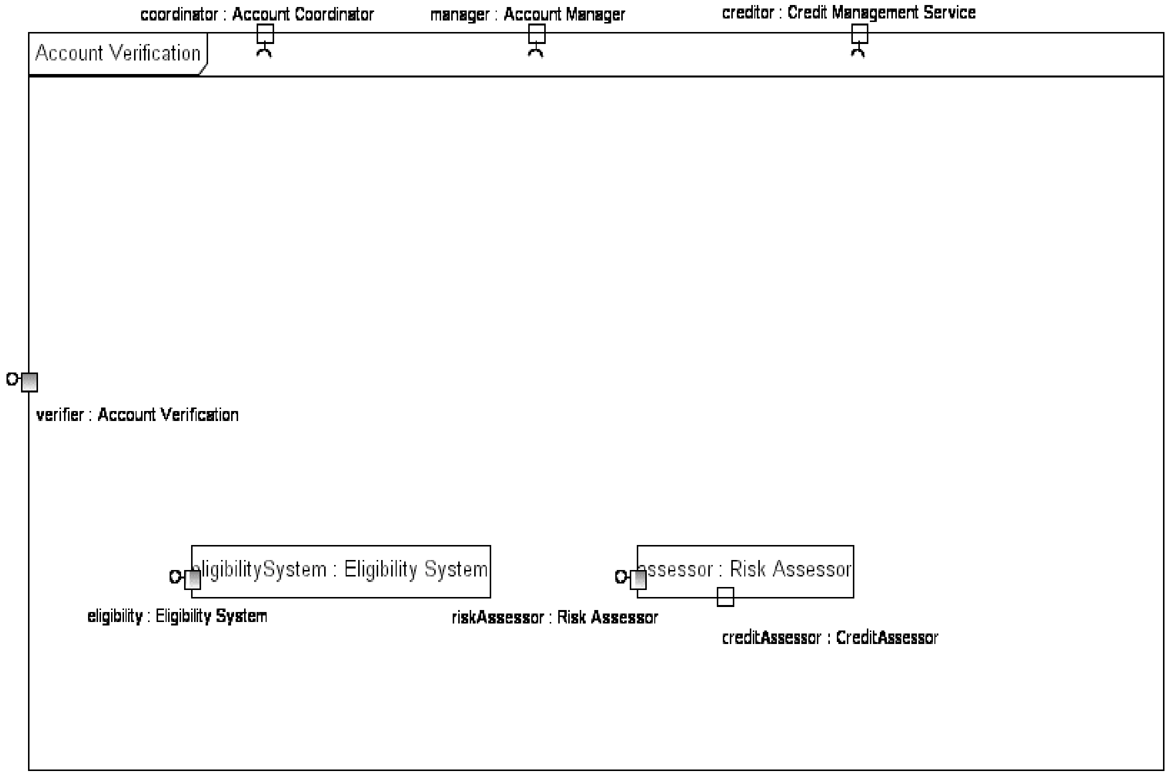
«interface»
Account Verification
+ Account Verification (application : Customer Application) : Approval Message
«component»
Account Verification
verifier : Account Verification
coordinator : Account Coordinator
manager : Account Manager
creditor : Credit Management Service
Account Coordinator
+ Initial Application Review ( )
+ Request More Documentation ( )
+ Generate Decline ( )
+ Provide Pricing & Approval ( )
Account Manager
+ Final Application Review ( )
Credit Management Service
+ Retrieve Credit Report ( )
«use»
«use»
«use»
«portclass»
Account Coordinator
«portclass»
Account Manager
«portclass»
Credit Management Service

Develop the Account Verification internal structure

Next Al starts developing the internal structure of the Account Verification component. He selects the component and adds a composite structure diagram to it for this purpose. All sees all the ports from the external view in this internal structure diagram and adds two additional parts to the component. One part is called eligibilitySystem, and its type is the Eligibility System interface. This part provides access to a particular instance of that component. The second part is called assessor and is of type Risk Assessor. This part allows the component to assess the risk associated with a customer based on their credit history. Recall that Al did not feel that these providers needed to be exposed on ports because he felt there would be no need to be able to configure the Account Verification component for different providers of Eligibility System or Risk Assessor services.

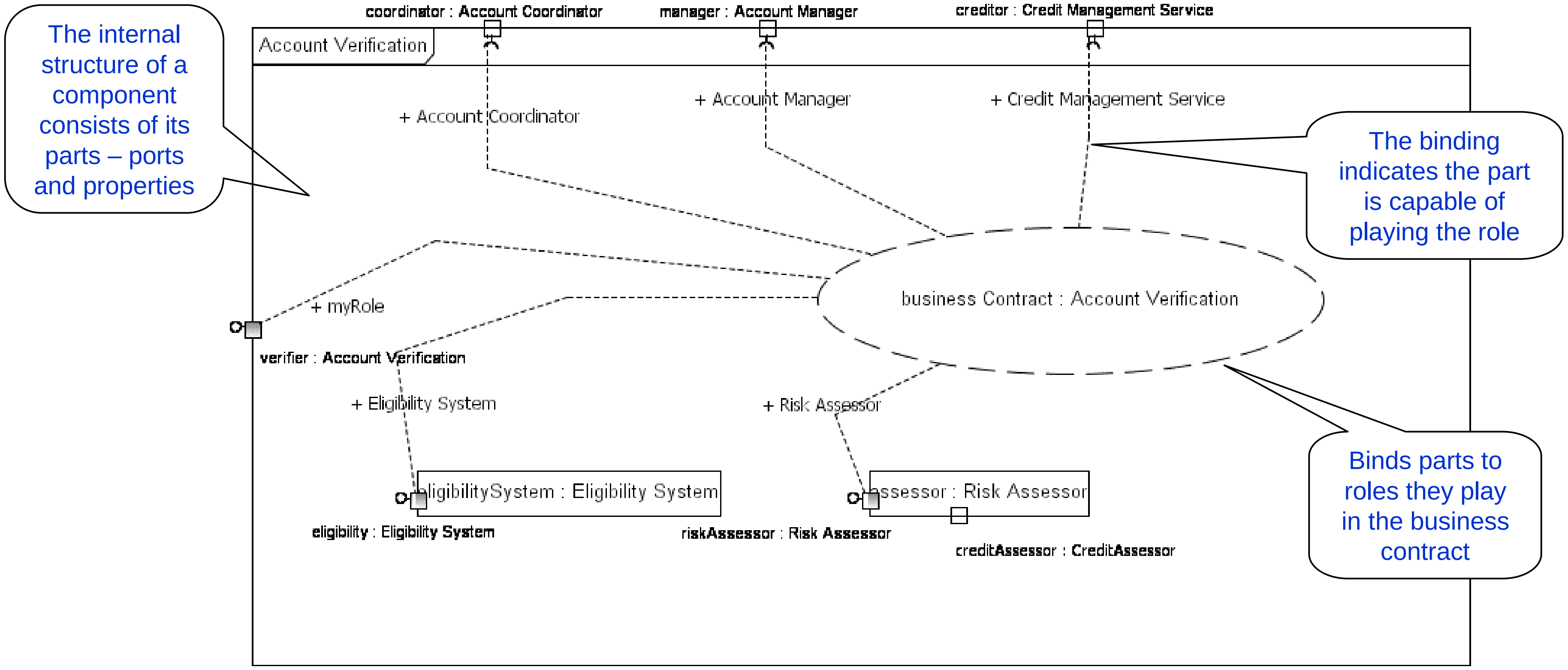


- accountverification
  - Account Verification
    - Account Verification
    - assessor
    - eligibilitySystem
    - coordinator
    - creditor
    - manager
    - verifier
    - «portclass» Account Coordinator
    - «portclass» Account Manager
    - «portclass» Credit Management Service
    - Account Verification
    - business Contract
  - Account Verification
  - Account Verification Contract
  - Account Verification Rules
  - Account Verification
- eligibilitysystem


coordinator : Account Coordinator
manager : Account Manager
creditor : Credit Management Service
Account Verification
verifier : Account Verification
eligibilitySystem : Eligibility System
assessor : Risk Assessor
eligibility : Eligibility System
riskAssessor : Risk Assessor
creditAssessor : CreditAssessor

Link the component to the business contract it fulfills

AI now ties the component's internal structure back to the business operational requirements it fulfills. He adds a collaboration use to the internal structure of the Account Verification component, and creates bindings from each of the parts of the component (including its ports for interacting with the outside world) to the roles they play in the business process. These bindings provide AI with the flexibility of fulfilling the business process requirements with a solution architecture that addresses other IT and reuse concerns while maintaining complete traceability between the requirements and services solution.


The internal structure of a component consists of its parts – ports and properties
coordinator : Account Coordinator
manager : Account Manager
creditor : Credit Management Service
Account Verification
+ Account Coordinator
+ Account Manager
+ Credit Management Service
The binding indicates the part is capable of playing the role
+ myRole
business Contract : Account Verification
verifier : Account Verification
+ Eligibility System
+ Risk Assessor
eligibilitySystem : Eligibility System
assessor : Risk Assessor
eligibility : Eligibility System
riskAssessor : Risk Assessor
creditAssessor : CreditAssessor
Binds parts to roles they play in the business contract

Create the account Verification service implementation

Now Al is ready to deal with the details of implementing the Account Verification service operation provided by the AccountVerification component. He begins by creating an Account Verification activity in the component and sets its operation to the corresponding provided service operation. When selecting the service operation, the chooser shows only operations of interfaces provided by the component for selection. Setting the specification operation automatically creates the parameters for the activity, and the activity parameter nodes in the activity diagram. Al looks back at the activity specifying the rules for how the role interact in the business services specification, and creates an activity model that performs similar actions but using the services accessed through the coordinator, manager, and creditor ports, and the eligibility system and risk assessor properties. We know this activity is realizing the rules of the business process because of the binding between the provided port and the role in the business service contract.

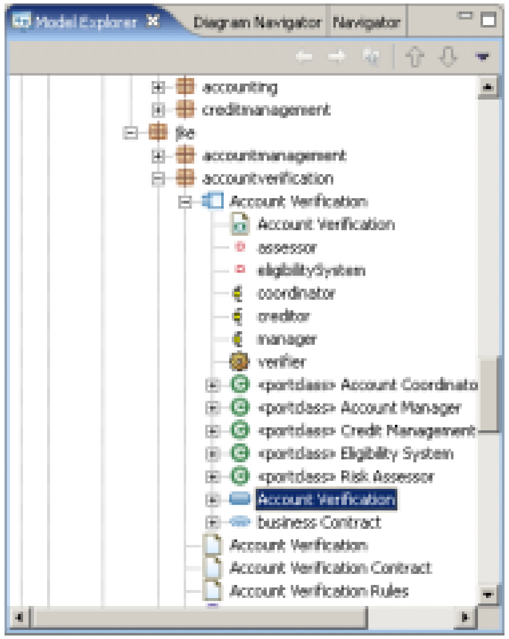
Al also looks at the business service contract and sees that the collaboration contains a number of constraints that realize KPIs for the business goals and objectives. The component must provide detailed constraints that realize each of the constraints in the contract. He adds these constraints. There is no need to directly connect the realizing constraints to the specification constraints because this is already implicit in the fact that the component contains a collaboration use whose type is the business contract. These component constraints are used to specify the results expected of the component, and what should potentially be monitored to ensure the component implementation actually achieves its results. These constraints can then be traced up through the business services contracts to the KPIs they realize, all the way to the business goals and objectives JK Enterprises is trying to achieve.

