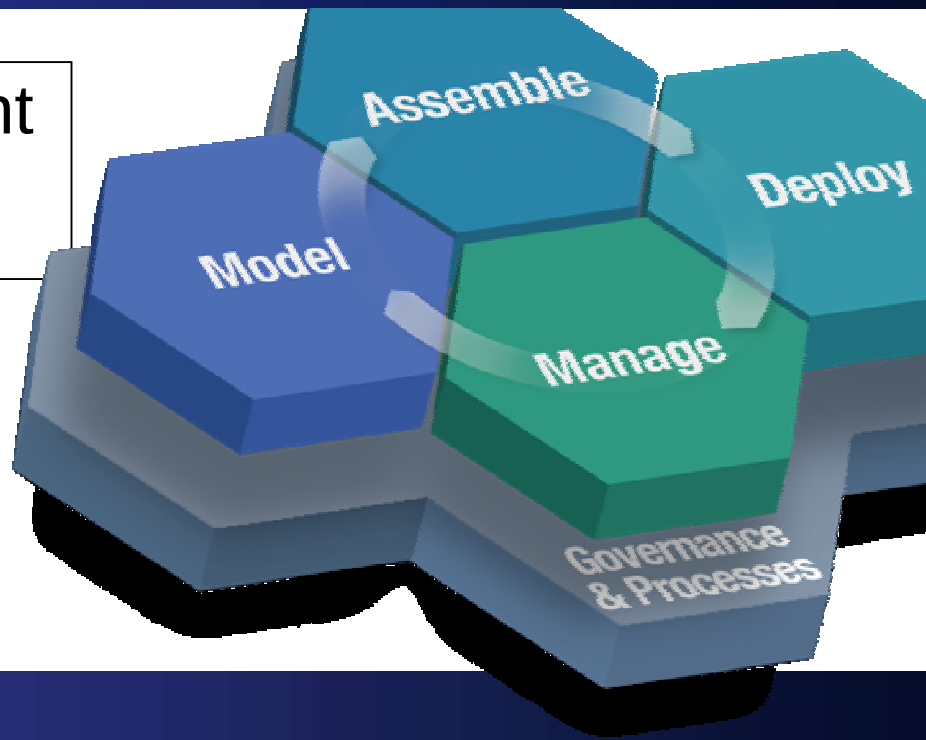
Business Driven Development for SOA

From Business Goals
to SOA Solutions that
fulfill them



Jim Amsden, IBM
jamsden@us.ibm.com

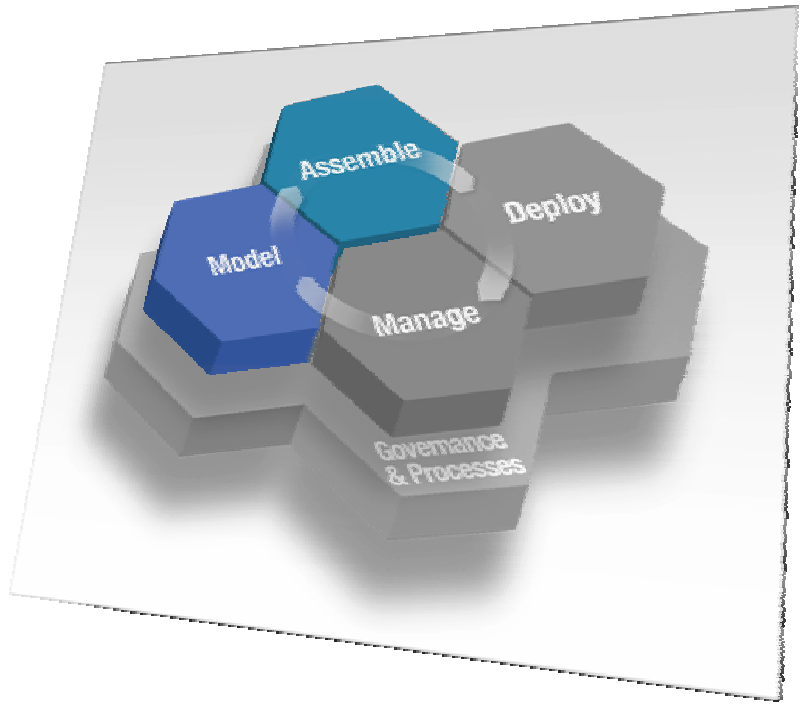
ON DEMAND BUSINESS™

Questions customers ask to leverage business opportunities:

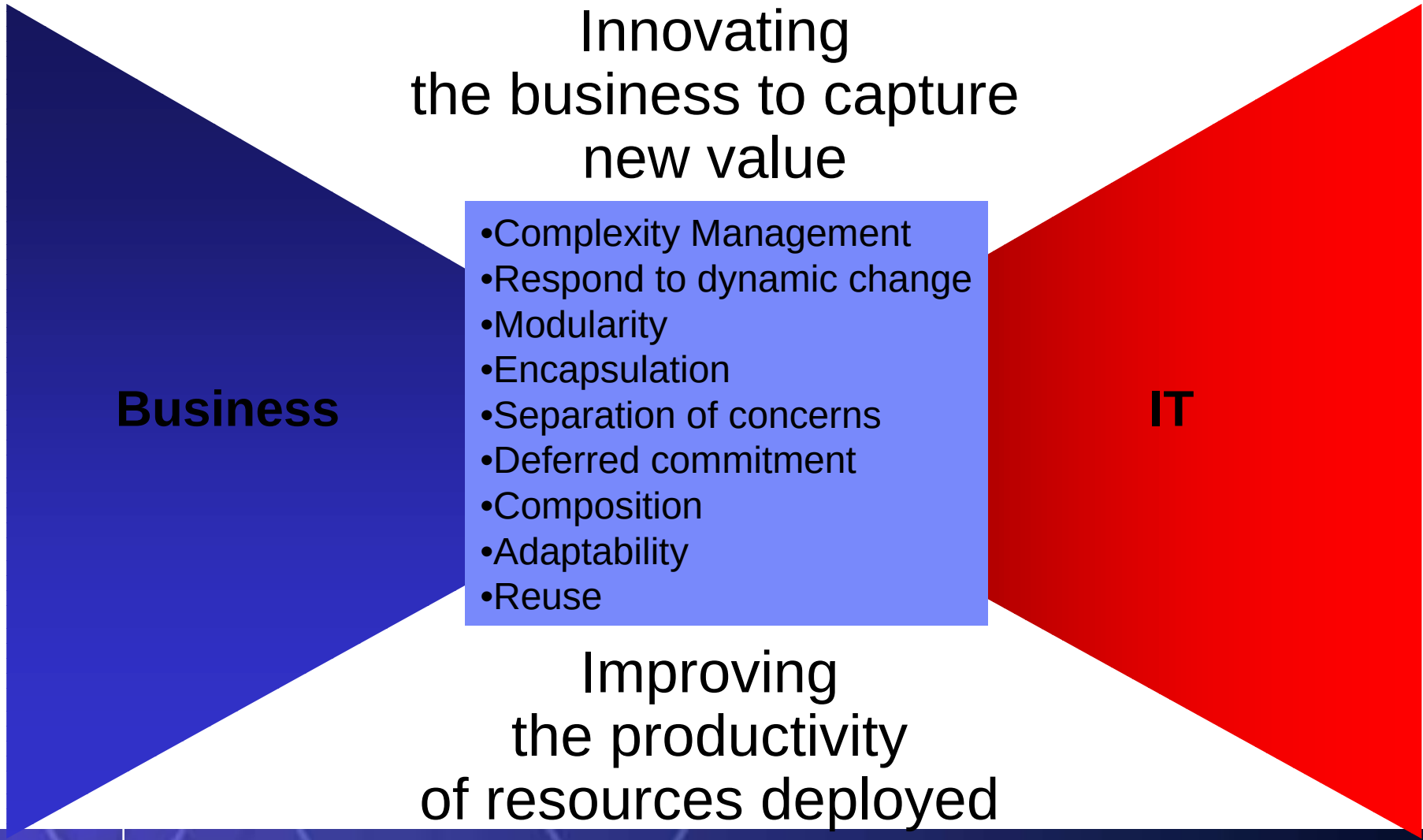
- What are we trying to accomplish and why?
- How will we know if we accomplished it?
- What business operational requirements are necessary to meet the goals and objectives?
- How do we ensure we're addressing the right requirements?
- How should we realize those operational requirements while addressing IT and business integration concerns?
- How do we create solutions based on the chosen architectures?
- How do we monitor the results and collect data to measure performance?
- How can I manage projects to make all this happen?
- How can collected data be used to improve simulation data for continuous process refinement?
- How fast can I turn the crank on solutions in order to meet rapidly changing business opportunities?

Agenda

- Bridging the gap between Business Requirements and SOA Solutions
- 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



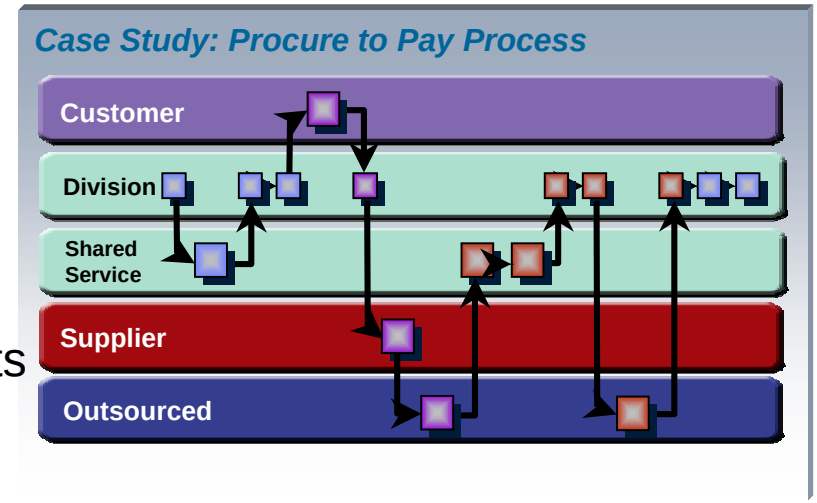
The Business and IT have to address similar concerns



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



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 design, 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