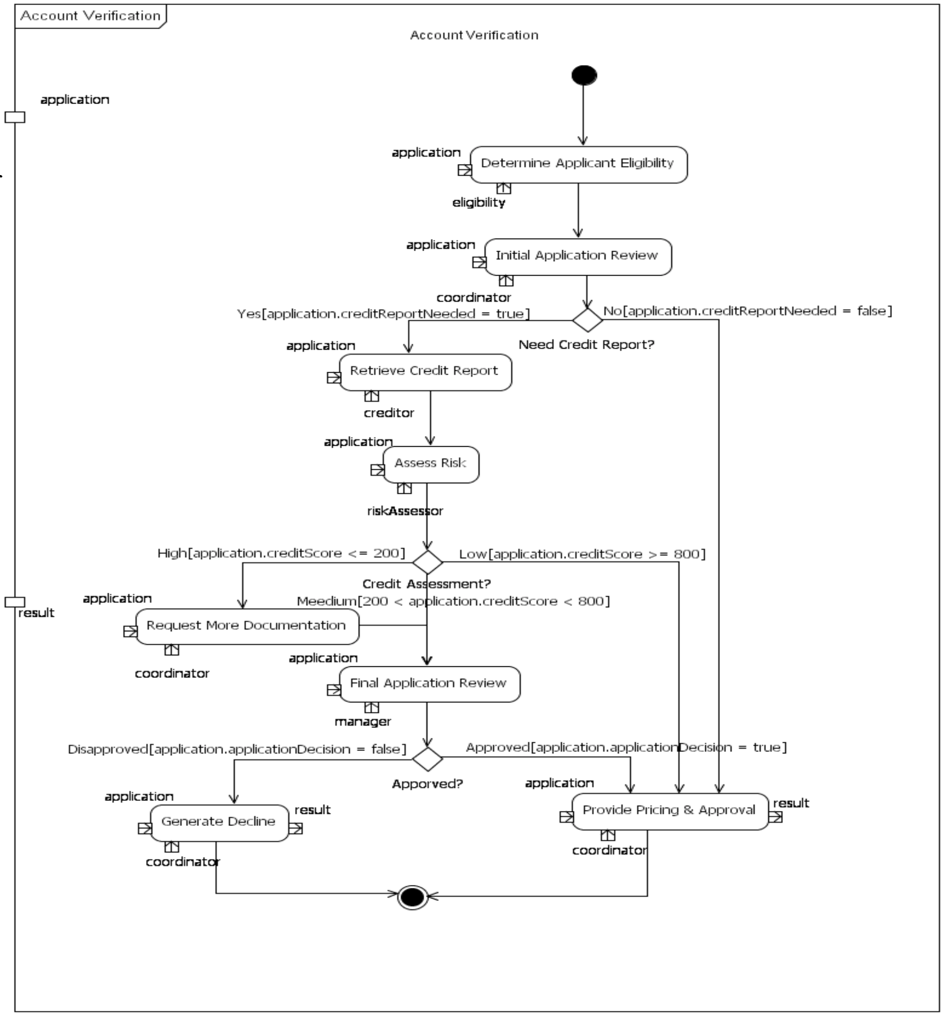
Al has now completed the services design and is ready to create the initial solution implementation.

Account Verification implementation

The Account Verification activity is the method for the Account Verification operation provided by the component

«interface» Account Verification
+ Account Verification (application : Customer Application) : Approval Message

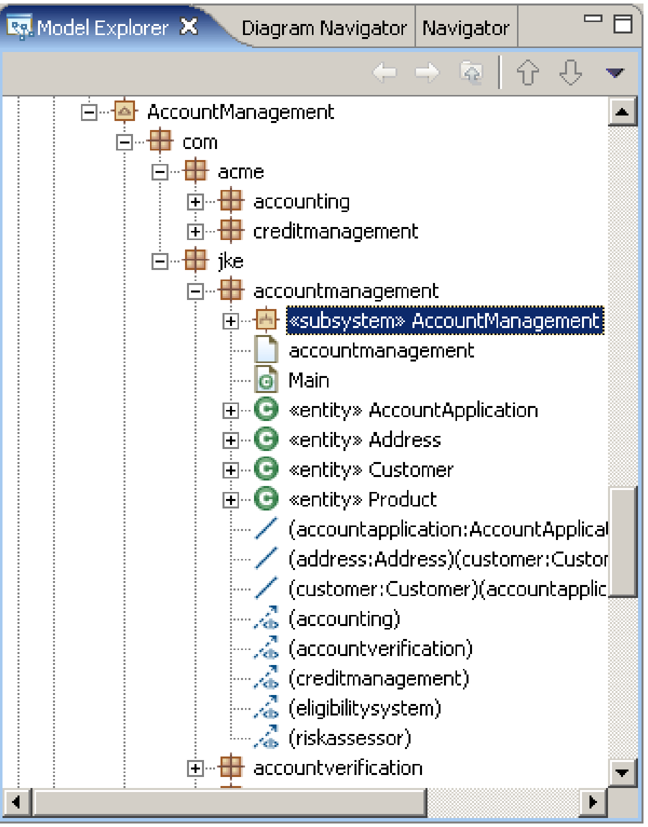




Account Verification
Account Verification
application
application
Determine Applicant Eligibility
eligibility
application
Initial Application Review
coordinator
Yes[application.creditReportNeeded = true]
No[application.creditReportNeeded = false]
Need Credit Report?
application
Retrieve Credit Report
creditor
application
Assess Risk
riskAssessor
High[application.creditScore <= 200]
Low[application.creditScore >= 800]
Credit Assessment?
Meedium[200 < application.creditScore < 800]
result
application
Request More Documentation
coordinator
application
Final Application Review
manager
Disapproved[application.applicationDecision = false]
Approved[application.applicationDecision = true]
Apporved?
application
application
Generate Decline
result
coordinator
Provide Pricing & Approval
result
coordinator

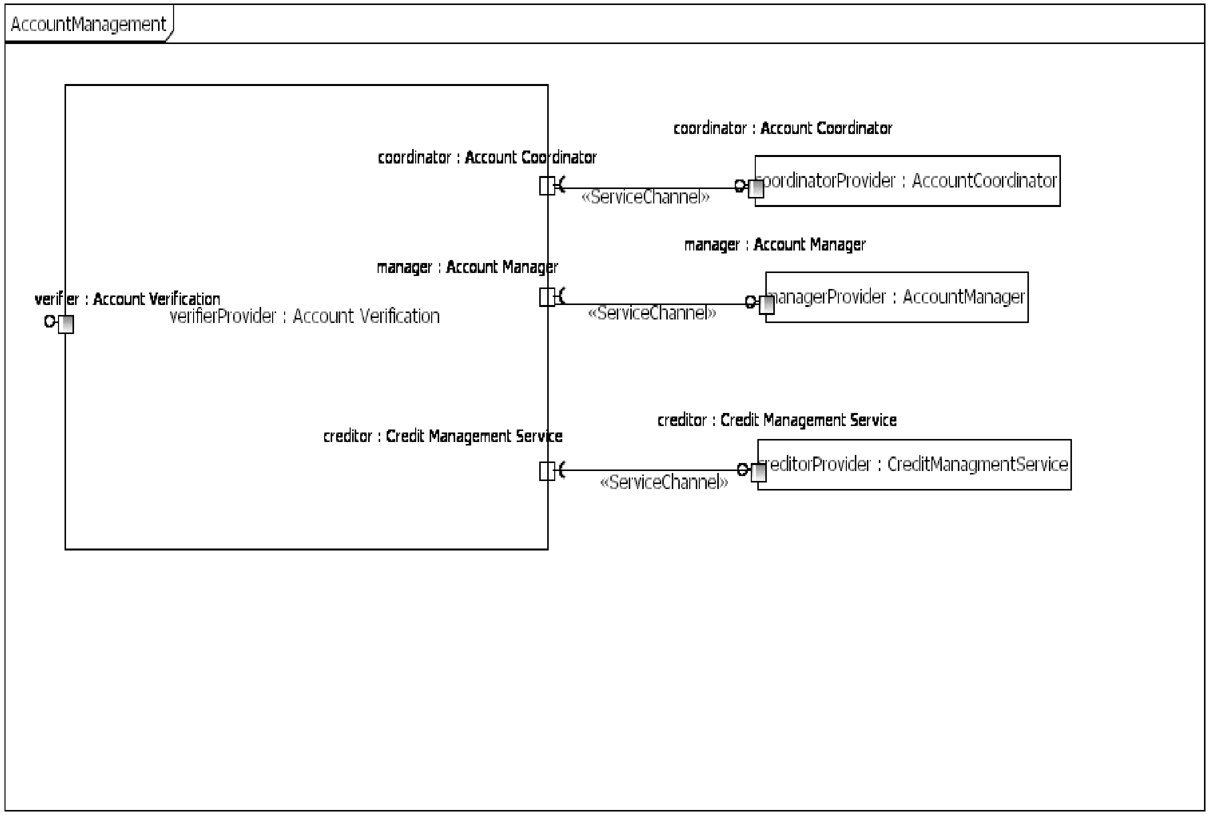
Assemble the Account Management Solution

Al is aware of existing components that provide the Account Coordinator, Account Management and Credit Management Service interfaces. To complete the solution development, he assembles these service providers into a complete, deployable solution for the AccountManagement application.



Model Explorer | Diagram Navigator | Navigator

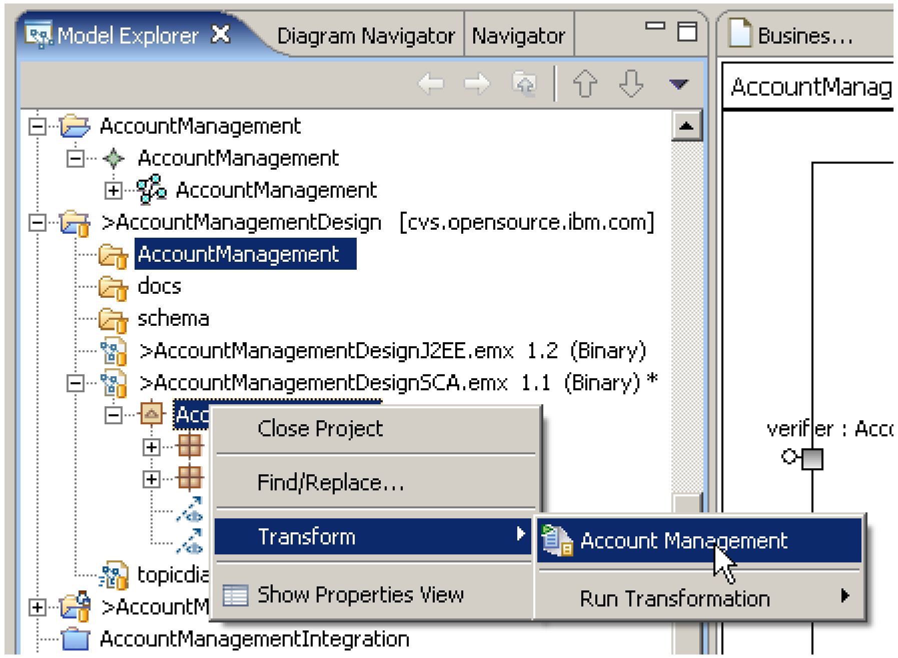
- AccountManagement
  - com
    - acme
      - accounting
      - creditmanagement
    - jke
      - accountmanagement
        - «subsystem» AccountManagement
        - accountmanagement
        - Main
        - «entity» AccountApplication
        - «entity» Address
        - «entity» Customer
        - «entity» Product
        - (accountapplication:AccountApplical
        - (address:Address)(customer:Custor
        - (customer:Customer)(accountapplic
        - (accounting)
        - (accountverification)
        - (creditmanagement)
        - (eligibilitysystem)
        - (riskassessor)
      - accountverification