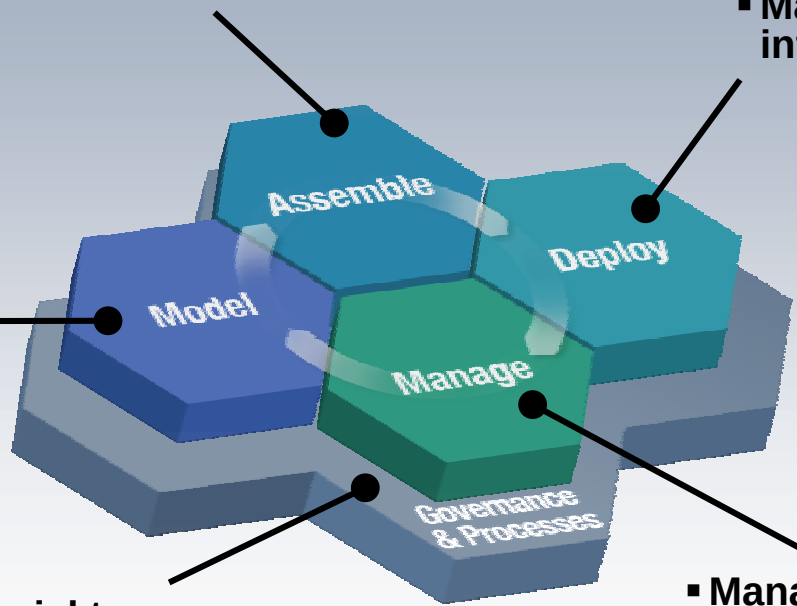
-- Focus on Flexibility and Reuse

- *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

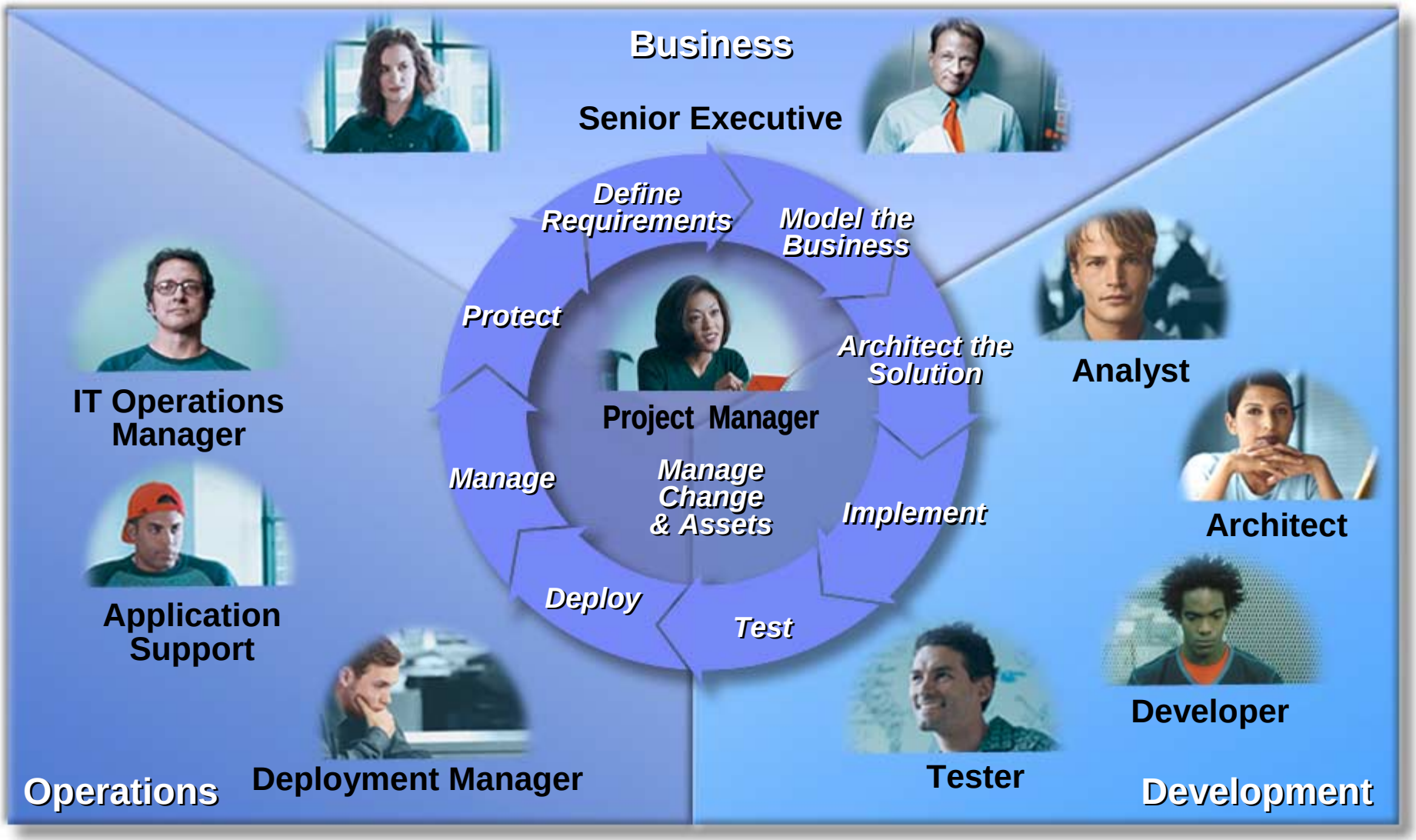


- Gather requirements
- Model & Simulate
- Design

- Financial transparency
- Business/IT alignment
- Process control

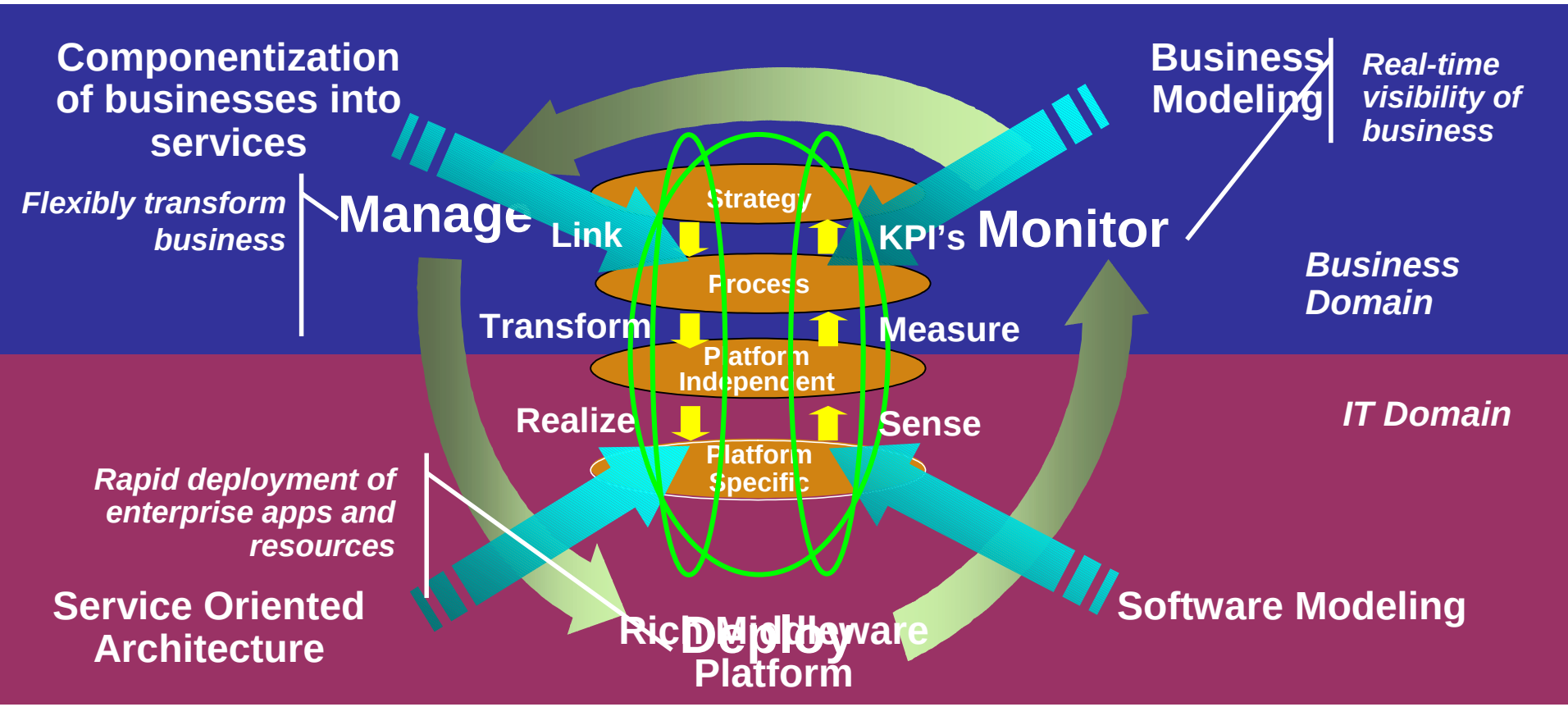
- Manage applications & services
- Manage identity & compliance
- Monitor business metrics

The Business Driven Development lifecycle



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**

Business Driven Development in a nutshell

What are we trying to accomplish and why? How can we know if our objectives have been met?

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

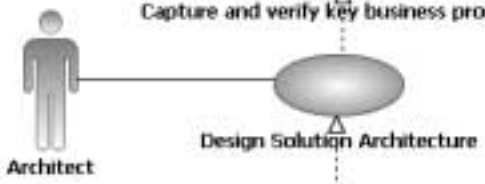
What are the business organizational and operational requirements necessary to realize these objectives?

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

How do we ensure we are addressing the right requirments within operational constraints?

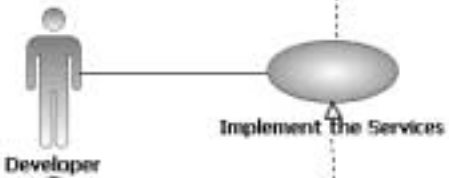
Model as-is and to-be business processes that realize the business requirements. Use simulation to measure KPIs and verify business goals are met.

How should we realize those operational requirements while addressing IT and business integration concerns?



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.

How do we create solutions based on the chosen architectures?

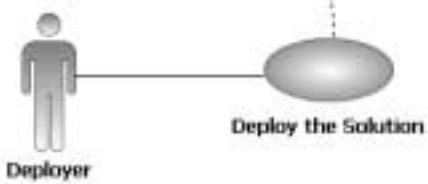


Use MDA to generate the initial the IT solution based on the chosen architecture and realizing the system use case requirements.

How can we deliver solutions that respond to rapidly changing business opportunities?

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

How do we collect data and monitor the rsults for solution validation and continuous improvement?



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

Computation Models have evolved to meet increased demand

- Computers and software applications are now involved in many facets of life
- Solve more complex problems
- Manage rapid change in many dimensions
- Enable integration between systems
- Increased security
- Globalization – flat world
- Loose coupling and deferred commitment for increased agility

The evolution of computation models reflects the emerging evolution of businesses

Evolution addressed the need for increased abstraction and reduced coupling

- Functional Decomposition
- Abstract Data Types
- Object-Oriented
- Component Based Development
- SOA

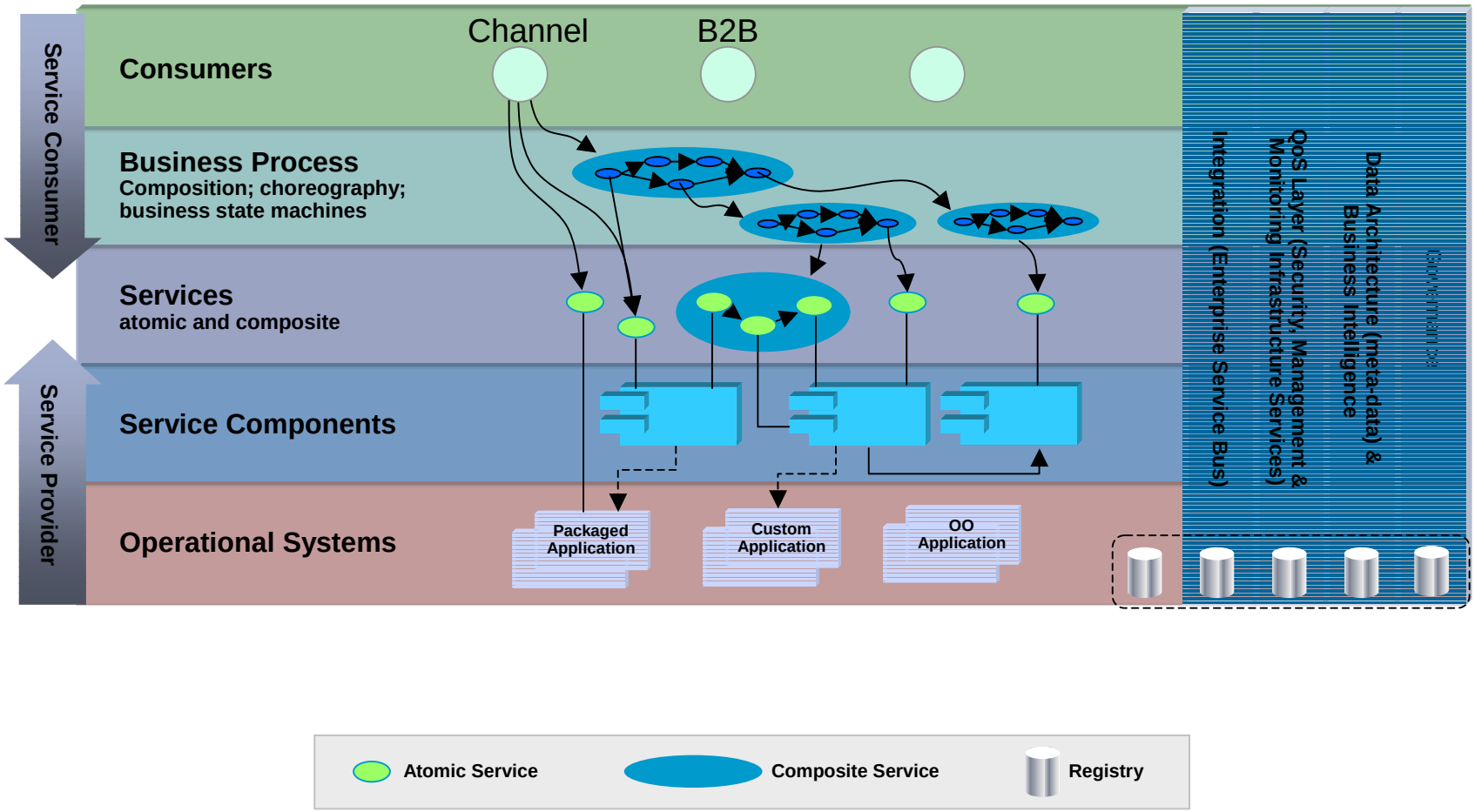
Component Based Development

- Separates component specification from realization
  - Clients only depend on the specification (interfaces)
  - Can substitute evolving realizations to fix bugs or add new features
  - Specification captures one set of concerns
  - Realization addresses those concerns while handling others
- Adds ports for better encapsulation and isolation
  - Better decoupling between requestors and providers
  - Component client only depends on what they need not the whole component
- Provides a better unit of reuse
  - Component is an autonomous entity
  - Specifies what it provided and what is necessary for its use
  - More formal support for commonality and variability

Service Oriented Architectures were introduced to:

- Addresses the effect of application integration across ownership boundaries
- Use Service Level Agreements to capture contracts
- Extends CBD with additional distributed computing and deployment concerns
- Provide more reflective and dynamic systems
  - Behavior can come and go
  - Clients query for service with acceptable QoS
  - Raise exceptions if none found
- Include concepts for publishing, finding, and dynamically binding to services
- Driven by practical implications of the Web and existing middleware platforms
  - Integration between J2EE and .Net

Moving to Services-Oriented Solutions



Why is an SOA (or any Architecture) needed?

- You have a lot of inter-related business processes
- Roles are responsible for a large number of tasks
- You have to rely on many services provided by others
- You need to support complex, automated business logic
- You need to address complex IT concerns such as distribution, persistence, integrity, security
- You need to build services that can be reused in workflow applications
- And there's a lot of variability in the processes and tasks because of different market segments and channels
- You need high performance, high security, high availability

Why aren't executable business processes enough?

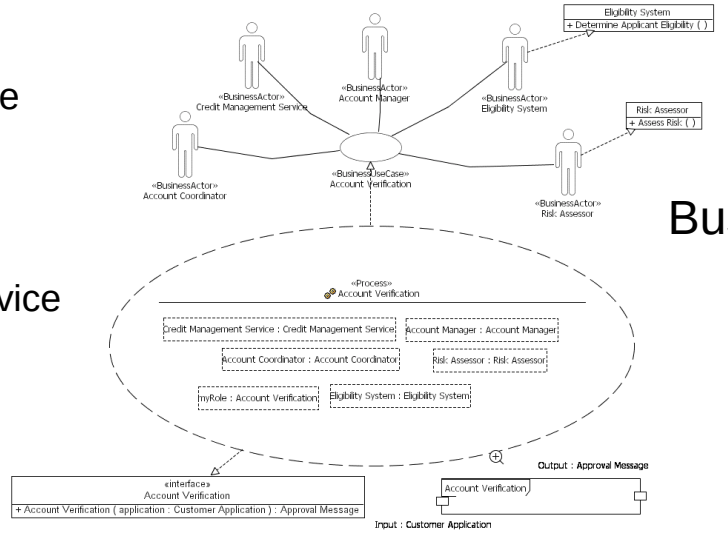
- Good Business processes aren't necessarily good SOA solutions
- SOA models are not Business Process models
- Decouple business analysis from IT SOA solution
- Primary value proposition for SOA is flexibility achieved through separation of concerns, loose coupling and late binding
- This is what Services modeling is about – its different than process modeling
- SOA modeling for solution, not just functional decomposition for addressing both business and IT requirements
- May result in significant refactoring of business processes that express SOA requirements

But what if your business analysts gave you a bunch of process models?

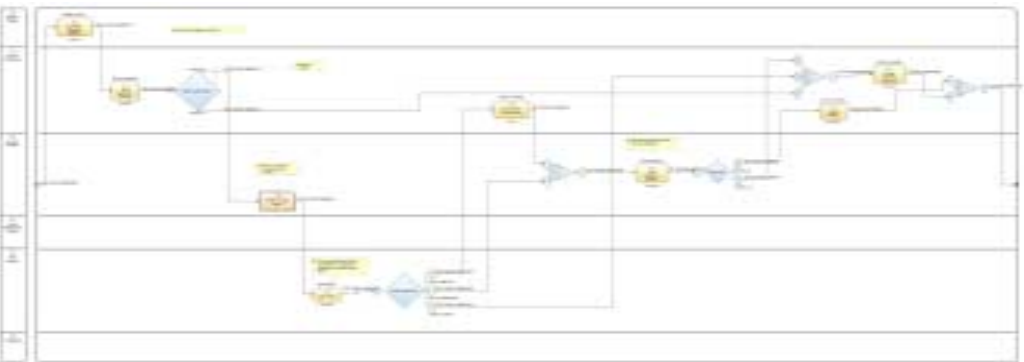
- What do you do with them?
- How do your SOA solutions relate to the business processes?
- How do you guarantee your solution meets the original business objectives?
- What do you do if the business analyst changes the process?
- How can you tell what parts of the SOA solution might need to be updated?
- Is J2EE (or code) your only option?
- What if you're anticipating a significant update to your runtime platform?
- How can you isolate your applications from changes in the middleware?