AccountManagement
coordinator : Account Coordinator
coordinator : Account Coordinator
coordinatorProvider : AccountCoordinator
«ServiceChannel»
manager : Account Manager
manager : Account Manager
managerProvider : AccountManager
«ServiceChannel»
verifier : Account Verification
verifierProvider : Account Verification
creditor : Credit Management Service
creditor : Credit Management Service
creditorProvider : CreditManagmentService
«ServiceChannel»

Generate the implementation based on the chosen architecture

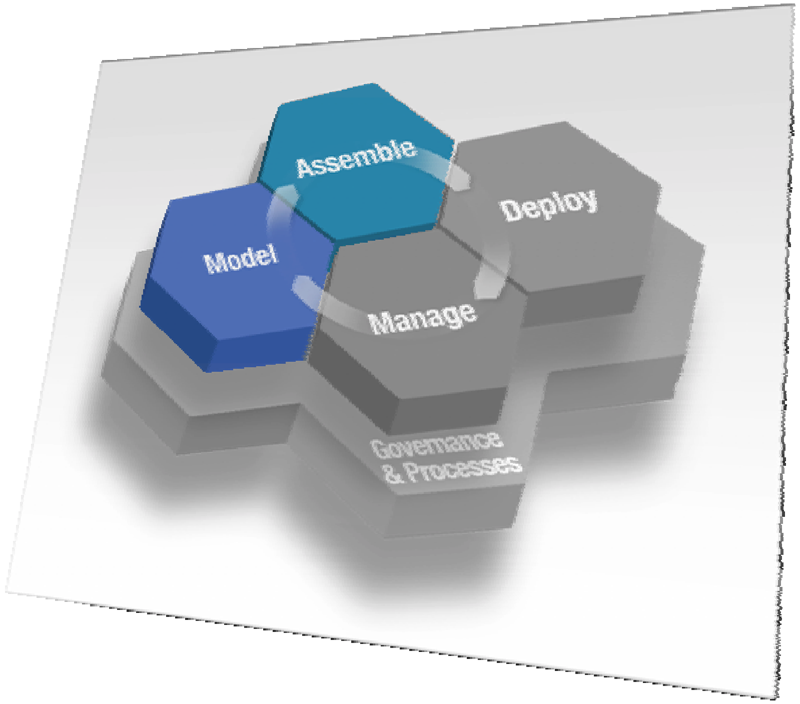
Al has decided to use Web Services to realize his SOA solution. He selects the Web Services transform and configures the transform to generate the Web Services using the desired interchange styles and transformation options.

Selecting Web Services as an implementation target for the component adds another profile to the model and extends the component, its parts and provided and required interfaces with additional modeling detail required to perform the transformation. Al fills in this detail as best he can, but the developer may update this information in order to generate a better solution implementation (Notice this introduces a transition between the architect and developer where they have to share and update the design model).



Agenda

- Business Driven Development
  - A development process for deriving solutions from business objectives
- Software Development Platform for BDD and SOA
- Capture Business Goals and Objectives
- Analyze Business Processes to realize Business Goals
- Architect a Services Solution
- Assemble, Deploy & Test
- Summary



Assemble, Deploy & Test

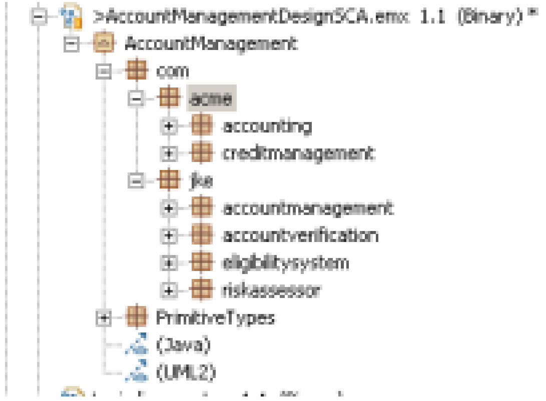
Architect Al has completed a services design model that fulfills the business requirements and has generate the initial solution artifacts using UML to SOA transforms. Developer Dan now must complete the solution by assembling the parts, implementing the incomplete components, deploying the solution and testing to make sure it works. Specifically, Dan must:

- Review the architected solution generated by UML-toSOA
- Implement the persistence mechanisms to allow service implementations to access the persistent data sources
- Implement the eligibilityImpl.determineApplicationEligibility method
- Implement the riskAssessorImpl.assessRisk method by invoking the CreditRiskAssessor session bean
- Create UI portals for the human tasks.
- Assemble the AccountVerification model by binding to the AccountCoordinatorImpl, AccountManagementImpl, and CreditManagementServiceImpl
- Deploy to IBM WebSphere Process Server
- Test

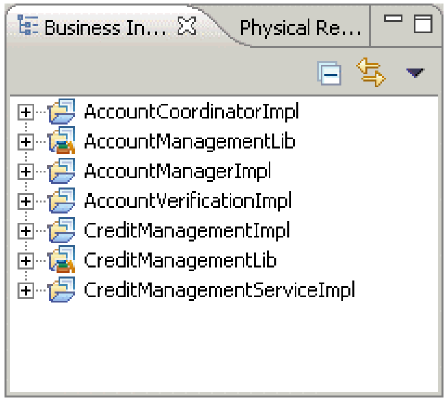
Review the architected solution

A WID library is created for each model involved in the transformation. The libraries corresponding to the models implement the business objects and interfaces defined by classes and interfaces in the UML model. The business object and interfaces are contained in folders and have namespaces based on the package containment hierarchy in the model.

A WID module is created for each component in the transformed models.

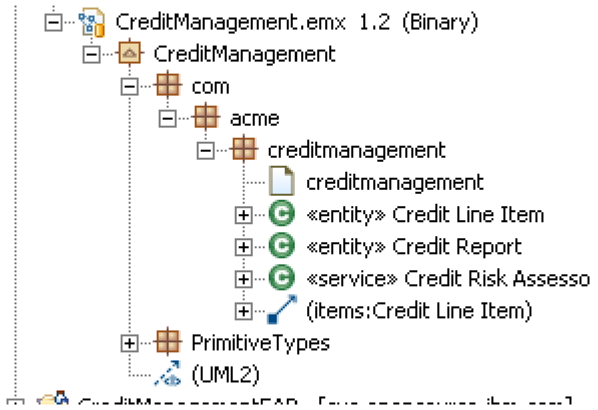


- >AccountManagementDesignSCA.emx 1.1 (Binary) *
  - AccountManagement
    - com
      - acme
        - accounting
        - creditmanagement
      - jke
        - accountmanagement
        - accountverification
        - eligibilitysystem
        - riskassessor
    - PrimitiveTypes
    - (Java)
    - (UML2)



Business In... Physical Re...

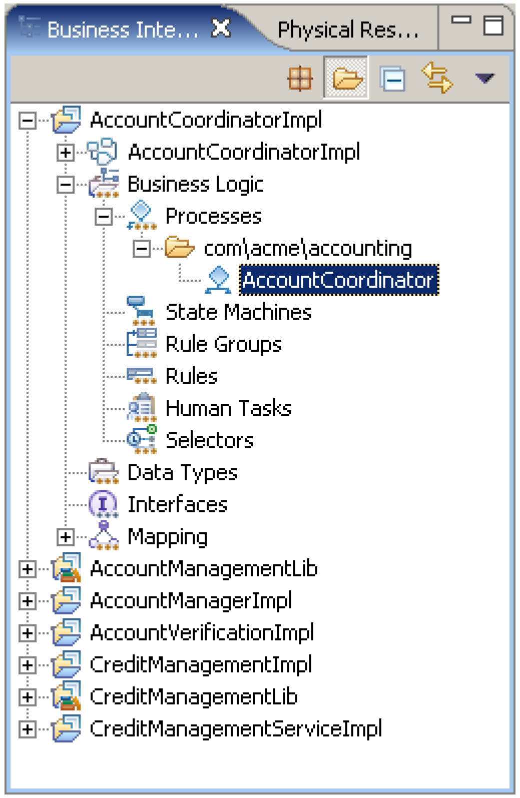
- AccountCoordinatorImpl
- AccountManagementLib
- AccountManagerImpl
- AccountVerificationImpl
- CreditManagementImpl
- CreditManagementLib
- CreditManagementServiceImpl



- CreditManagement.emx 1.2 (Binary)
  - CreditManagement
    - com
      - acme
        - creditmanagement
          - creditmanagement
          - «entity» Credit Line Item
          - «entity» Credit Report
          - «service» Credit Risk Assesso
          - (items:Credit Line Item)
    - PrimitiveTypes
    - (UML2)

AccountCoordinatorImpl

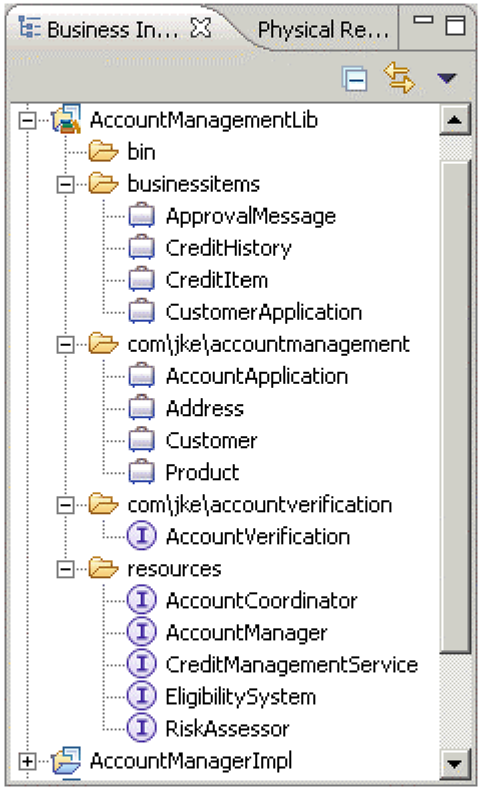
Provides an implementation of the AccountCoordinator interface required by the AccountVerification process. This component is implemented by a BPEL process containing an activity for each operation of the AccountCoordinator interface.



```
AccountCoordinatorImpl
    AccountCoordinatorImpl
    Business Logic
        Processes
            com\acme\accounting
                AccountCoordinator
        State Machines
        Rule Groups
        Rules
        Human Tasks
        Selectors
    Data Types
    Interfaces
    Mapping
AccountManagementLib
AccountManagerImpl
AccountVerificationImpl
CreditManagementImpl
CreditManagementLib
CreditManagementServiceImpl
```