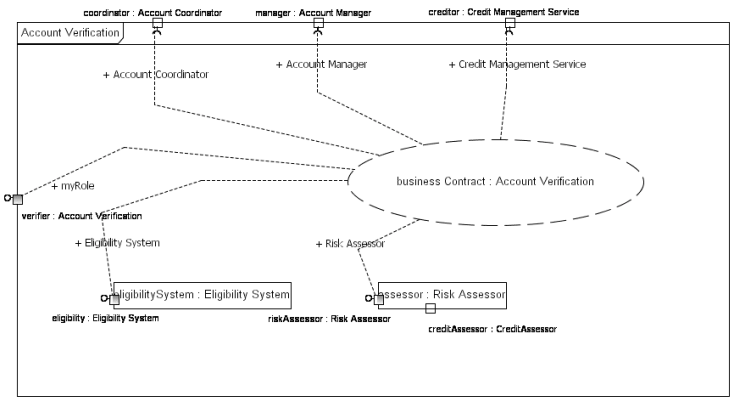
BDD provides a way to bridge between business processes and SOA solutions – Architecture is Key

- Business process as service contract
- Role-based view
- Role responsibilities across processes
- Roles suggest possible service providers



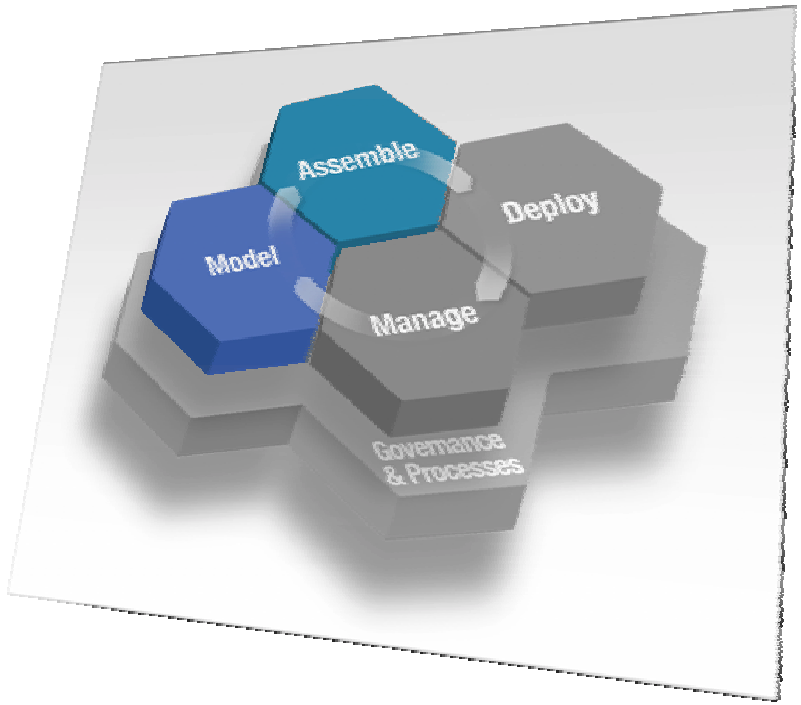
Business Services Contract



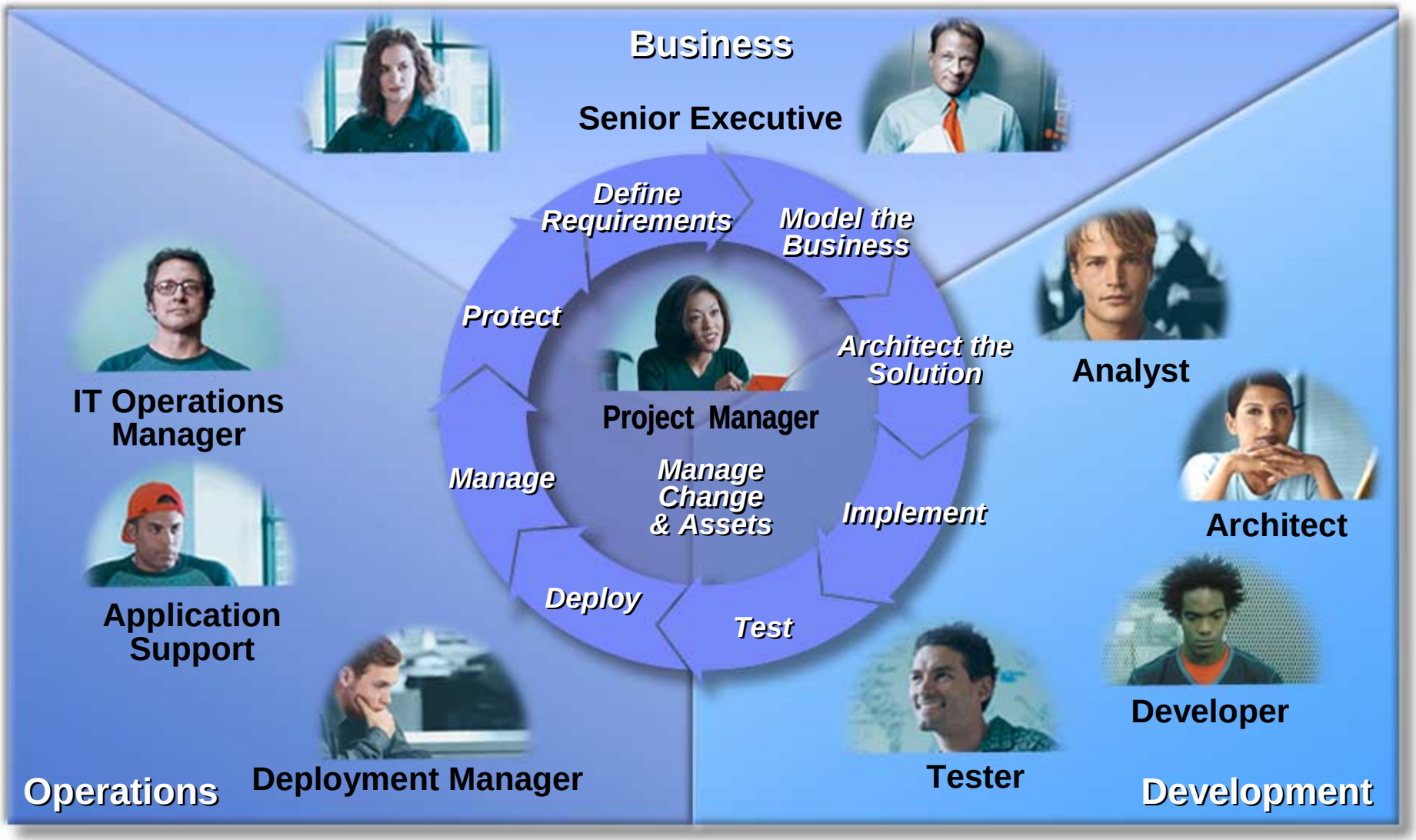


Agenda

- 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

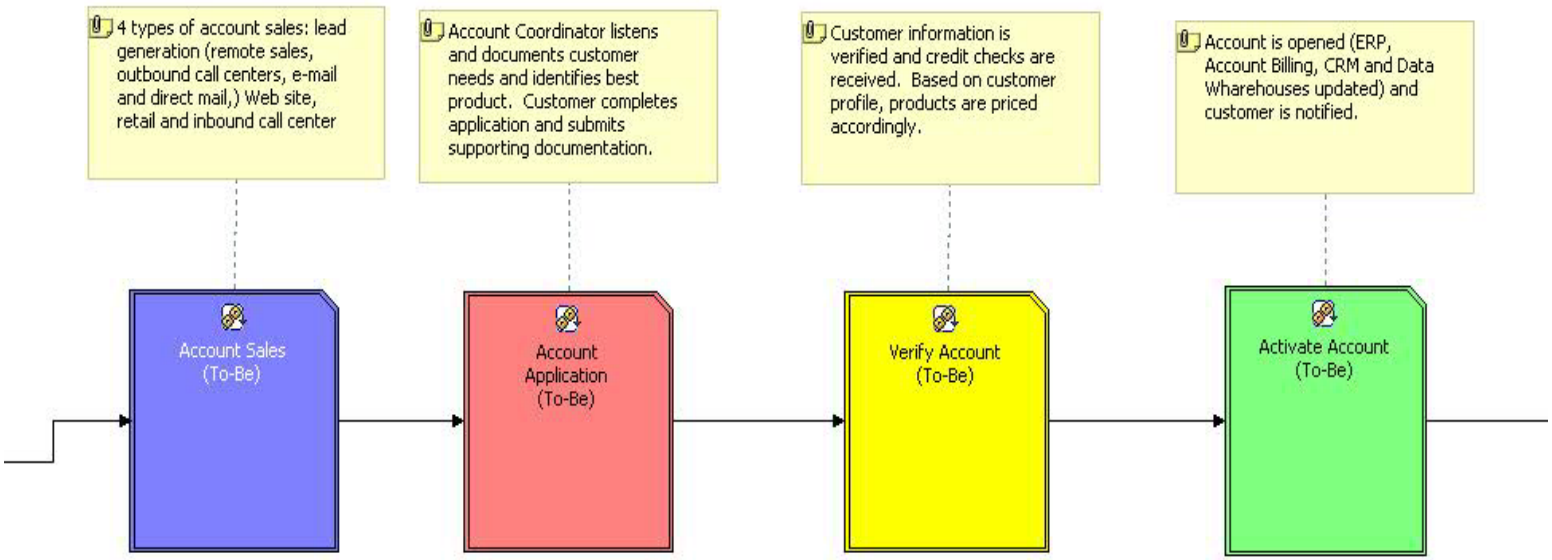
Rational Portfolio Manager

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



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 (re)entering information in various internal systems (CRM, ERP, etc.)

Account Open – Business Goals

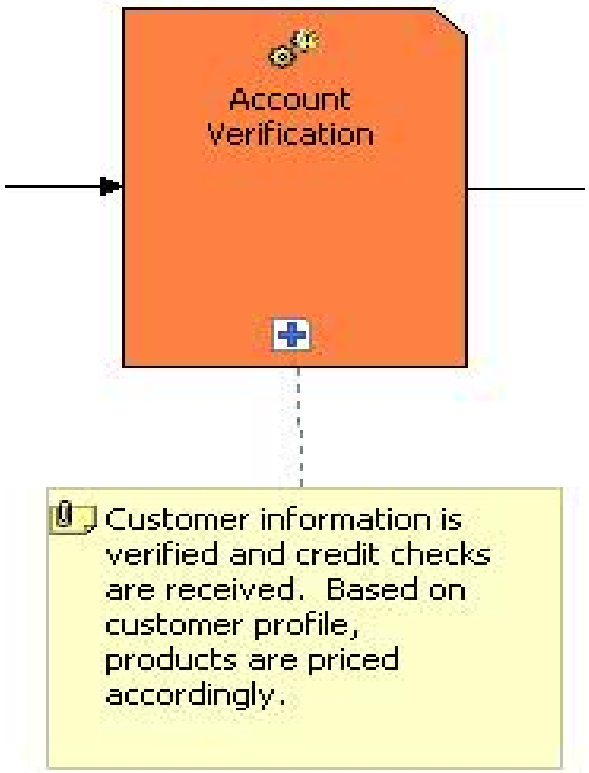
- Strategic Initiative - Optimize Account Setup
  - Reduce Cost and Latency – grow efficiently and profitably
  - Improve sales and customer service through increased speed and responsiveness
  - Enhance productivity through reductions in total cost of ownership (TCO)
  - Increase the value and reach 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

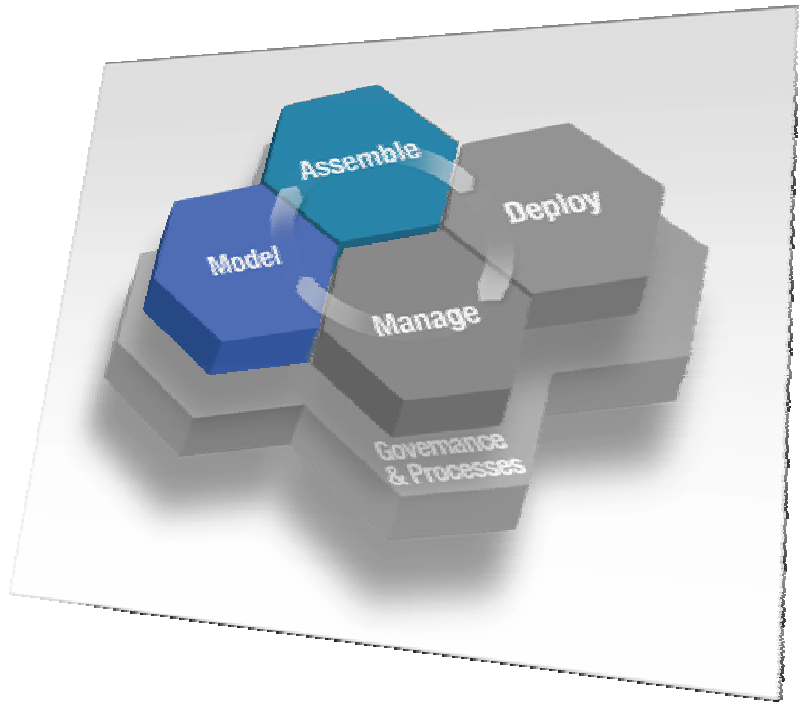
Account Open – Account Verification Process



- 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

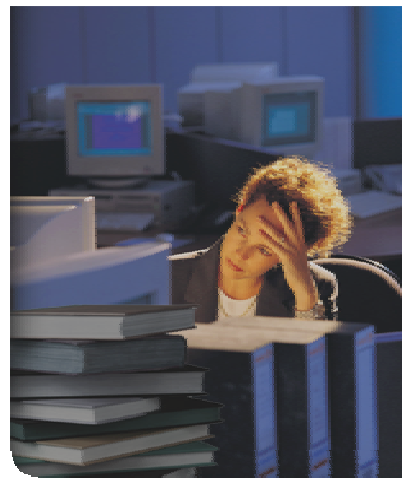
Agenda

- 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

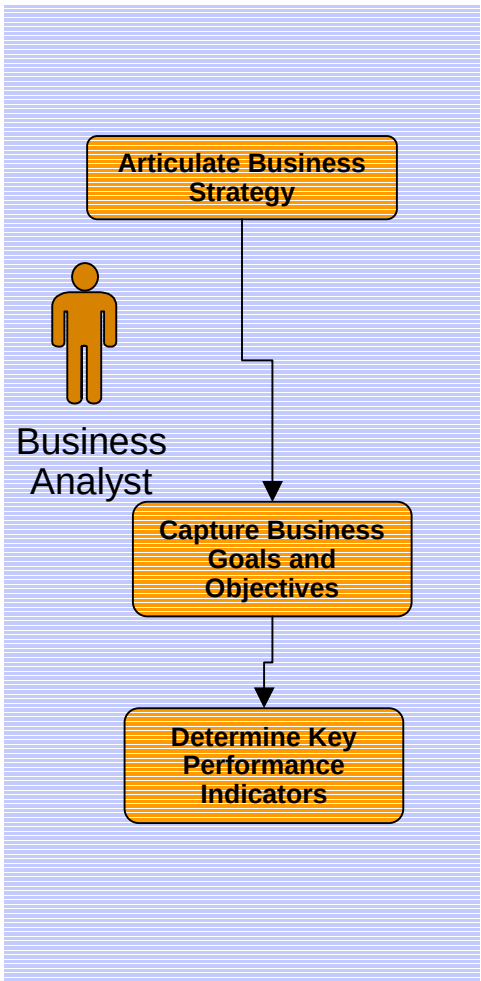
- **Understanding and using requirements**
  - **Know what you want to accomplish and how to measure success**
  - Centrally manage requirements located in many documents, charts, and models
  - Provide context for business value
  - Make available to designers, developers and testers
- **Organizing 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