AccountManagementLib

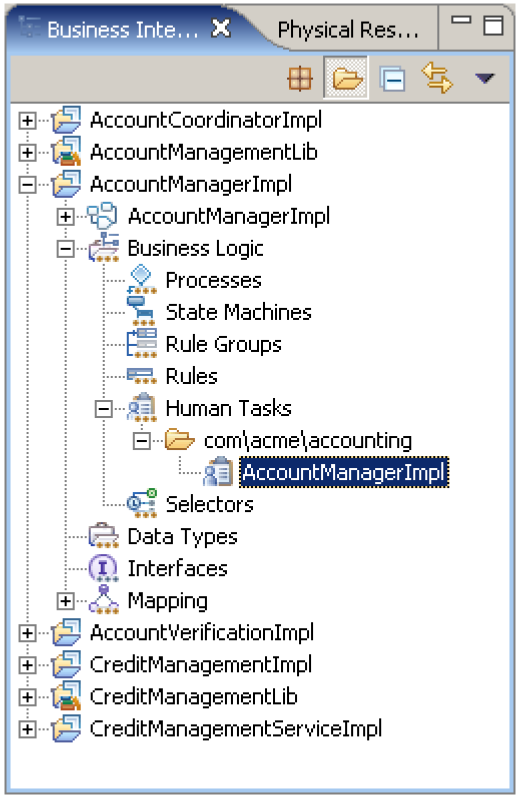
This library contains the business objects and interfaces generated from the AccountManagement business services model. It also includes the business objects and interfaces from the AccountManagementDesign model since that model has the same name as the business services model. This was done to integrate the elements from the business services and architected solution models. This library is used by many of the other libraries for shared data and interfaces.



```
AccountManagementLib
  bin
  businessitems
    ApprovalMessage
    CreditHistory
    CreditItem
    CustomerApplication
  com\jke\accountmanagement
    AccountApplication
    Address
    Customer
    Product
  com\jke\accountverification
    AccountVerification
  resources
    AccountCoordinator
    AccountManager
    CreditManagementService
    EligibilitySystem
    RiskAssessor
AccountManagerImpl
```

AccountManagerImpl

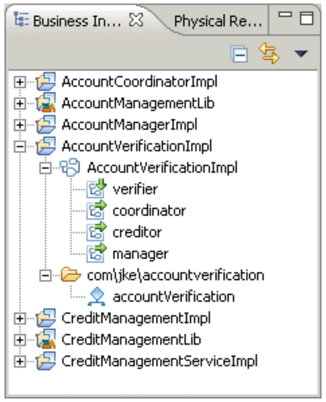
Provides an implementation of the AccountManager interface required by the AccountVerification process. This component is implemented using a human task.



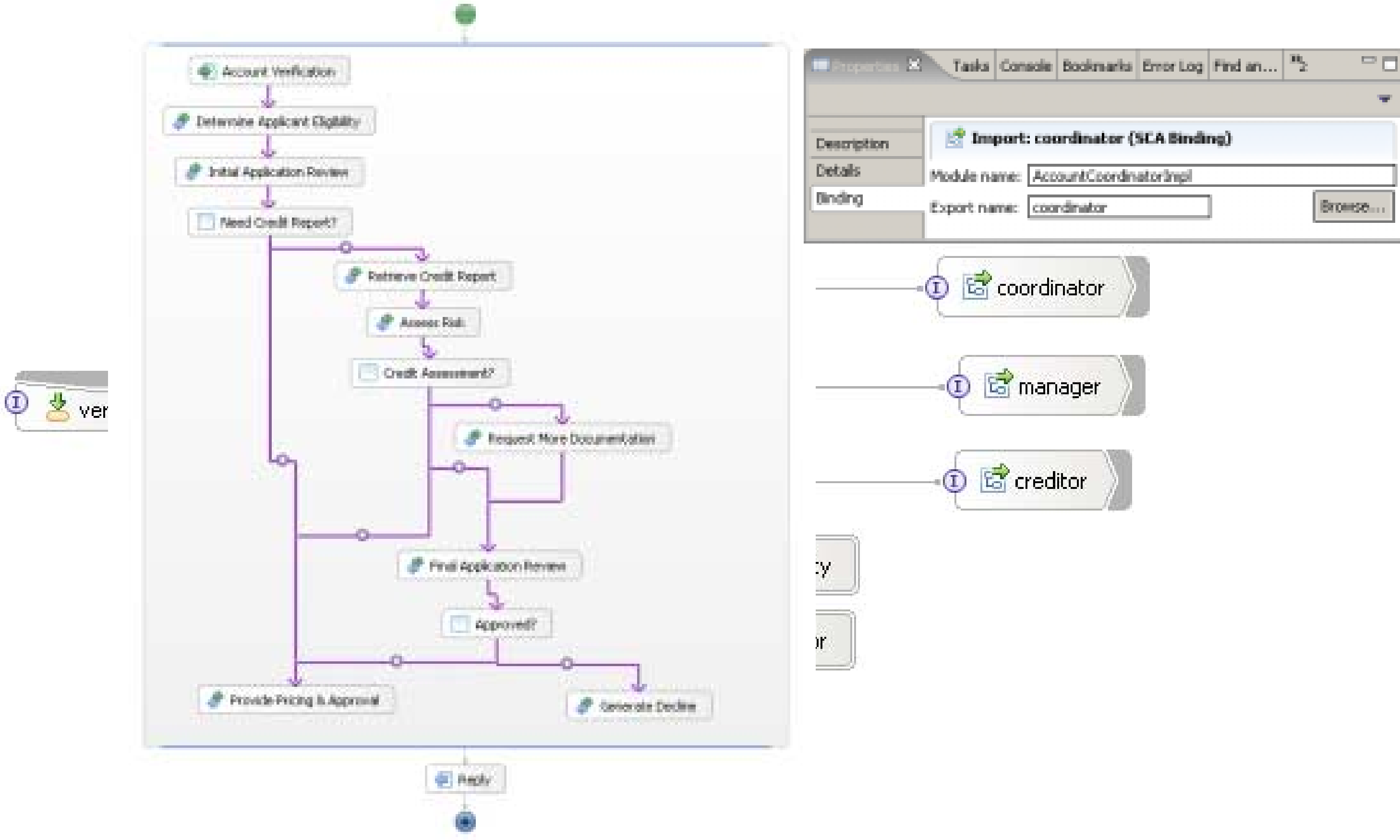
```
AccountCoordinatorImpl
AccountManagementLib
AccountManagerImpl
    AccountManagerImpl
    Business Logic
        Processes
        State Machines
        Rule Groups
        Rules
        Human Tasks
            com\acme\accounting
                AccountManagerImpl
        Selectors
    Data Types
    Interfaces
    Mapping
AccountVerificationImpl
CreditManagementImpl
CreditManagementLib
CreditManagementServiceImpl
```

AccountVerificationImpl

This module provides and implementation of the AccountVerification interface using a BPEL process. It is generated from the AccountVerification component in the solution architecture model.


Business In... Physical Re...

- AccountCoordinatorImpl
- AccountManagementLib
- AccountManagerImpl
- AccountVerificationImpl
  - AccountVerificationImpl
    - verifier
    - coordinator
    - creditor
    - manager
  - com\jke\accountverification
    - accountVerification
- CreditManagementImpl
- CreditManagementLib
- CreditManagementServiceImpl


Tasks | Console | Bookmarks | Error Log | Find an...

Description | Details | Binding

Import: coordinator (SCA Binding)

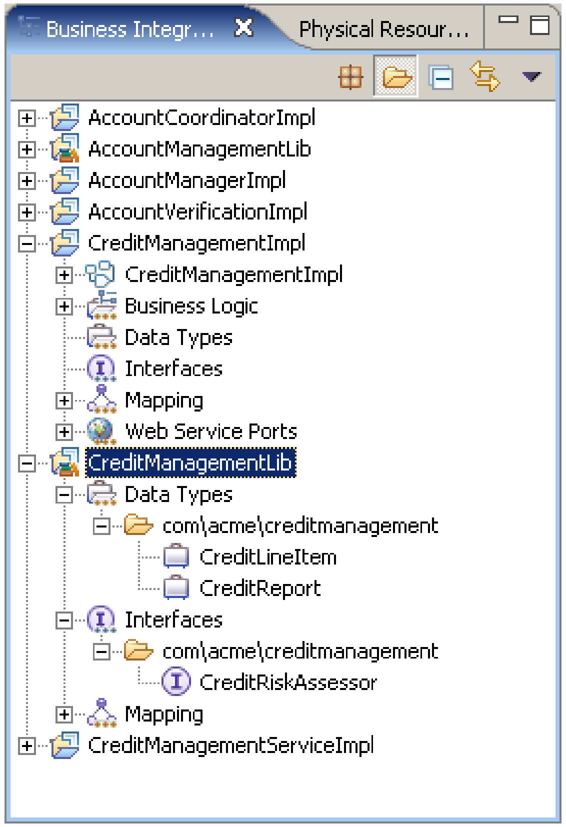
Module name: AccountCoordinatorImpl

Export name: coordinator

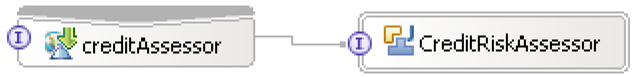
Browse...

CreditManagementLib, CreditManagementImpl

CreditManagementLib is an existing library that AI obtained from RAS and is used to implement risk assessment by adapting an existing service. CreditManagementImpl provides the service implementation using an existing EJB session bean.



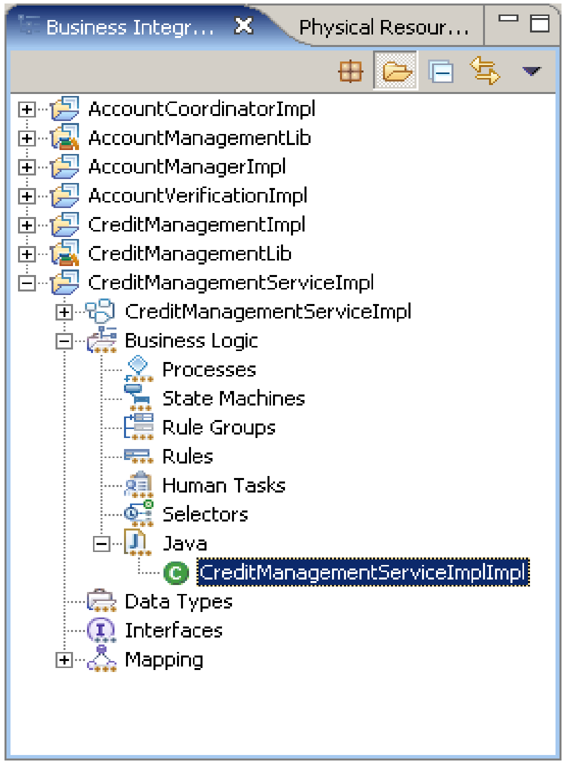
- AccountCoordinatorImpl
- AccountManagementLib
- AccountManagerImpl
- AccountVerificationImpl
- CreditManagementImpl
  - CreditManagementImpl
  - Business Logic
  - Data Types
  - Interfaces
  - Mapping
  - Web Service Ports
- CreditManagementLib
  - Data Types
    - com\acme\creditmanagement
      - CreditLineItem
      - CreditReport
  - Interfaces
    - com\acme\creditmanagement
      - CreditRiskAssessor
  - Mapping
- CreditManagementServiceImpl

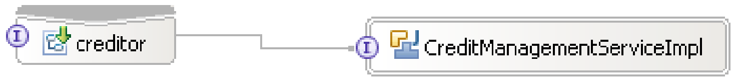


CreditManagementServiceImpl

JK Enterprises maintains some credit history information on customers in its own databases. The EligibilitySystem determineApplicationEligibility operation fills in the credit history fields of the customer application in order to calculate the initial credit score. This service is used to retrieve a credit report from a credit agency that contains more complete, and up-to-date information. The retrieveCreditReport could be a service provided by others get the report, or a human task might be used to enter data from a printed report.