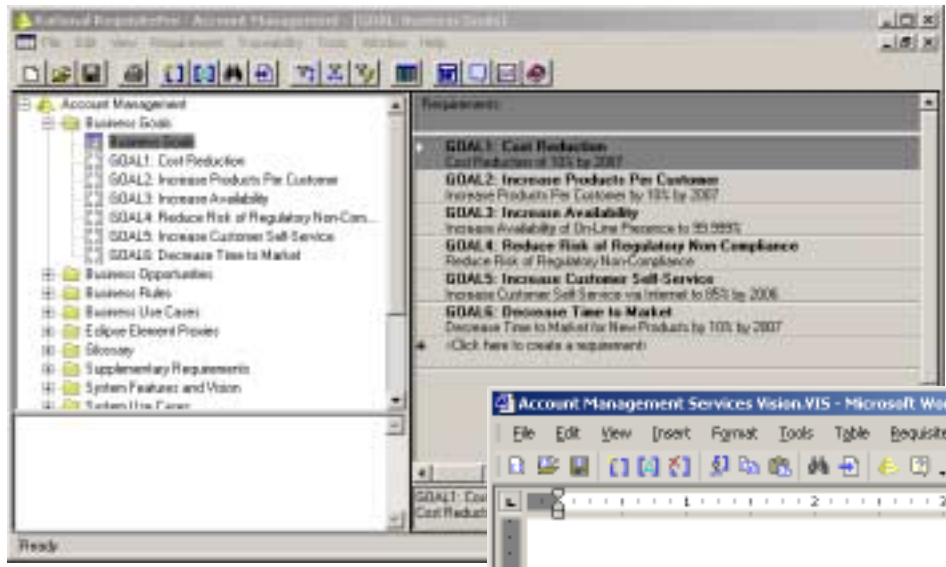


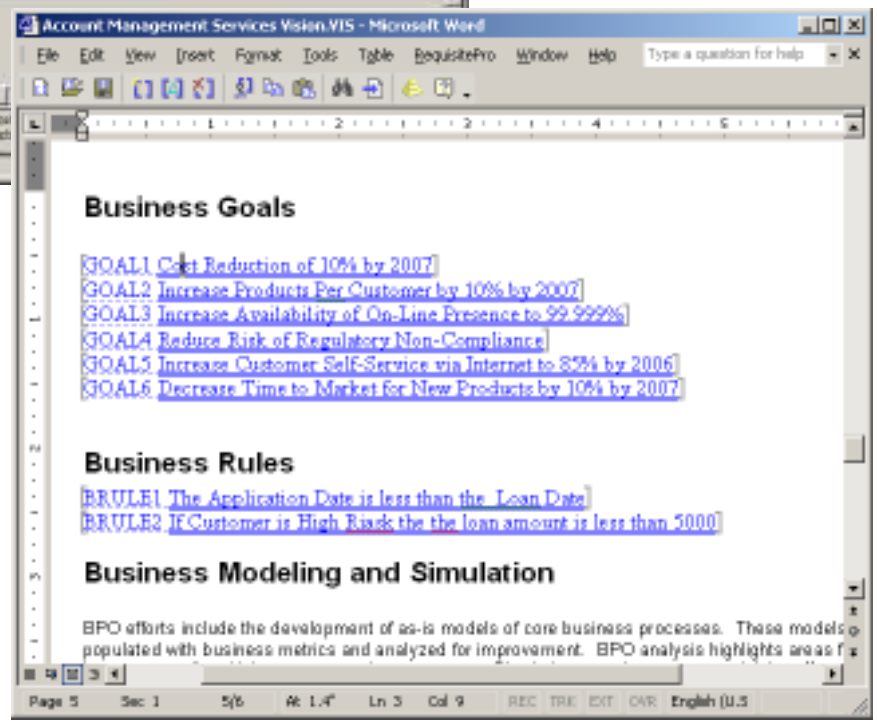
Unsolved RM Challenges ➔ Software Rework ➔ Cost, Delays, Quality Issues

Capture Business Goals and Objectives and KPIs

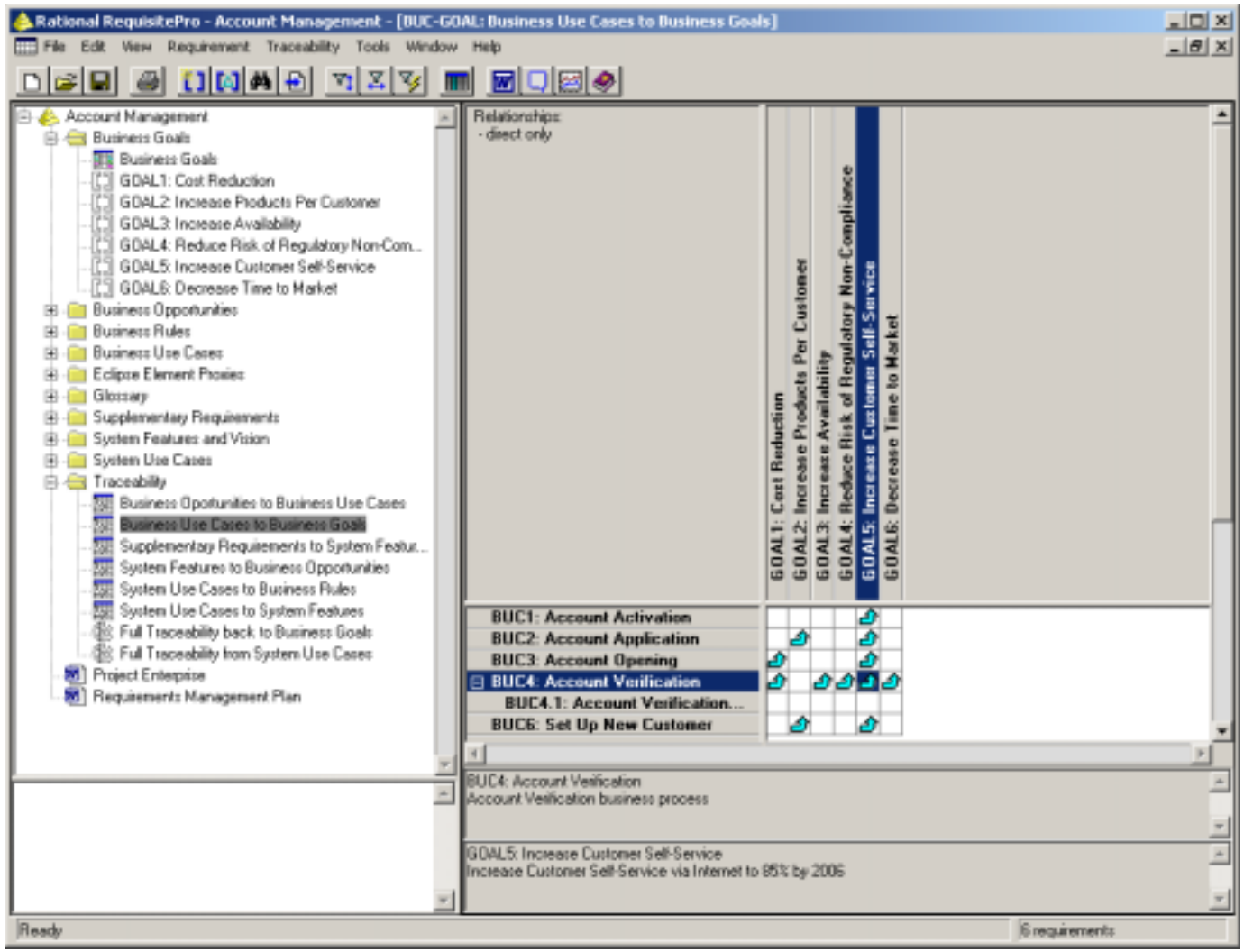
Rational RequisitePro



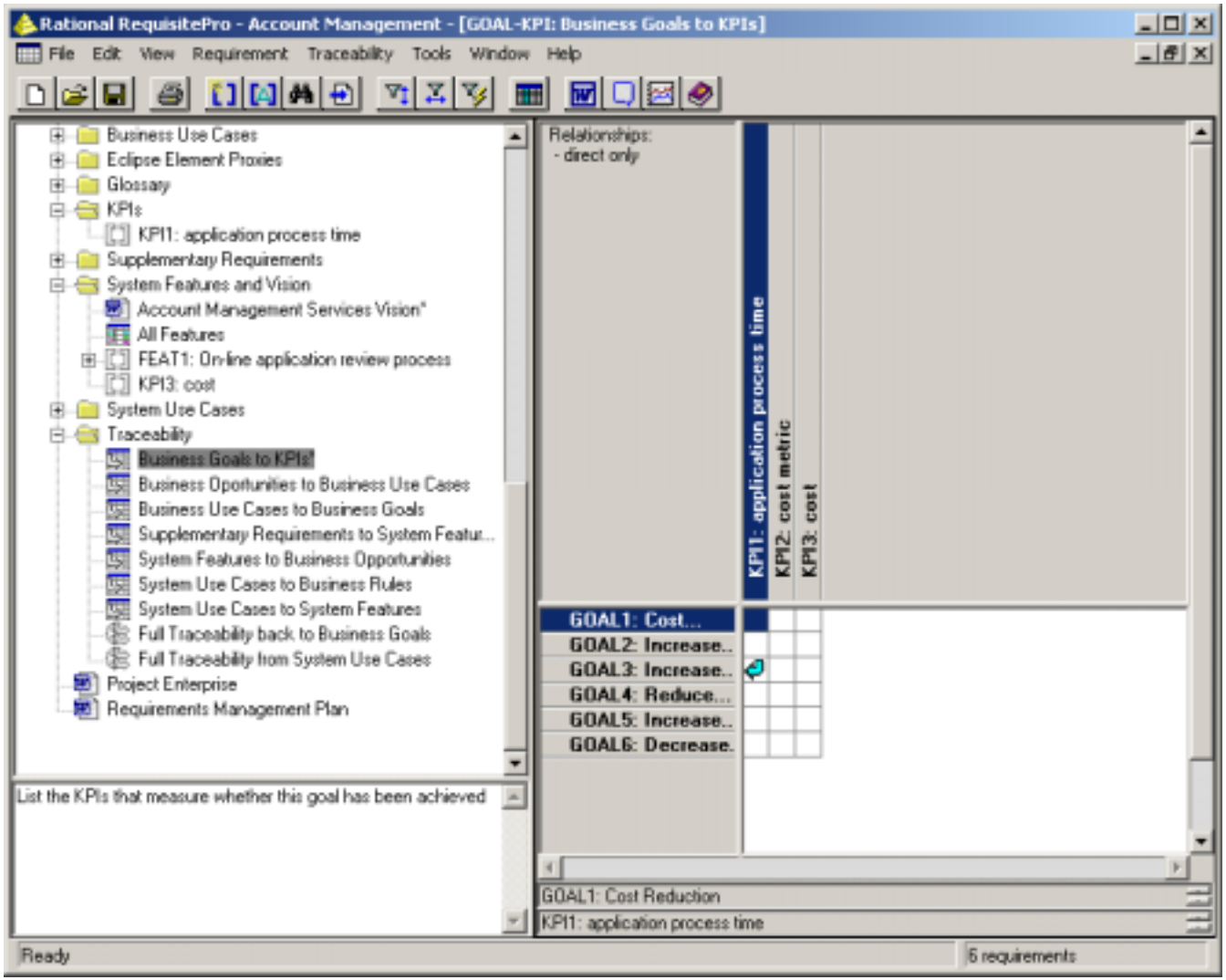




Trace Business Use Cases to Business Goals they Realize

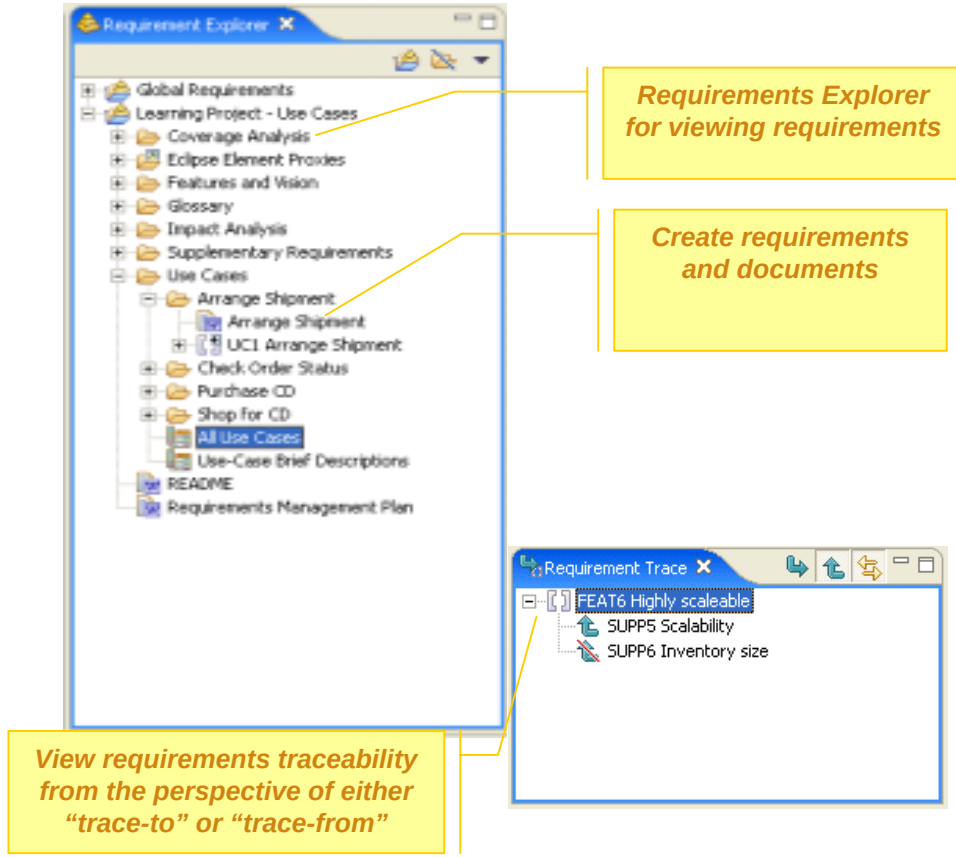


Trace Business Goals to KPIs that measure them



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

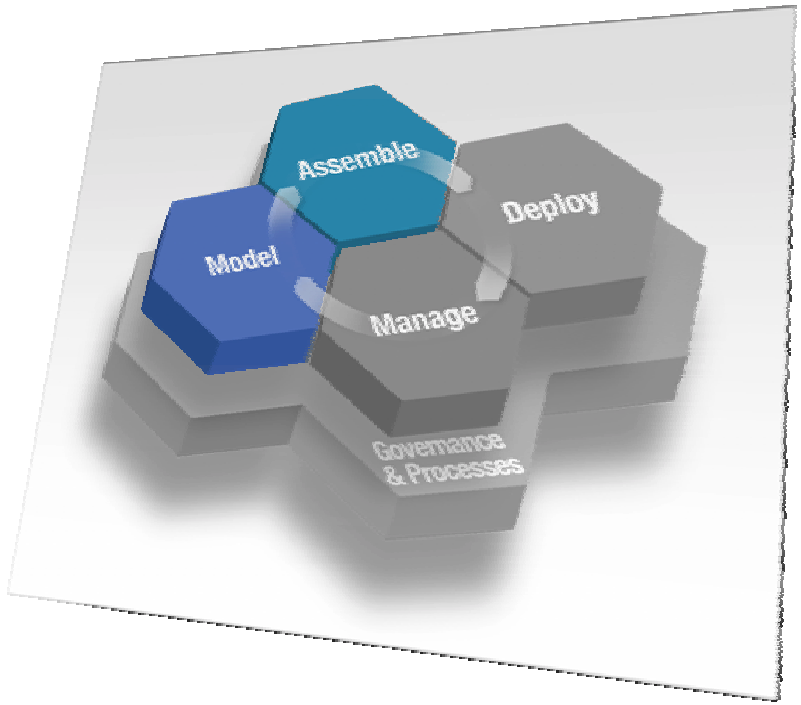


Customer Benefit:

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

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.