This module provides an implementation of the CreditManagementService interface required by the AccountVerification process. This service provides the access to a credit report while CreditManagementLib provides a service to calculate credit scores. The service is implemented in Java (and currently does nothing in this sample).

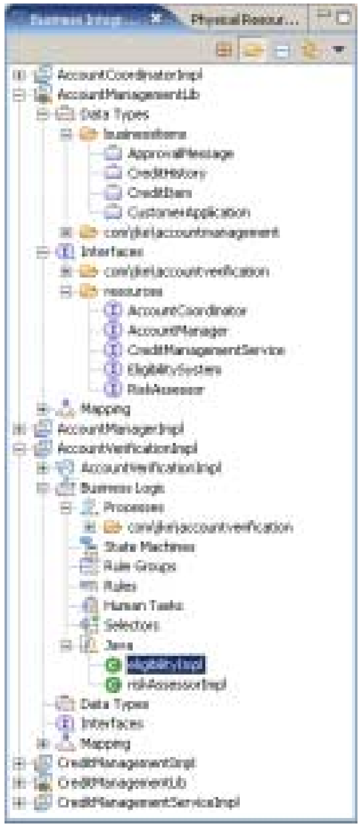




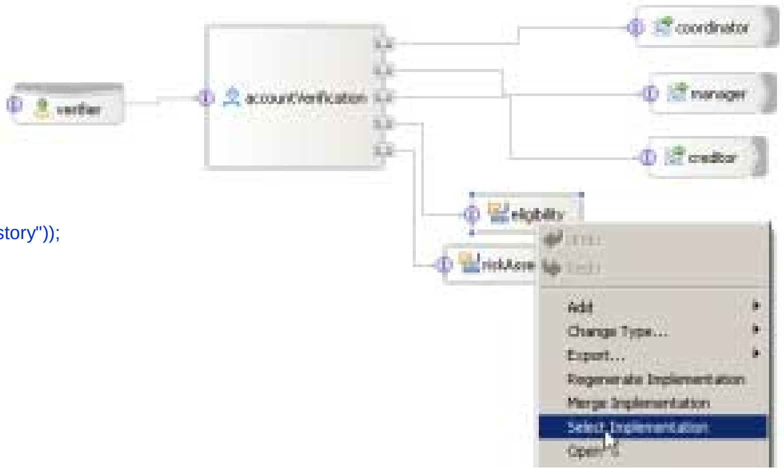
```
/**
 * Method generated to support implemention of operation "retrieveCreditReport" defined for WSDL port type
 * named "interface.CreditManagementService".
 *
 * The presence of commonj.sdo.DataObject as the return type and/or as a parameter
 * type conveys that its a complex type. Please refer to the WSDL Definition for more information
 * on the type of input, output and fault(s).
 */
public DataObject retrieveCreditReport(DataObject credappin) {
                        //TODO Needs to be implemented.
                        return null;
}
```

Implement EligibilitySystem determineApplicantEligibility

Architect Al decided the eligibility system would have a private implementation inside the AccountVerification Component. Dan provides the implementation for the component using Java.

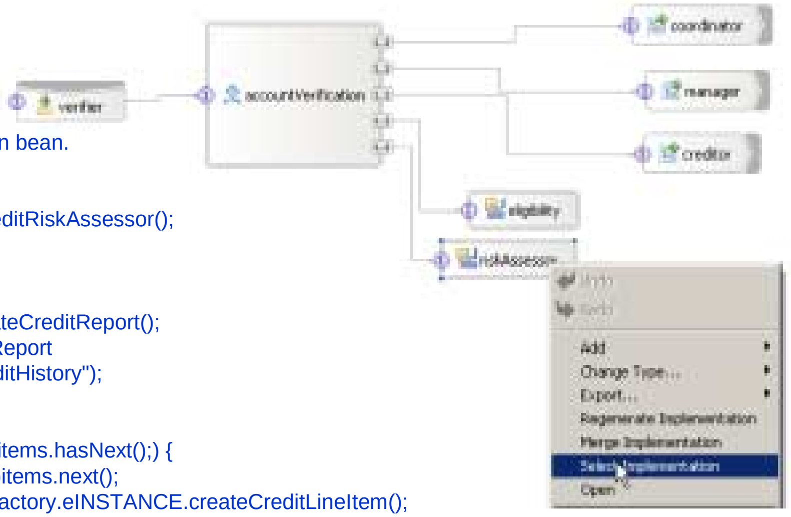


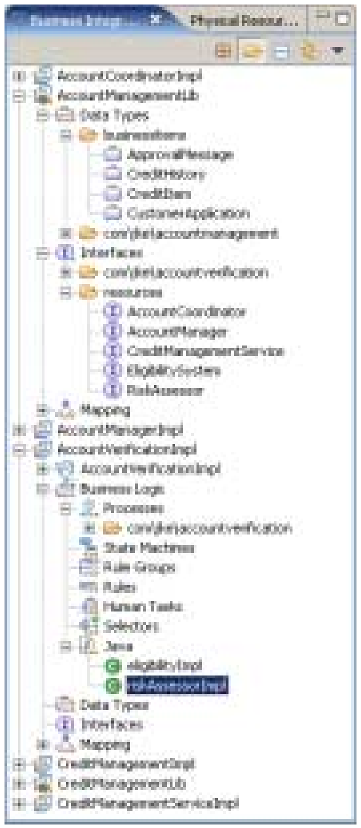
```
public DataObject determineApplicantEligibility(DataObject credappin) {
    boolean isEligible = true;
    // Get the Credit History for the applicant
    CreditReportFacadeRemote creditReportMediator = createCreditReportFacadeRemote();
    CreditReport creditReport = null;
    try {
        creditReport = creditReportMediator
                      .getCreditReportByKey(new CreditReportKey(
                         credappin.getString("contactFirstName"),
                         credappin.getString("contactLastName")));
    } catch (FindException e1) {
    } catch (RemoteException e1) {
        e1.printStackTrace();
    }
    credappin.createDataObject("creditHistory");
    creditReport2CreditHistory(creditReport, credappin.getDataObject("creditHistory"));
    // Ensure the necessary data has been provided:
    // customer last name must be provided
    isEligible = credappin.getString("contactLastName") != null
                && credappin.getString("contactLastName") != "";
    // customer must live in the US in order to be automatically eligible
    isEligible = credappin.getString("customerCountry") == "USA";
    // the request amount must be specified and less than $5,000.
    try {
        isEligible = (new Double(credappin.getString("requestAccountAmount")))
                      .doubleValue() < 5000.00;
    } catch (NumberFormatException e) {
        isEligible = false;
    }
    // last credit score must be known and less than 4
    int creditScore = credappin.getInt("creditScore");
    isEligible = creditScore != 0 && creditScore < 4;
    credappin.setBoolean("applicationDecision", isEligible);
    return credappin;
}
```



Implement the RiskAssessor assessRisk operation

The RiskAssessor interface is realized by an adapter on an existing J2EE session bean that provides a getCreditScore method. This implementation copies the credit history information from the customer application into the parameter for getCreditScore, invokes the session bean, and then scales the returned credit score to the range expected by the RiskAssessor interface.



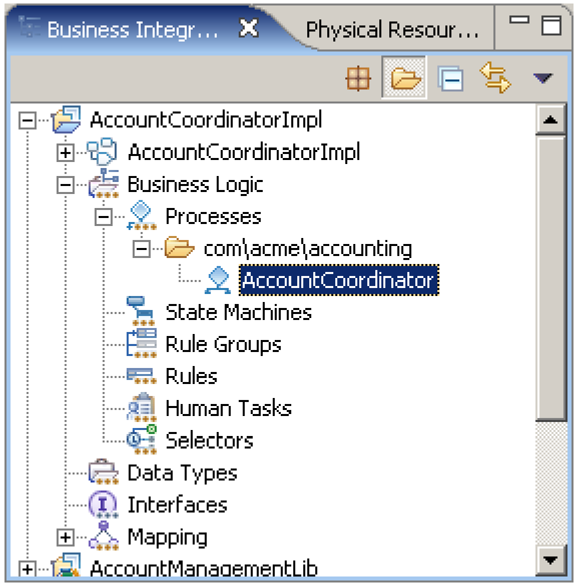


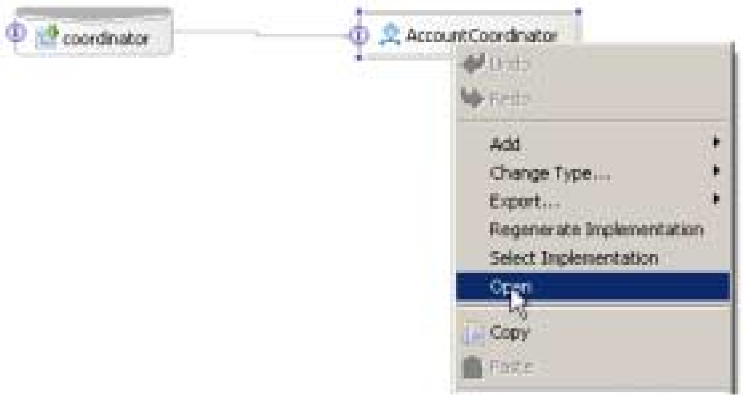
```
public DataObject assessRisk(DataObject credappin) {
        // Get an instance of the CreditRiskAssessor session bean.
        // It is this service we are adapting
        if (creditRiskAssessor == null) {
                creditRiskAssessor = createCreditRiskAssessor();
        }
        // Create a CreditReport that we can map to the
        // CustomerApplication CreditHistory property
        CreditReport report = SdoFactory.eINSTANCE.createCreditReport();
        // Copy the data from the CreditHistor to the CreditReport
        DataObject history = credappin.getDataObject("creditHistory");
        report.setFirstName(history.getString("firstName"));
        report.setLastName(history.getString("lastName"));
        for (Iterator items=history.getList("items").iterator(); items.hasNext();) {
                DataObject item = (DataObject)items.next();
                CreditLineItem lineItem = SdoFactory.eINSTANCE.createCreditLineItem();
                lineItem.setId(new Integer(item.getInt("id")));
                lineItem.setCompany(item.getString("company"));
                lineItem.setAmountOwed(item.getDouble("amountOwed"));
                report.getItems().add(item);
        }
        // Get the credit score from the reused service
        try {
                // scale the credit score from 1..10 to 1..1000 which is
                // expected by the RiskAssessor interface
                credappin.setInt("creditScore", creditRiskAssessor.getCreditScore(report).intValue()*100);
        } catch (RemoteException e) {
                credappin.setInt("creditScore", 0); // means unknown
        }
        return credappin;
}
```

Implement the AccountCoordinator operations

The AccountCoordinator is responsible for a number of tasks. Although these tasks are all assigned to the AccountCoordinator role in the business process, developer Dan has decided that each of these responsibilities may need to be implemented different ways, and that these implementations may change over time. For example, the generateDecline operation could be implemented as a human task which displays an input field in which a person playing the AccountCoordinator role could type the refusal message. Alternatively Dan could decide to implement the generateDecline by automatically generating the decline message from the CustomerApplication data.

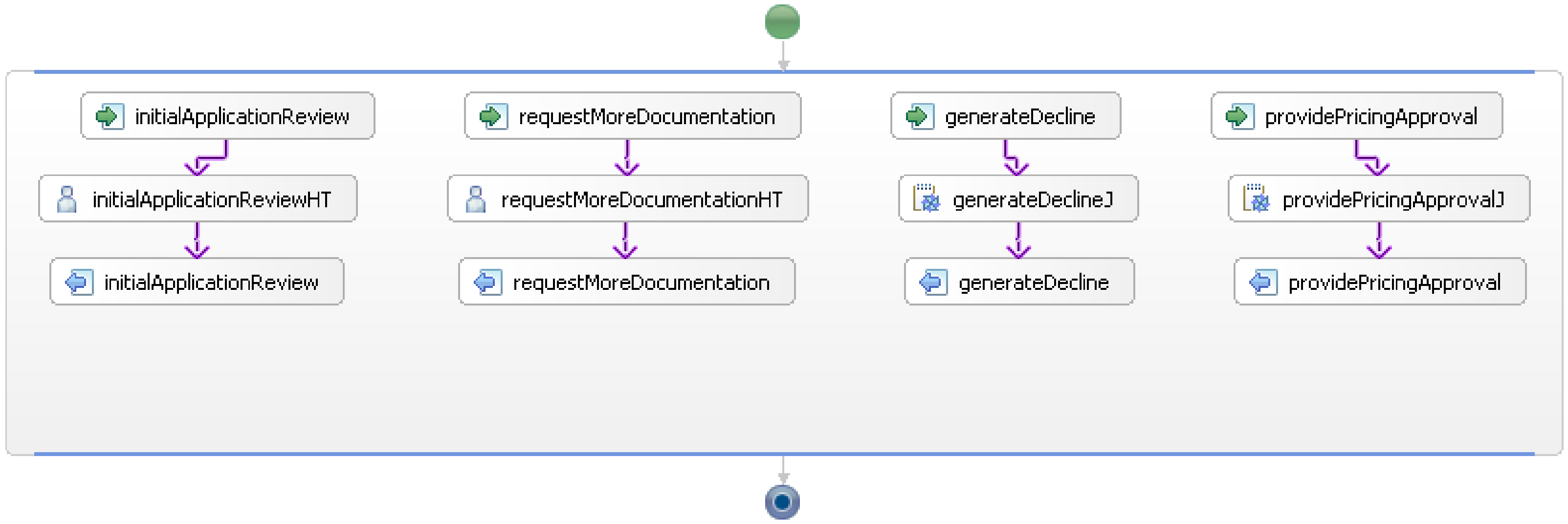
Dan could split up the AccountCoordinator operations into separate interfaces so that the interfaces for human tasks only contained a single operation, but this couples the external client interface design with the implementation style which Dan knows could easily change in the future. So Dan decides to leave all the operations in the AccountCoordinator interface as the business analyst and architect specified, and provide an implementation which supports different implementation types for each operation in the interface. To do this Dan creates an assembly for the AccountCoordinatorImpl which exports the AccountCoordinator interface and wires that interface to an AccountCoordinator long-running process to implement the operations.





Implementing the AccountCoordinator process

Inside the AccountCoordinator process, Dan creates a flow containing a receive activity for each operation in the provided interface. Each of these receive activities can create a new process instance. The initialApplicationReview receive activity is connected by a link to an initialApplicationReviewHT human task which is used to do the work. This is followed by a reply that returns the updated CustomerApplication. This completes the initialApplicationReview operation implementation. Dan creates similar implementations for the other operations. For generateDecline and providePricintApproval, Dan uses Java snippets activities to automatically generate the accept and decline messages based on information in the CustomerApplication.



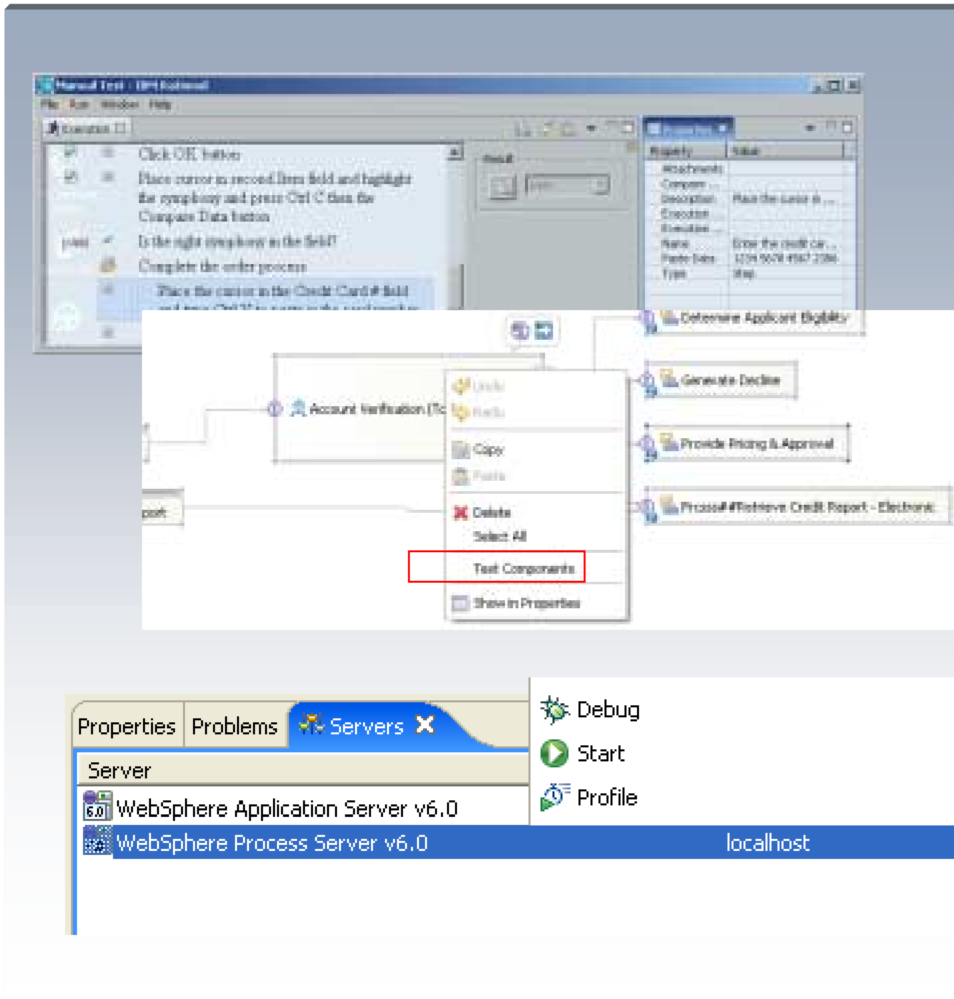
Human Task and User Interface Development

- The WID Human Task editor configures the task
  - Sets task administrator, creator, potential owner, editor and reader
  - The client UI to use
  - Escalation settings
- A portal client provides the UI
  - The human task links to the portal client using a unique portlet name
  - The portlet knows the WSDL interface for the page

Test Early, Test Often

- Testing needs to occur across Business Driven Development:
  - Component
  - Service
  - Business Process
  - Composite Application
  - Functional
  - User Interface
  - Performance
  - Regression
  - System

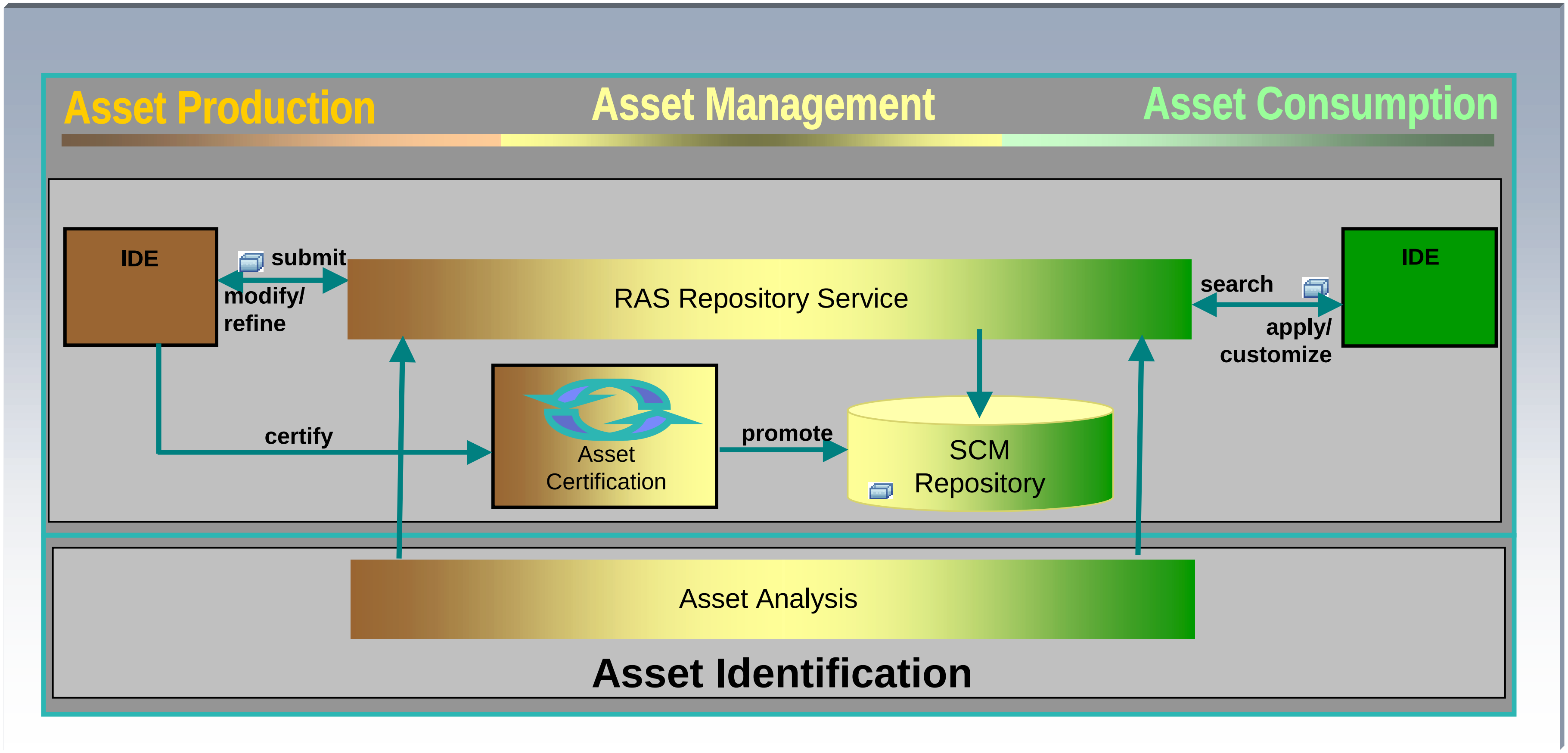
- Integrated set of test tools (that support SOA) and an integrated test environment is important