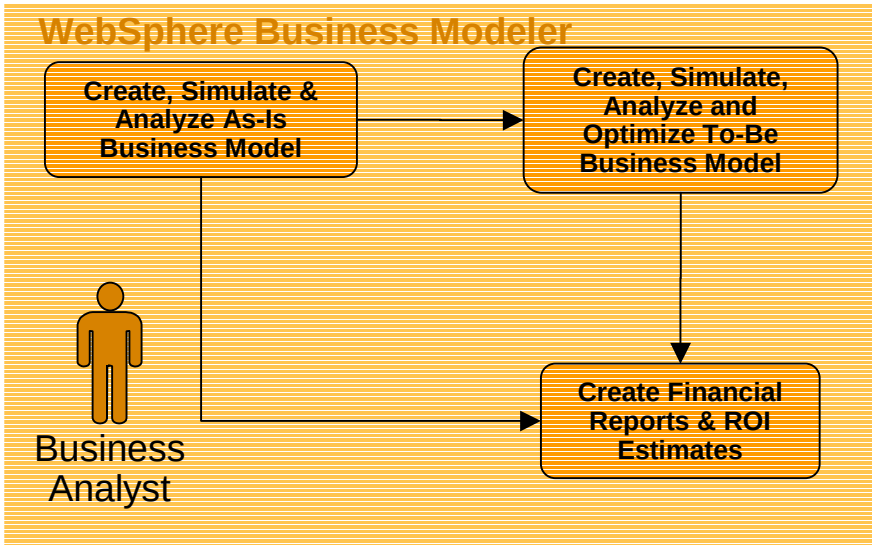
Bob assesses the service automation opportunities within JK Enterprise's existing Account Application and Account Verification process. Bob runs a simulation of the process using data provided by the BPO team. He finds that human-based tasks are the most expensive and the decline case is the most expensive 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
- Simulate TO-BE processes and verify business objectives are being met
- Creates new business opportunities
- Indicate the process enhancement work item is complete

Model Business Processes to realize goals and objectives

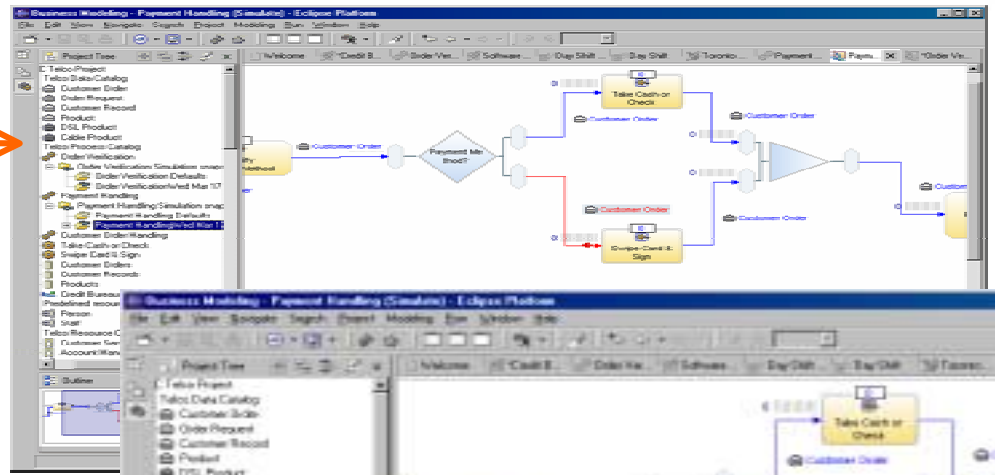


- Goal centered modeling
- Not concerned with IT at this point
- Formally describe current business
- Discover and design key business processes
- Capture business data items exchanged between processes & tasks
- Assign tasks to roles that are responsible for their performance
- Determine and allocate 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

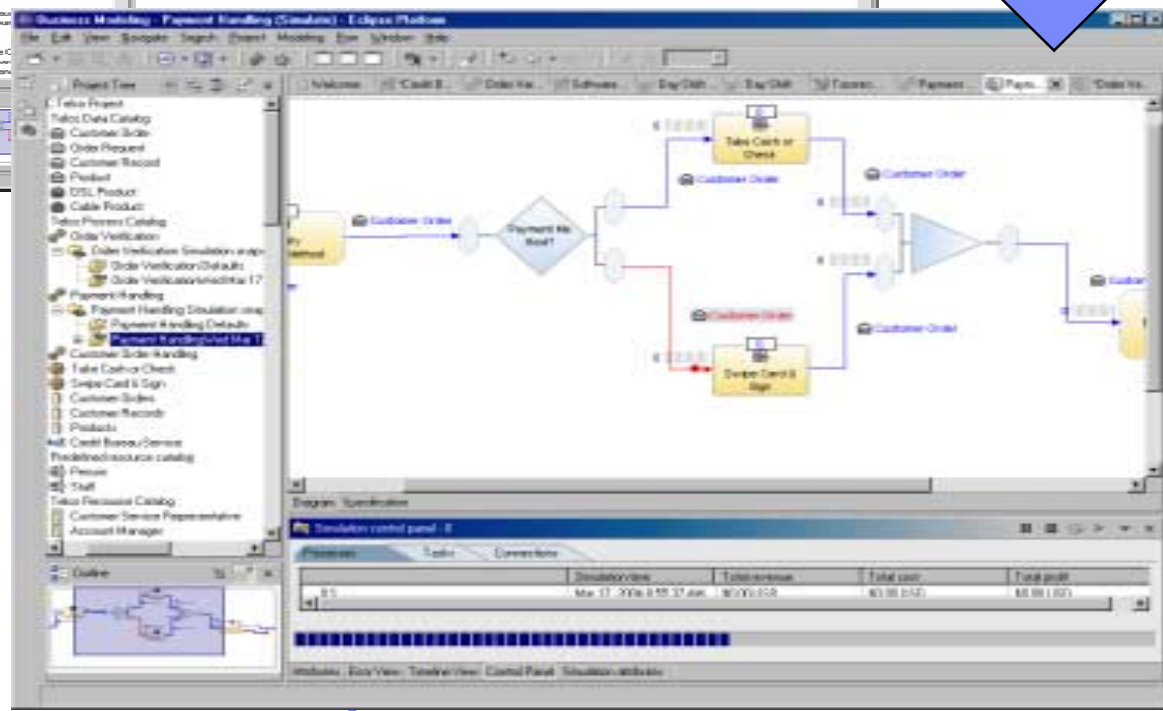
Model the business Operational Requirements

Business processes that realize goals and objectives

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

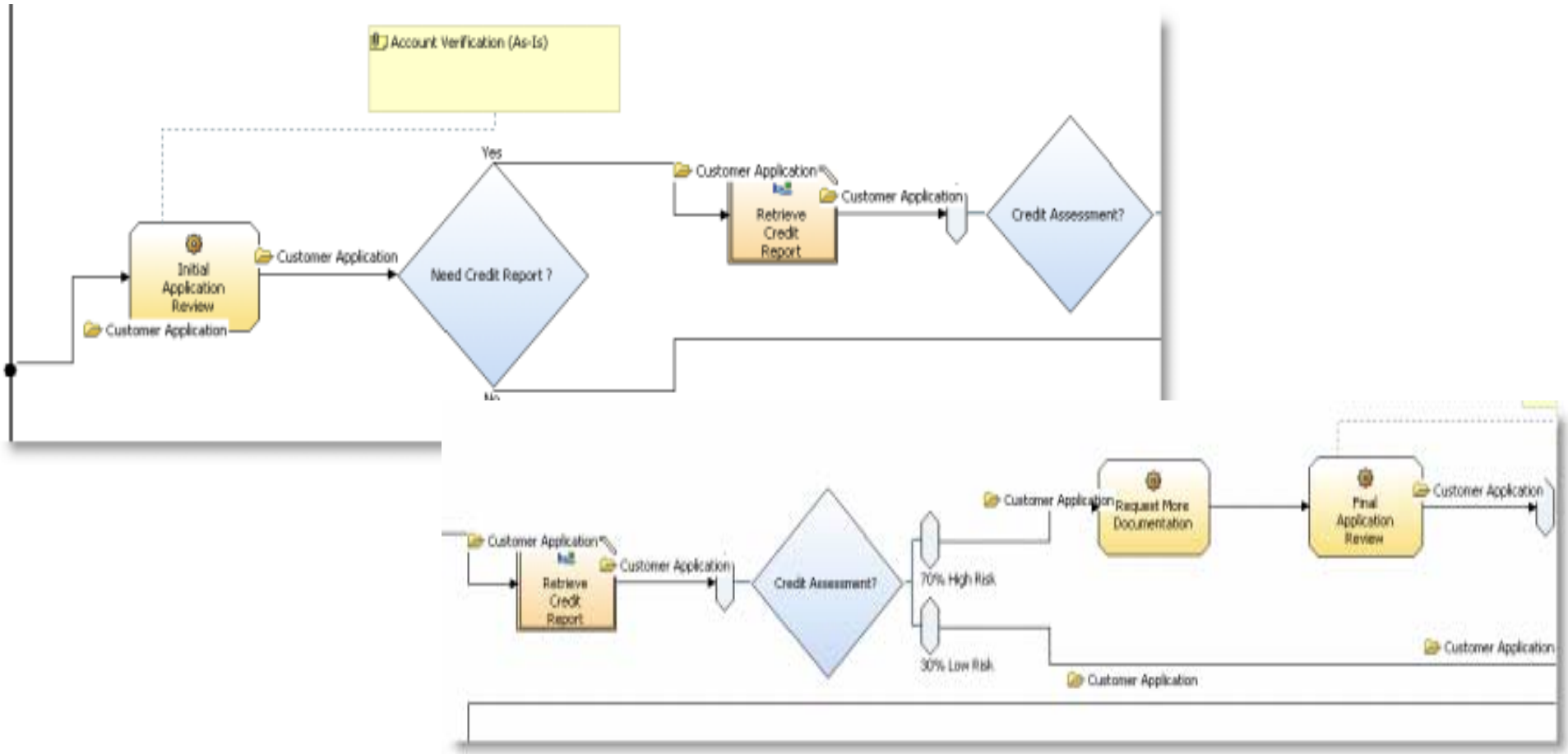


Verify "to be" business processes through simulation

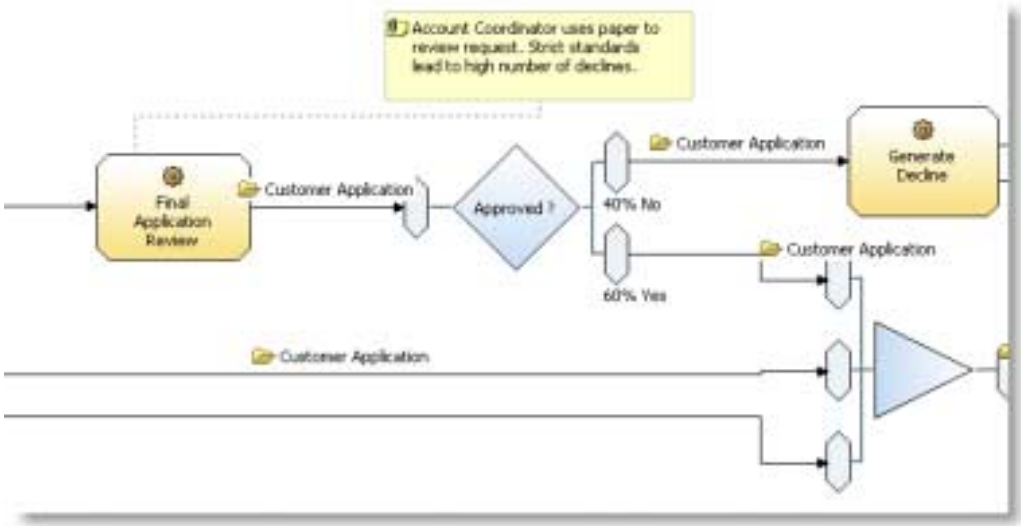


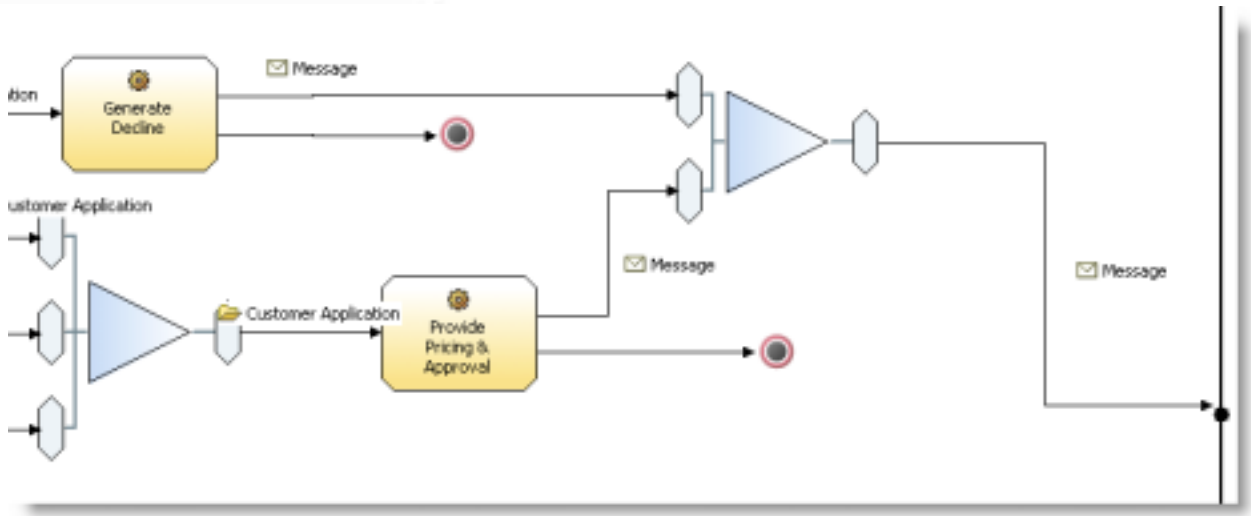
IBM WebSphere Business Modeler

Examine existing scenarios for as-is business processes

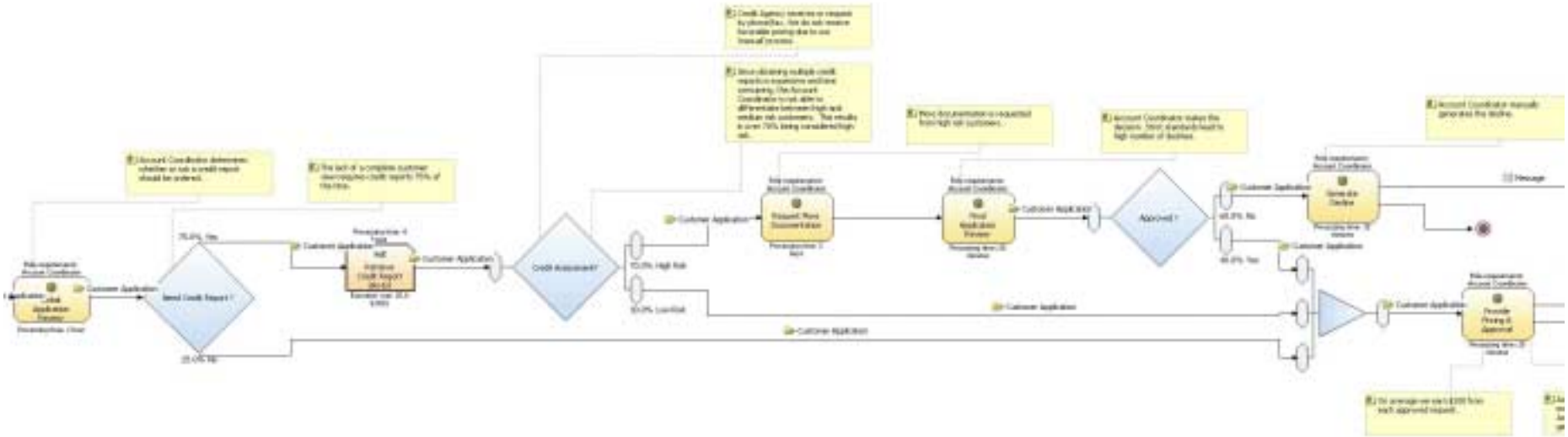


Account Verification (As-Is) Business Process (cont)