Manual Test - IBM Rational
Click OK button
Place cursor in second Item field and highlight the symphony and press Ctrl C then the Compare Data button
Is the right symphony in the field?
Complete the order process
Place the cursor in the Credit Card # field
Account Verification
Determine Applicant Eligibility
Generate Decline
Provide Pricing & Approval
Process#Retrieve Credit Report - Electronic
Undo
Redo
Copy
Paste
Delete
Select All
Test Components
Show in Properties
Properties
Problems
Servers
Server
WebSphere Application Server v6.0
WebSphere Process Server v6.0
localhost
Debug
Start
Profile

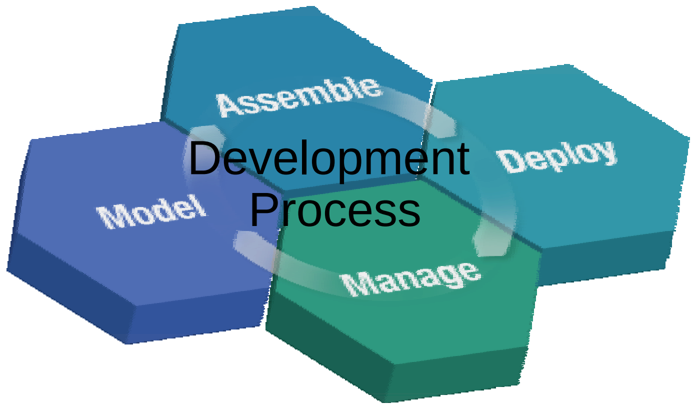
Development-time Service Lifecycle

- At development time services are:
  - Identified, Produced, Consumed, and Managed



Governance for Business Driven Development

A governed lifecycle end to end



An approach and tools that effectively enable organizations to :

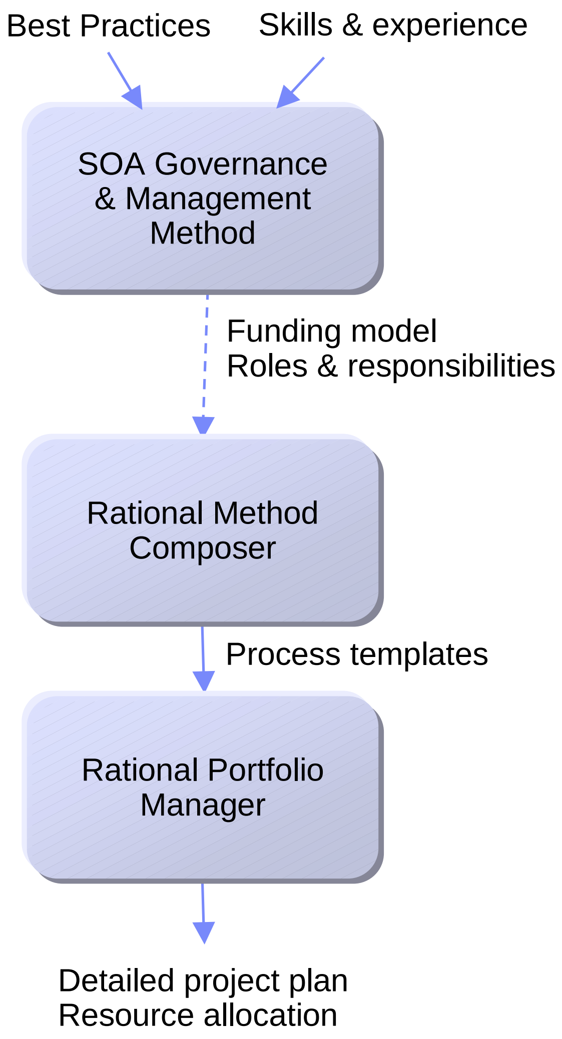
- Determine the business priorities
- Execute development against those priorities
- Measure their effectiveness

In the context of a secure / governed infrastructure;

- Supports complex sourcing models (including geographically disperse)
- Provides development compliance (audit trails and security that is transparent to the developers)

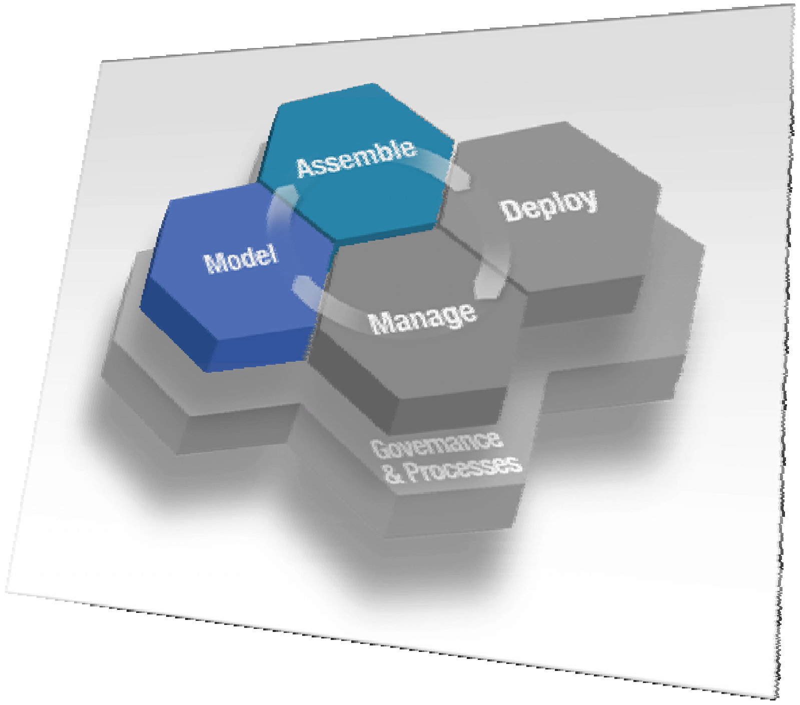
Doing Governance for Business Driven Development

- **Methods and best practices**
  - **IBM SOA Governance & Management Method**
- **Expertise and skills**
  - IBM Business Enablement Services for SOA
    - SOA Center of Excellence to help facilitate SOA adoption
    - SOA Assessment Services to determine current capabilities and gaps and develop SOA strategy
    - SOA Governance model to define roles and responsibilities, policies, measurements and controls mechanisms
  - IBM Organizational Design Services to help refine organizational model
- **Tools to document and help automate governance process**
  - **IBM SOA Governance & Management Methods for IBM Rational Method Composer** NEW!
  - IBM Rational Portfolio Manager

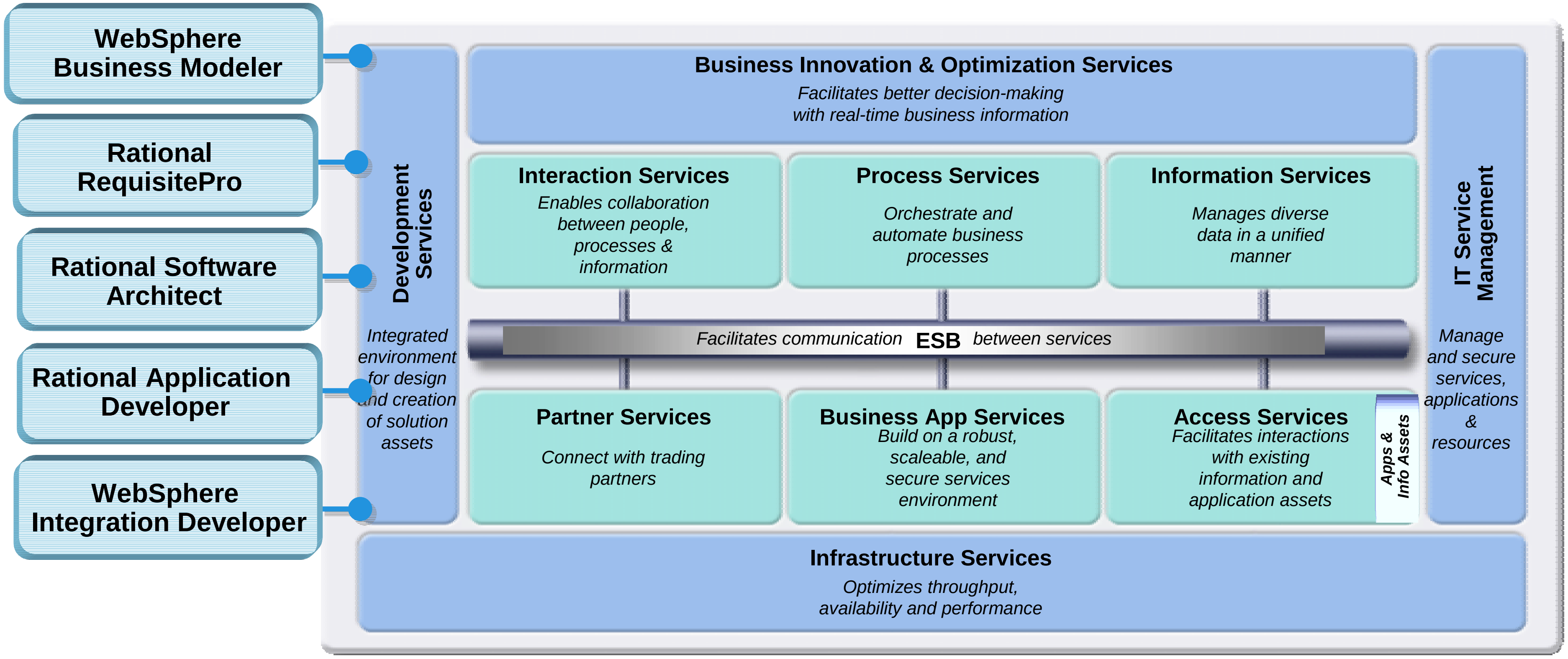


Agenda

- Business Driven Development
  - A development process for deriving solutions from business objectives
- Software Development Platform for BDD and SOA
- Capture Business Goals and Objectives
- Analyze Business Processes to realize Business Goals
- Architect a Services Solution
- Assemble, Deploy & Test
- Summary



Key Products - Business Driven Development


WebSphere Business Modeler
Rational RequisitePro
Rational Software Architect
Rational Application Developer
WebSphere Integration Developer
Development Services
Integrated environment for design and creation of solution assets
Business Innovation & Optimization Services
Facilitates better decision-making with real-time business information
Interaction Services
Enables collaboration between people, processes & information
Process Services
Orchestrate and automate business processes
Information Services
Manages diverse data in a unified manner
Facilitates communication ESB between services
Partner Services
Connect with trading partners
Business App Services
Build on a robust, scaleable, and secure services environment
Access Services
Facilitates interactions with existing information and application assets
Apps & Info Assets
IT Service Management
Manage and secure services, applications & resources
Infrastructure Services
Optimizes throughput, availability and performance