Simulate as-is Business Process



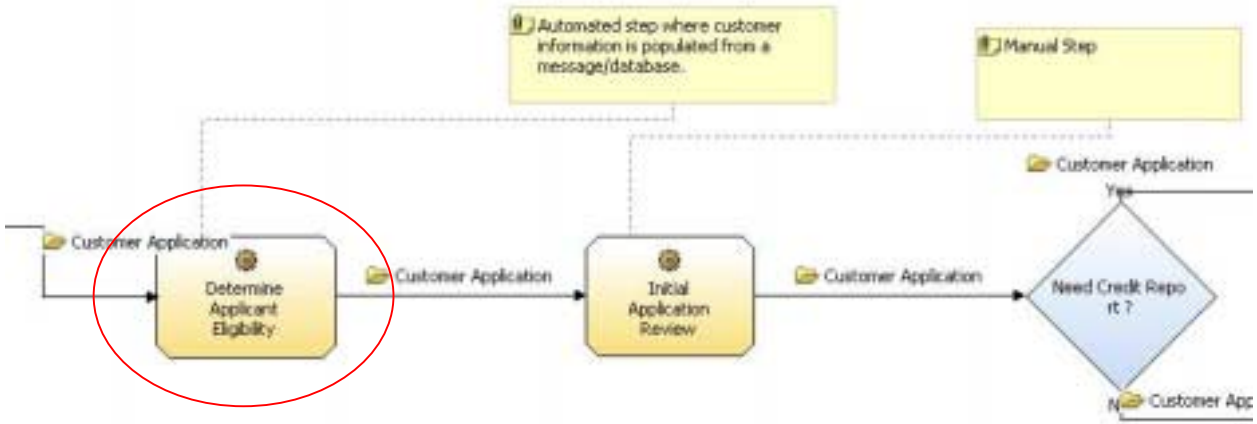
Case Name	Distribution	Success Status	Average Elapsed Duration	Average Throughput
Case 1	24%	Succeeded	2 hours 13 minutes 18.5 seconds	0.450084 work item / hour
Case 2	22%	Succeeded	7 hours 41 minutes 12.181 seconds	0.130095 work item / hour
Case 3	22%	Succeeded	3 days 11 hours 30 minutes 24 seconds	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

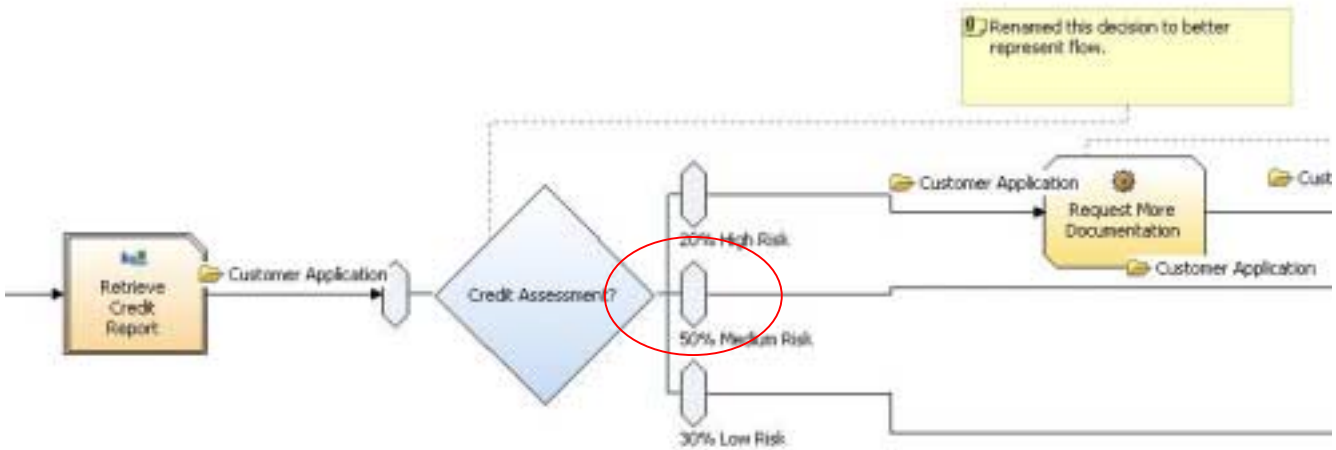
Re-engineer business processes to realize goals



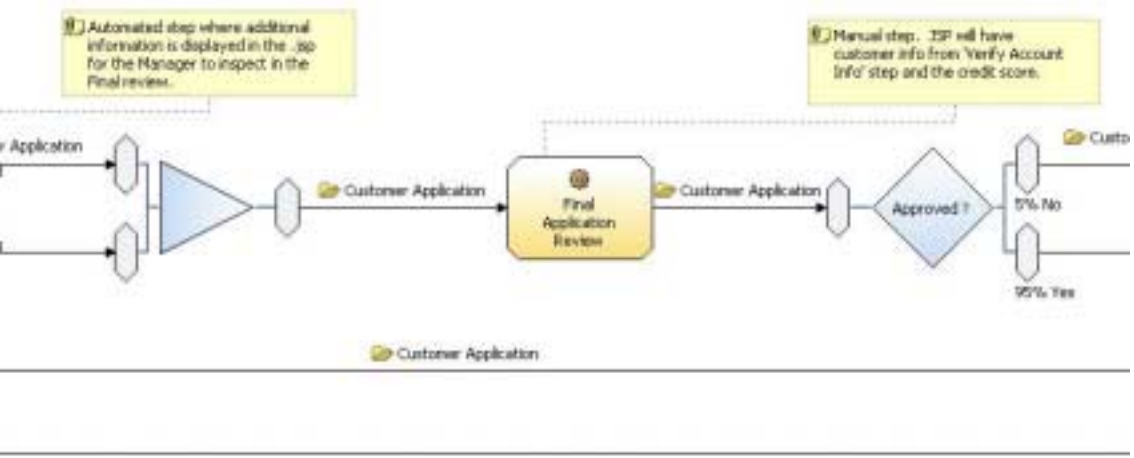
Shorter Process Duration

More Automation

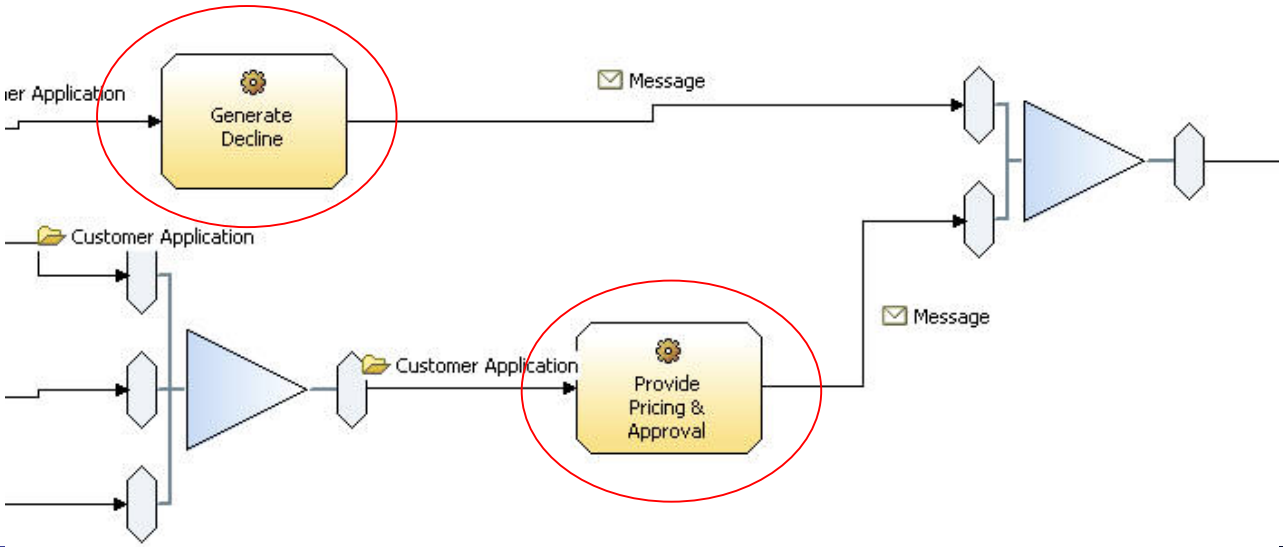
Lower Process Cost



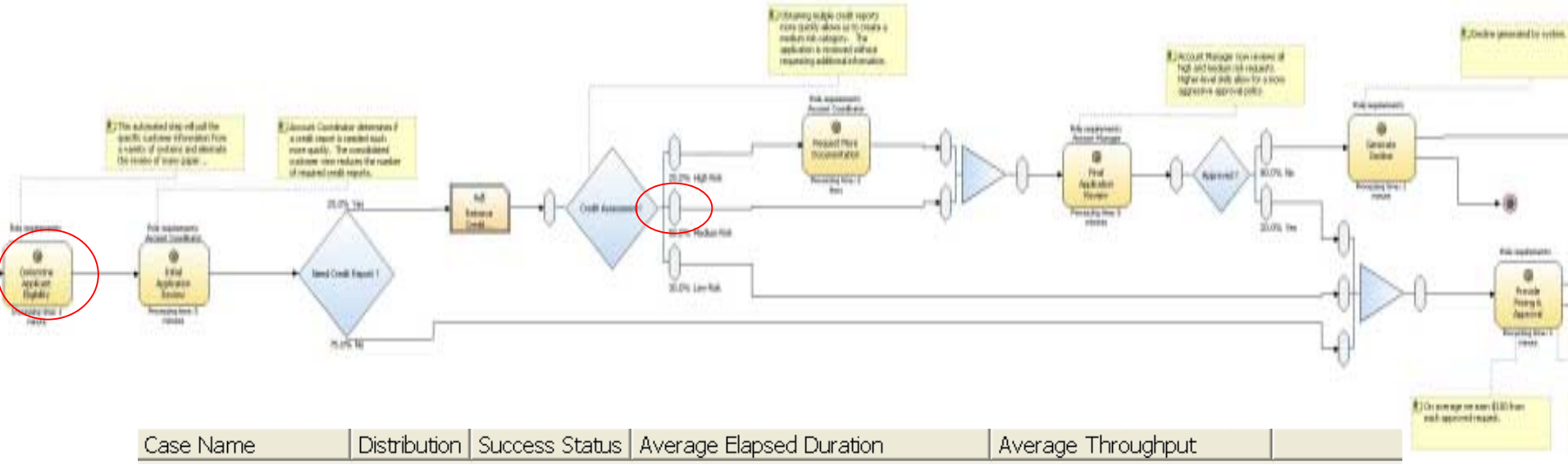
Verify Account (To-Be) Business Process (cont)



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

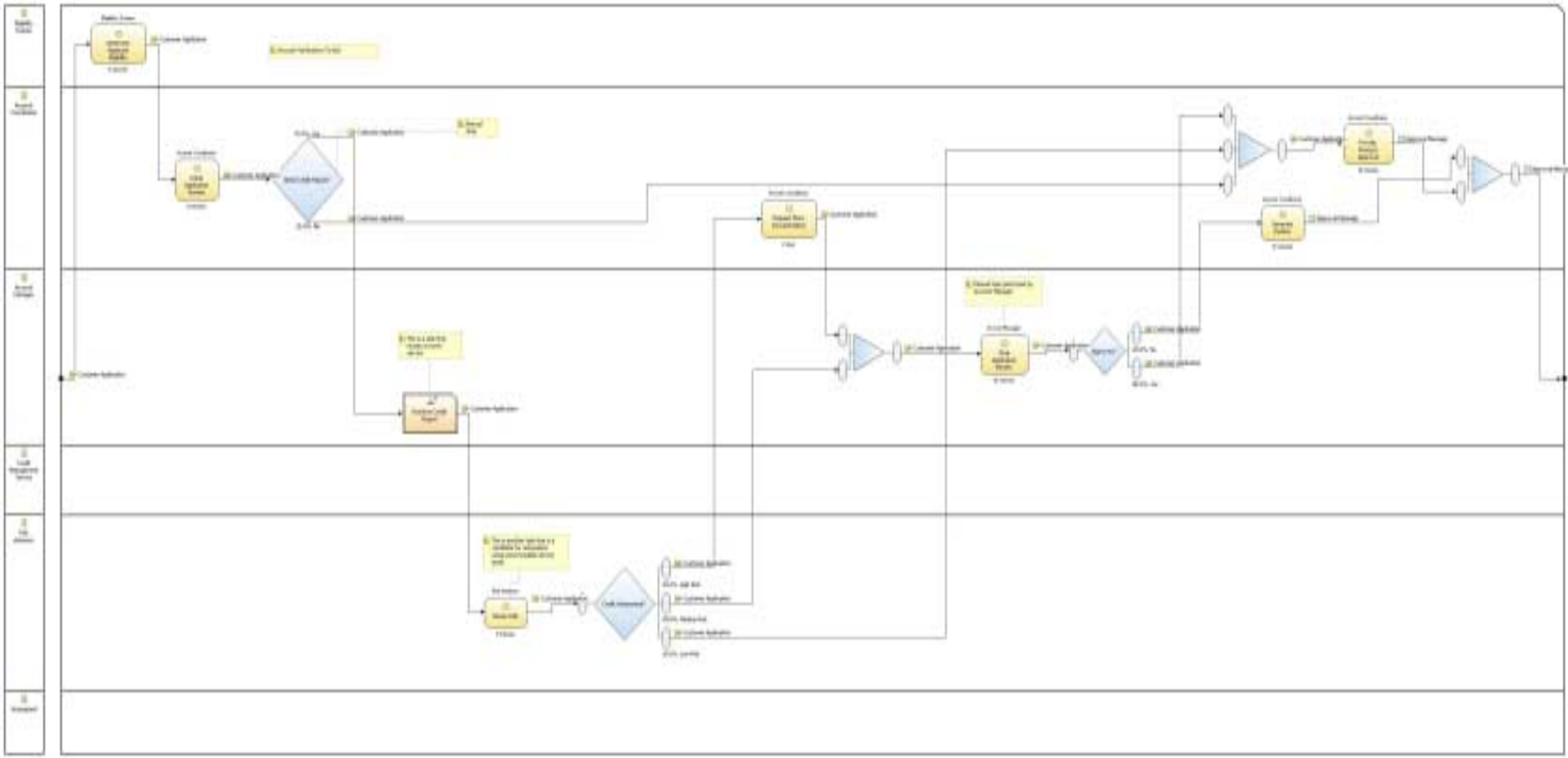
Compare results and see if objectives are met

	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%

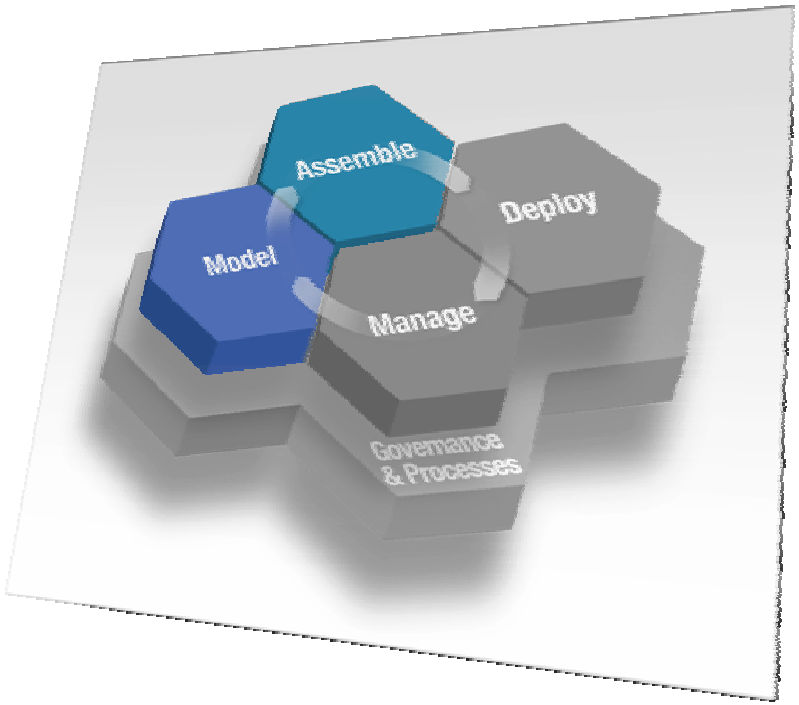
- Automated and more complete customer view reduces number of customers needed credit reports.
- Automated credit report is significantly less expensive and faster.
- Approving larger percentage of customer requests.

Final, to-be Verify Account business process



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 detail for an IT solution.

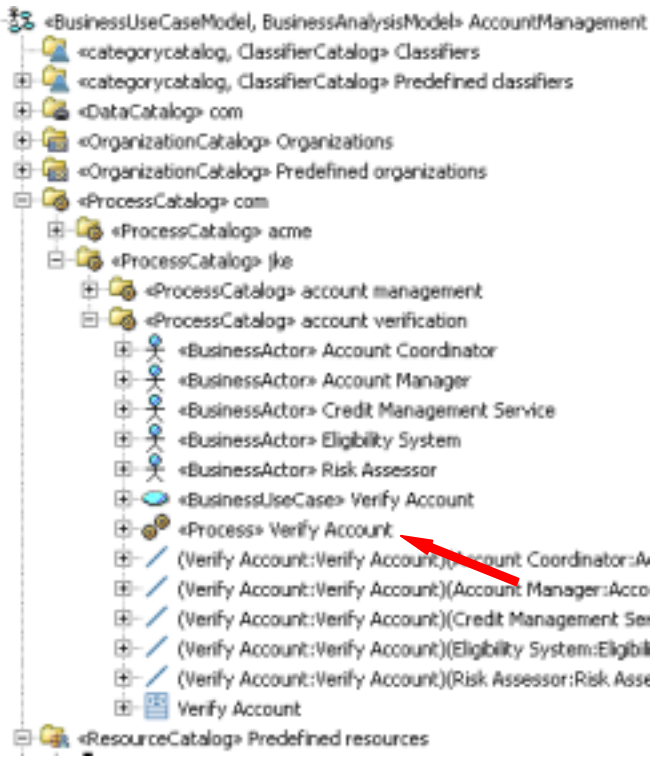
Once the business requirements are known, Architect Al can elaborate the business processes to ensure the business requirements are sufficiently specified that services designs can be created. Al is now ready to begin architecting a services solution to realize the business requirements in a manner that 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.

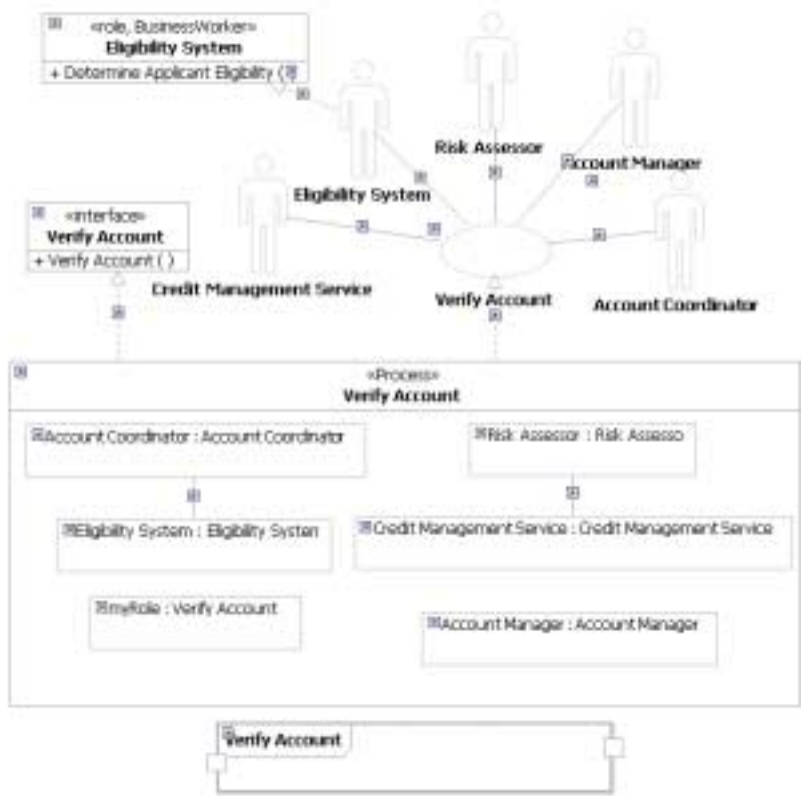
Al must look across a number of possibly complex processes with overlapping roles that may need to be significantly refactored for an effective, maintainable services solution. Al must also deal with IT concerns such as availability, distribution, integrity, security and persistence as well as the nonfunctional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

- **Business analysts focuses on tasks that have to be performed to meet a business objective**
- **They don't worry about cohesion and coupling, data interchange, reuse or many other IT concerns**
- **Some business processes can be executed directly as workflow applications**
- **But many require architected solutions in order to address IT concerns**
- **Architected solutions must relate somehow to the original business requirements**
- **This is needed for verification, validation, traceability and change management**

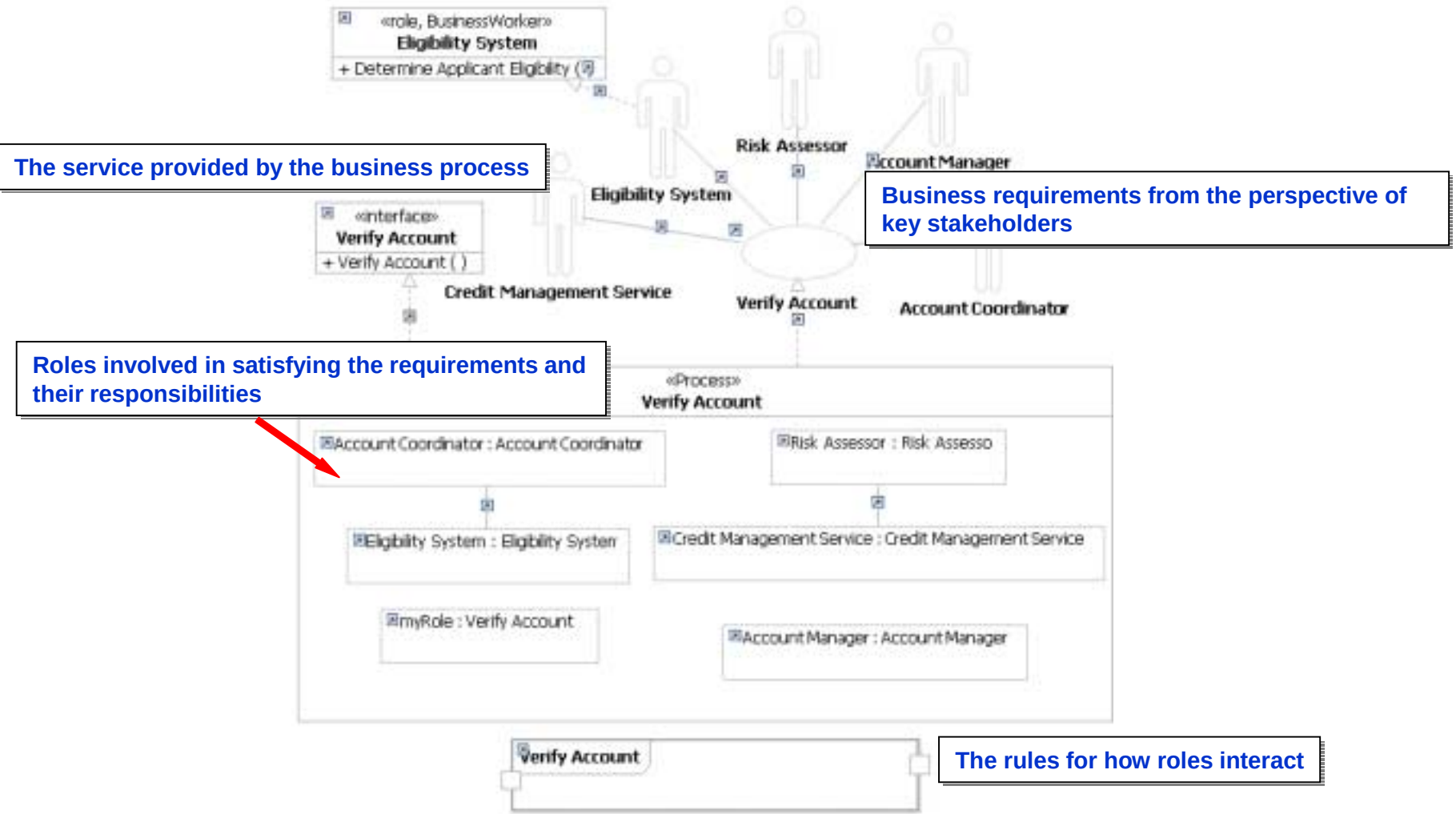
Explore Business Requirements

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.