More Information

- IBM's WebSphere Platform
  - ✓ www.watchit.com/websphere
- Business Integration
  - ✓ http://www.ibm.com/software/info/topic/perform/busintegration.html
- Information on IBM WebSphere Software
  - ✓ www.ibm.com/software/websphere
- Information on IBM Rational Software
  - ✓ www.ibm.com/software/rational/sw-bycategory/index.html
- IBM, Web Services and SOA
  - ✓ http://www.ibm.com/developerworks/webservices/newto/
- IBM and Model-Driven Architecture (MDA)
  - ✓ http://www.ibm.com/software/rational/mda/
- Business Innovation and Optimization
  - ✓ http://www.ibm.com/software/info/topic/perform/

धन्यवाद

Hindi

多謝

Traditional Chinese

ขอบคุณ

Thai

Спасибо

Russian

Gracias

Spanish

Thank You

شكرا

Arabic

Obrigado

Brazilian Portuguese

Danke

German

Grazie

Italian

多谢

Simplified Chinese

Merci

French

நன்றி

Tamil

감사합니다

Korean

ありがとうございました

Japanese

Patterns Accelerate Business Driven Development

Systematic:
top-down approach to building systems

Business business processes and models for business problems	▪ Business Model/Process Patterns	▪ Cross-Industry Business Patterns
Operational Architecture solution architecture and runtime views mapped to business processes	▪ Patterns for e-Business (P4eb) ▪ Process integration patterns (P4eb) ▪ Data integration patterns (P4eb) ▪ SOA & Web Services patterns	▪ Enterprise integration patterns ▪ System management patterns (P4eb) ▪ Information integration patterns (P4eb)
Application Architecture application architecture, design and development solutions	▪ Cross-industry business application patterns (P4eb) ▪ LWP templates ▪ SOA design patterns	▪ Design patterns (ISSW, ODSD) ▪ Data model & Grid patterns ▪ Legacy transformation patterns
Deployment component and service mapping to servers and nodes	▪ PRiSM patterns ▪ Rainforest patterns ▪ Data management patterns	▪ Application & infrastructure deployment patterns ▪ Testing patterns

Opportunistic: start at any point

IBM Pattern Solutions

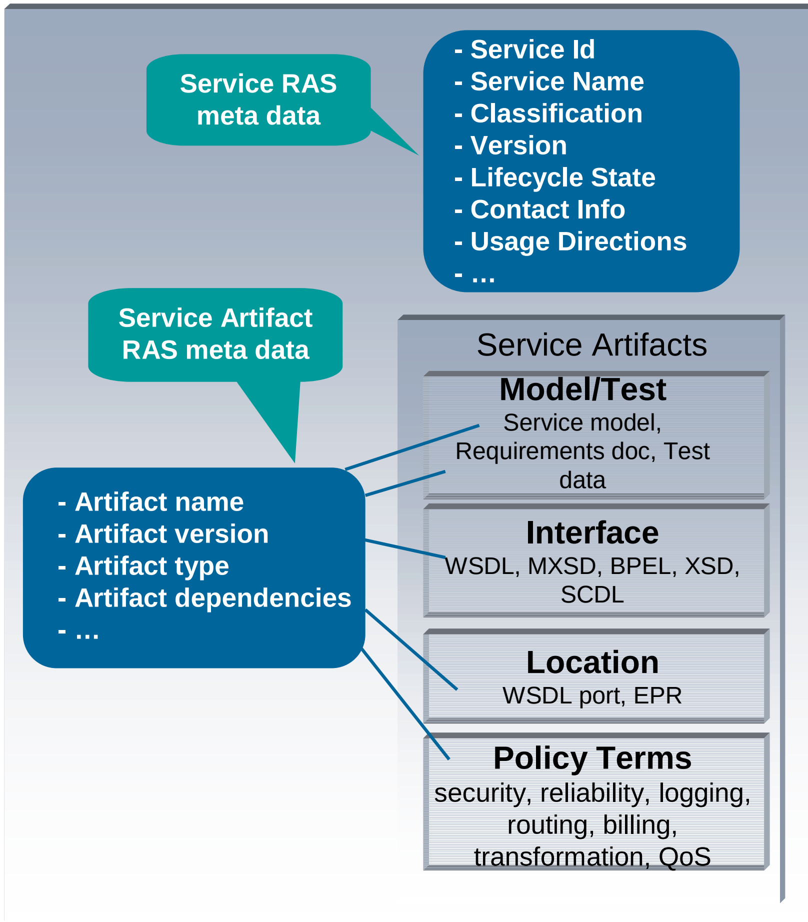
- IBM Pattern Solution Assets currently available:
  - Enterprise Patterns
    - J2EE Patterns
      - Session Façade, Business Delegate, Message Façade, etc.
  - WebSphere Platform Messaging Patterns
  - State Oriented Portlet Patterns
  - Tivoli System Automation Failover Configuration Pattern

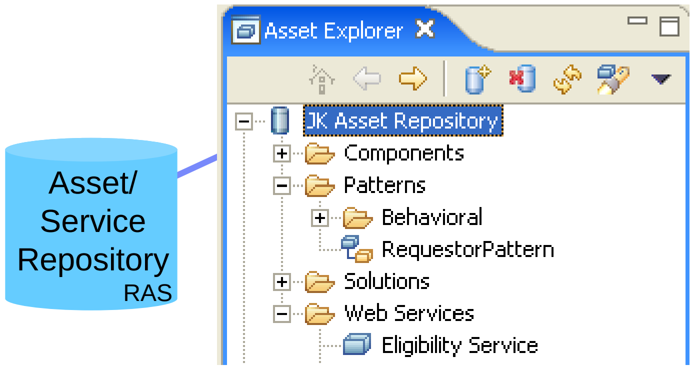
- New patterns are being developed

www.ibm.com/developerworks/rational/products/patternsolutions/

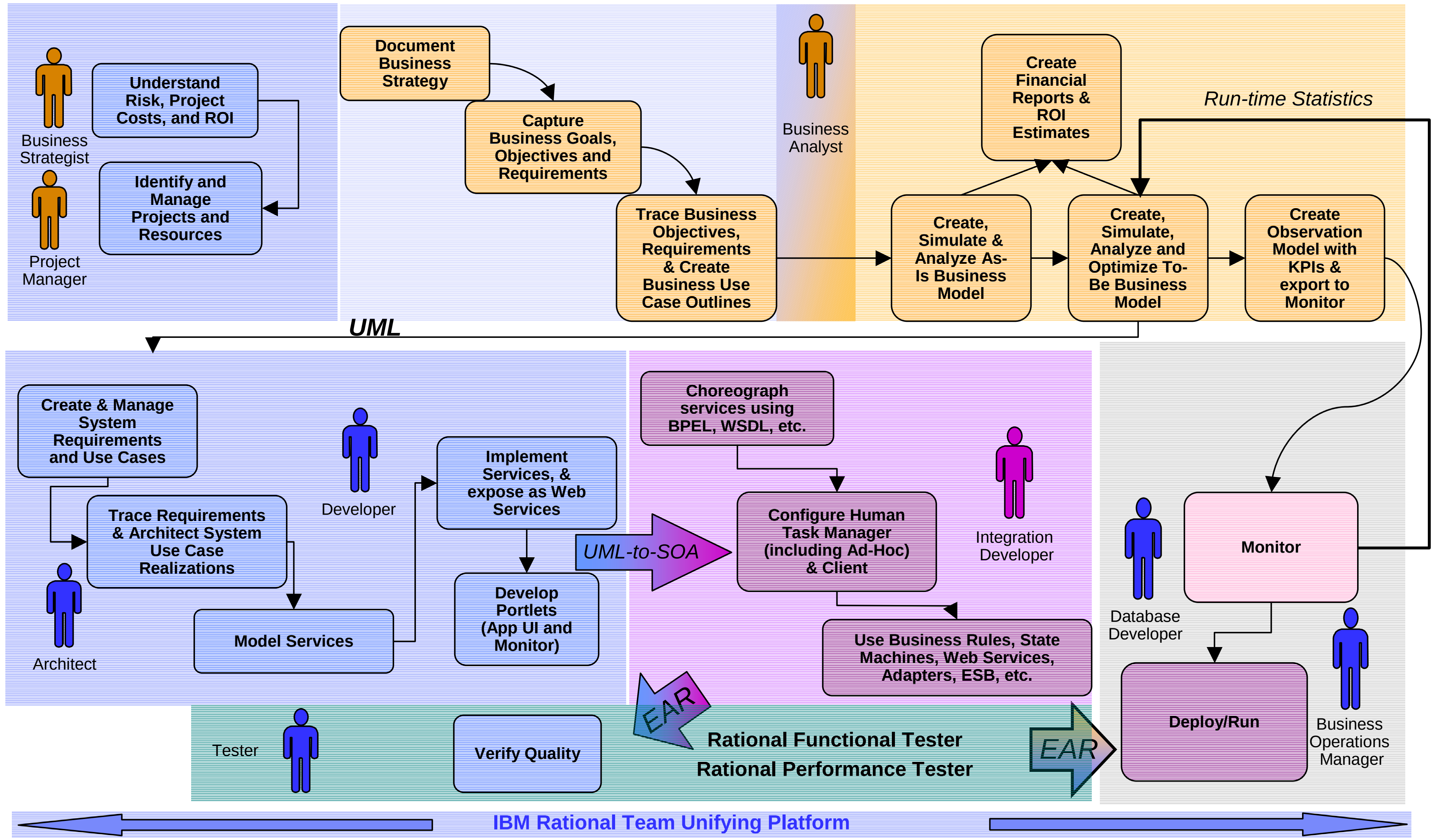
Reusable Asset Specification (RAS)

- Reusable Asset Specification (RAS)
  - A standard way to package services & other assets
  - Services are Assets
  - Assets can contain other elements:
    - Design Models, Test Cases, Documentation, etc
- Three dimensions to consider for Asset Specification
  1. Variability
  2. Granularity
  3. Articulation (The degree of completeness of the artifacts providing the solution)





Business Driven Development in the Larger Context



Business Strategist
Understand Risk, Project Costs, and ROI
Project Manager
Identify and Manage Projects and Resources
Document Business Strategy
Capture Business Goals, Objectives and Requirements
Trace Business Objectives, Requirements & Create Business Use Case Outlines
Business Analyst
Create Financial Reports & ROI Estimates
Run-time Statistics
Create, Simulate & Analyze As-Is Business Model
Create, Simulate, Analyze and Optimize To-Be Business Model
Create Observation Model with KPIs & export to Monitor
UML
Create & Manage System Requirements and Use Cases
Trace Requirements & Architect System Use Case Realizations
Architect
Model Services
Developer
Implement Services, & expose as Web Services
Develop Portlets (App UI and Monitor)
UML-to-SOA
Choreograph services using BPEL, WSDL, etc.
Configure Human Task Manager (including Ad-Hoc) & Client
Integration Developer
Use Business Rules, State Machines, Web Services, Adapters, ESB, etc.
Database Developer
Monitor
Deploy/Run
Business Operations Manager
EAR
Tester
Verify Quality
Rational Functional Tester
Rational Performance Tester
EAR
IBM Rational Team Unifying Platform