The Business Services Contract



The service provided by the business process

Business requirements from the perspective of key stakeholders

Roles involved in satisfying the requirements and their responsibilities

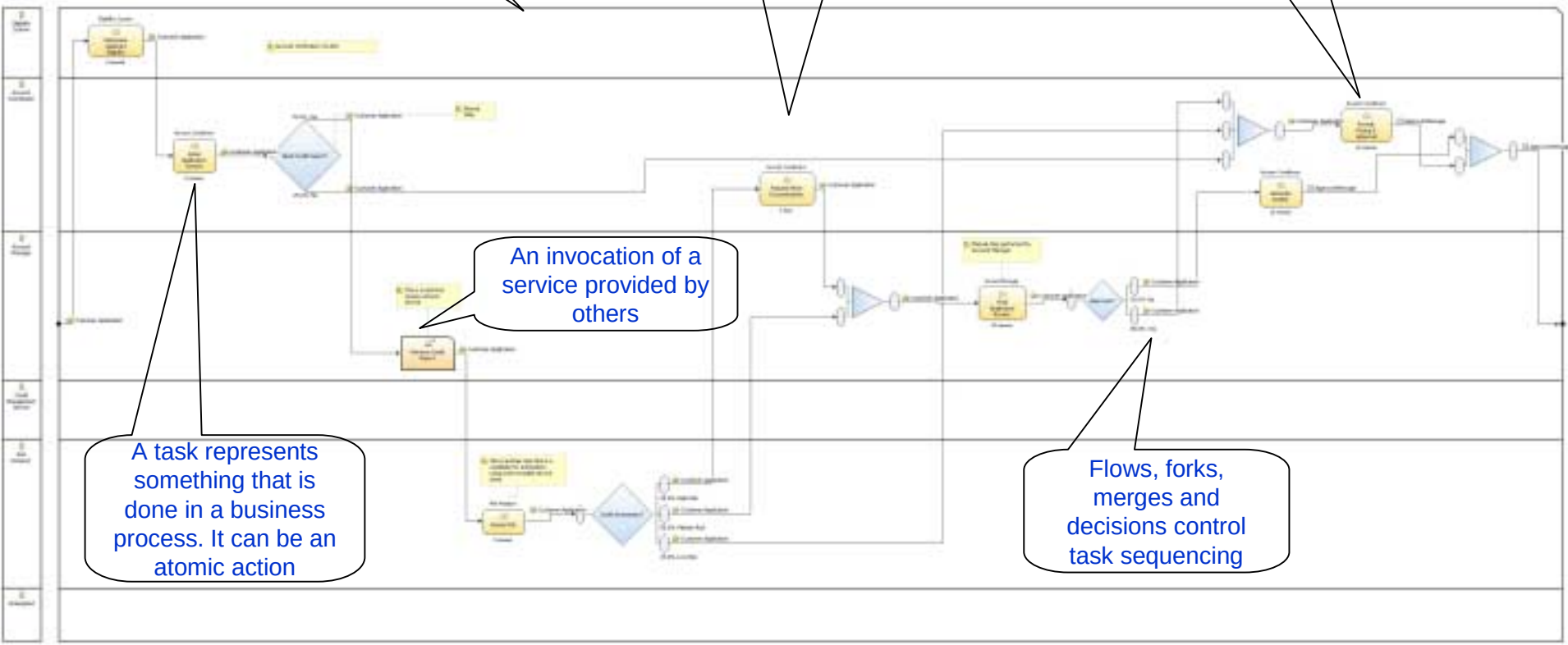
The rules for how roles interact

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

The invocation of another process

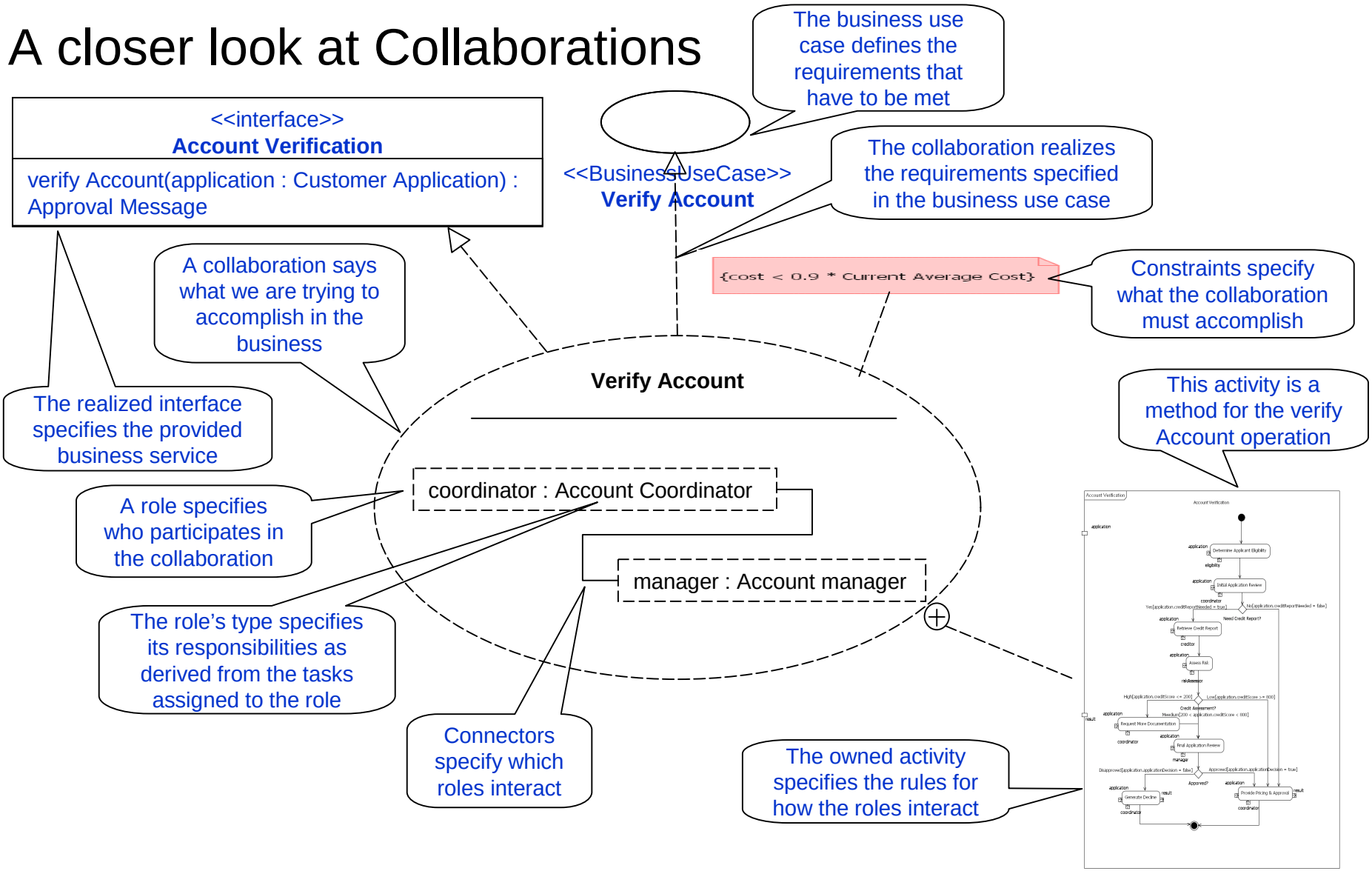


An invocation of a service provided by others

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

Flows, forks, merges and decisions control task sequencing

A closer look at Collaborations

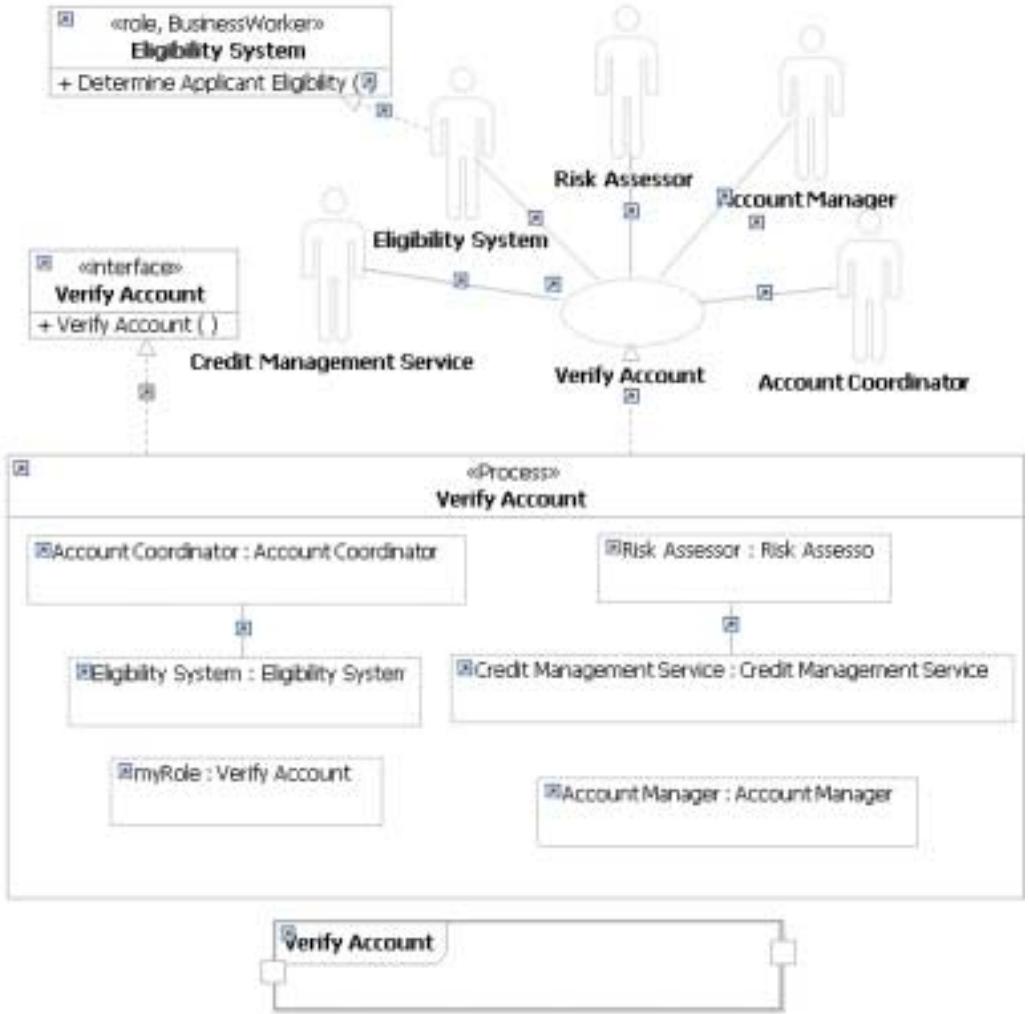


UML2 Collaborations can be used for a number of purposes

- Multiple views of systems
  - A way of abstracting different aspects of services solutions
  - Convey information to stakeholders and users of those services
  - Highlight different subject areas or concerns
- Specify service requirements
  - Without constraining the architecture for how those requirements might be realized
- Bridge between business process models and solutions
  - Separates the what from the how
  - Formal link between service implementation and the contracts it fulfills
  - More than just traceability

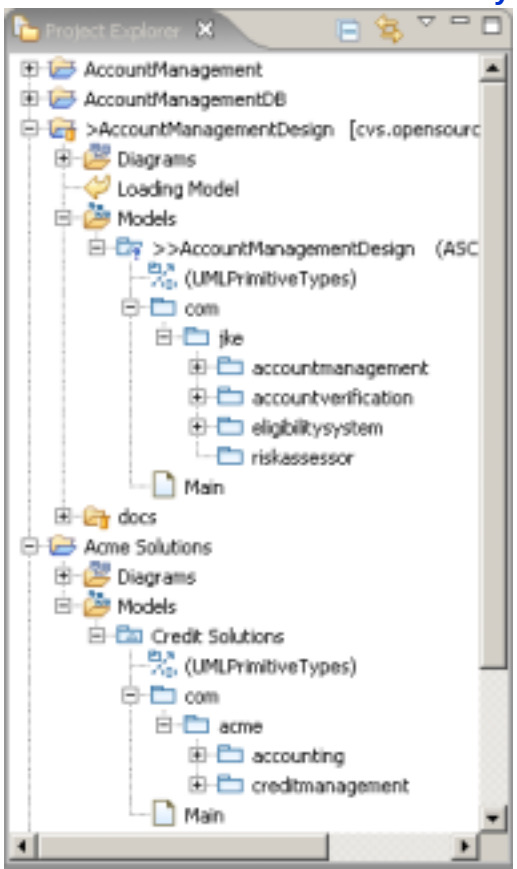
Design the Service Realization Architecture

- Recall the Business services Contract
- We can use it to help guide the services solution design
- We'll design the services by finding, building or adapting something capable of playing each of these roles

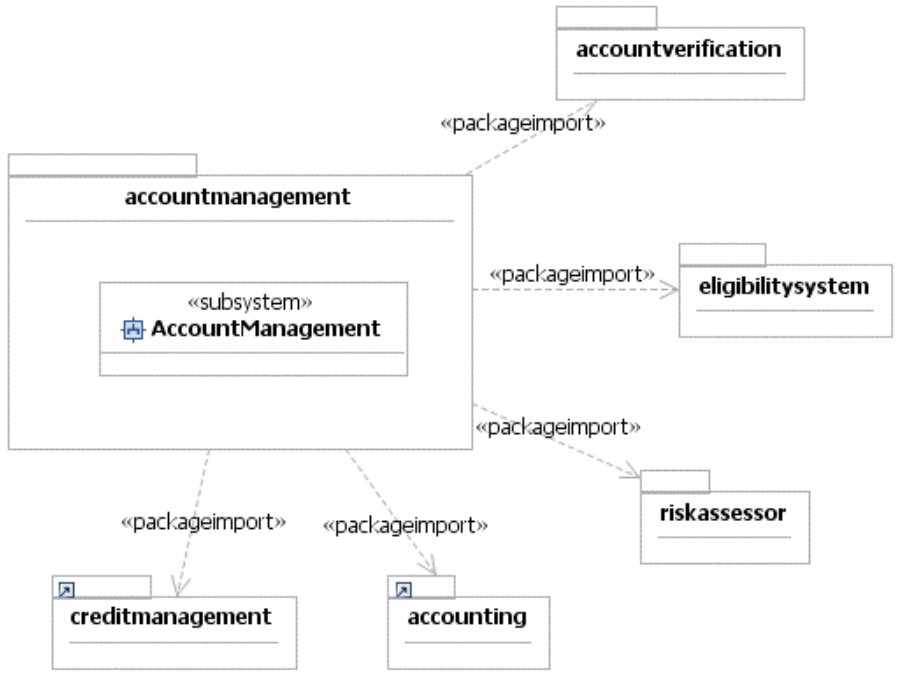


Design the business functional areas and solution result

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 initial 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.

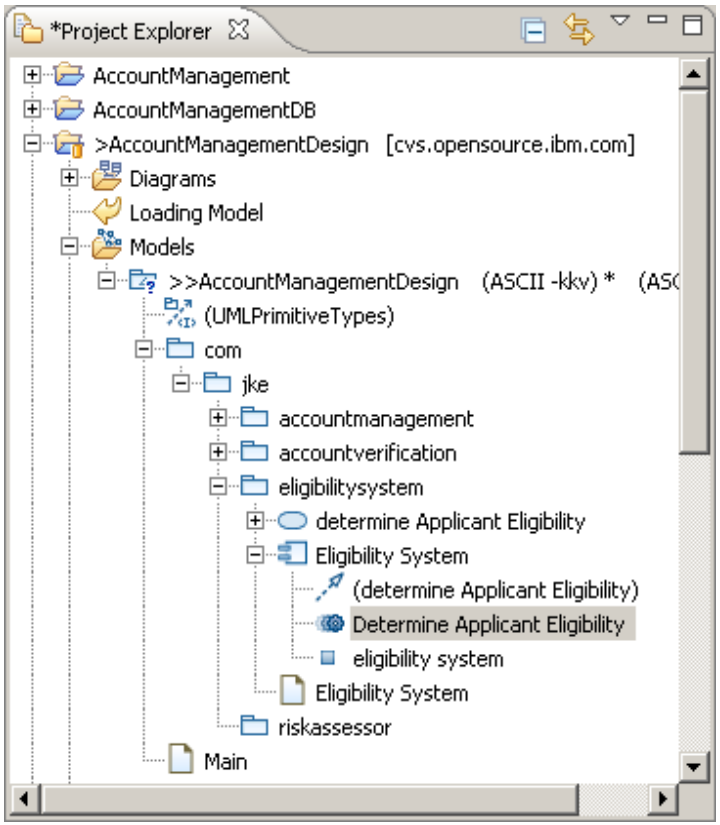


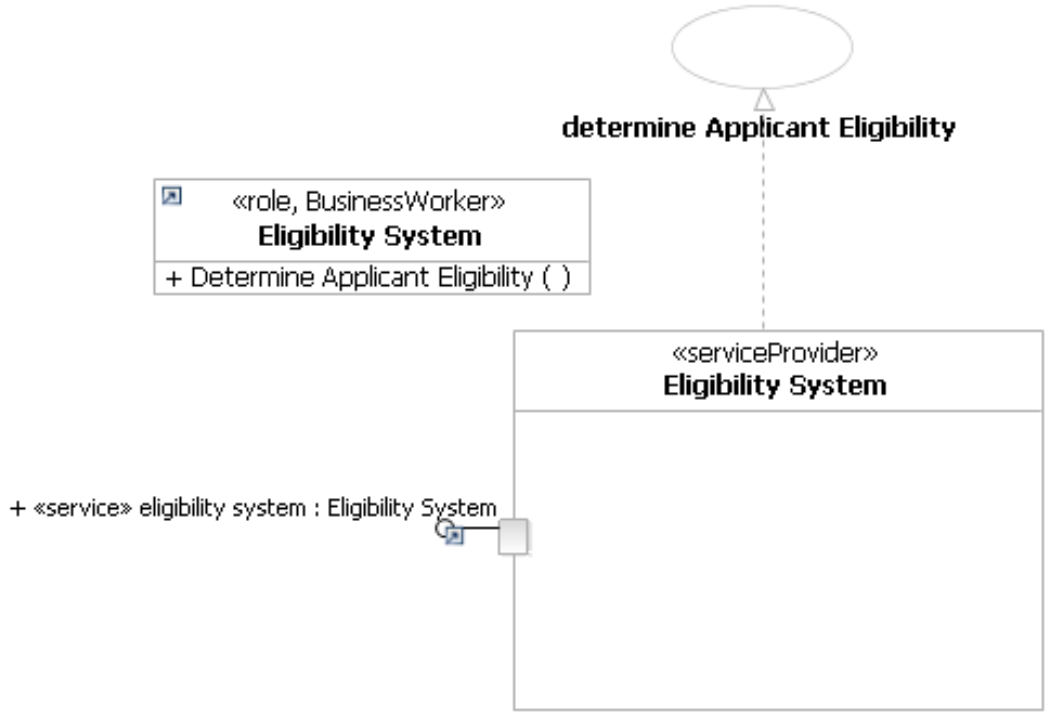
Start with separating concerns and packaging to manage complexity and facilitate reuse



Create The Eligibility System 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.





Develop the Business Domain Model

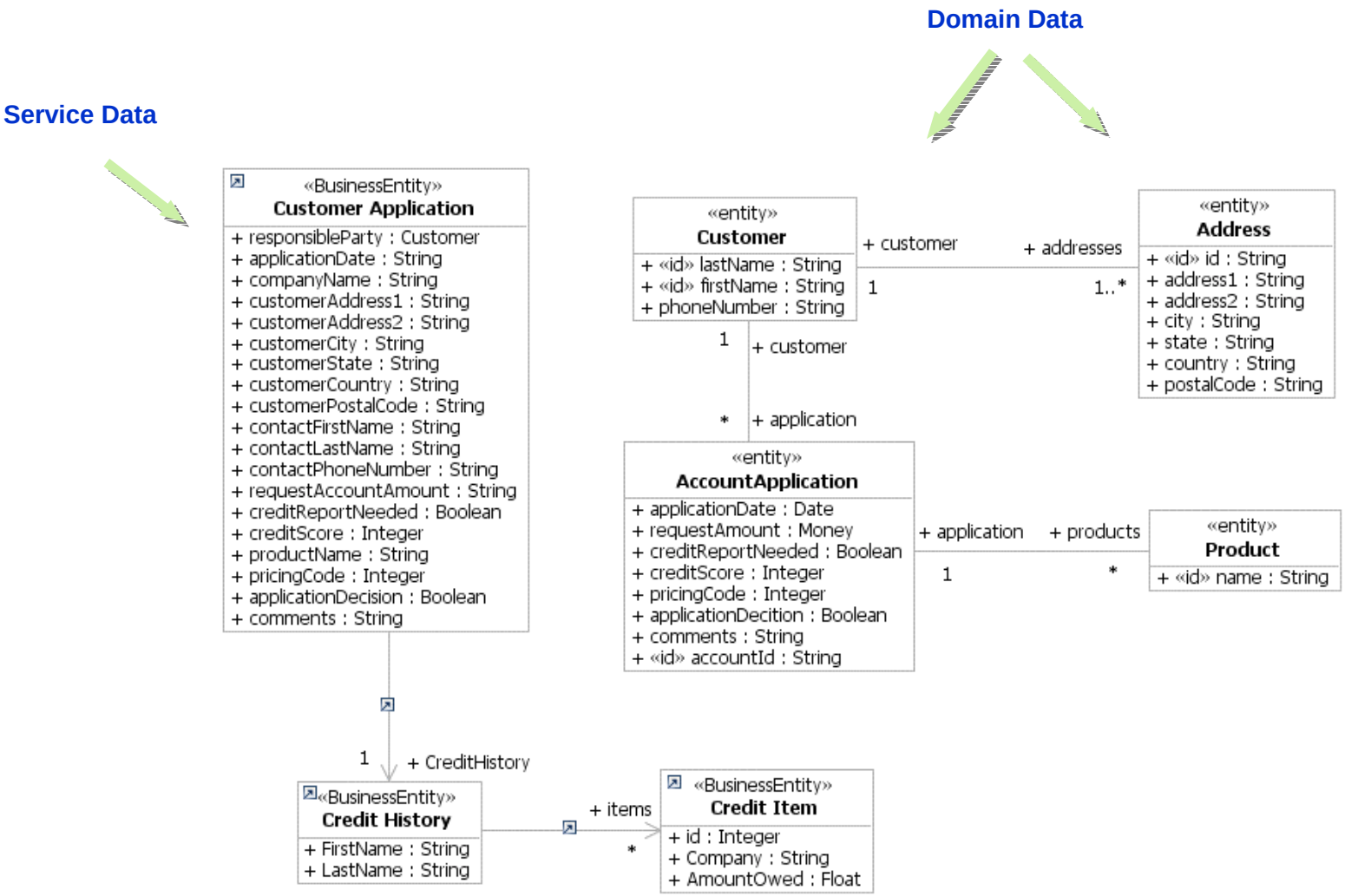
In the process of creating the activity realizing the provided services operations, Al realizes he needs some persistent business domain information. He needs information about customers, projects, accounts, addresses, and credit history in order to verify the account request. Some of this information 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 business 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.

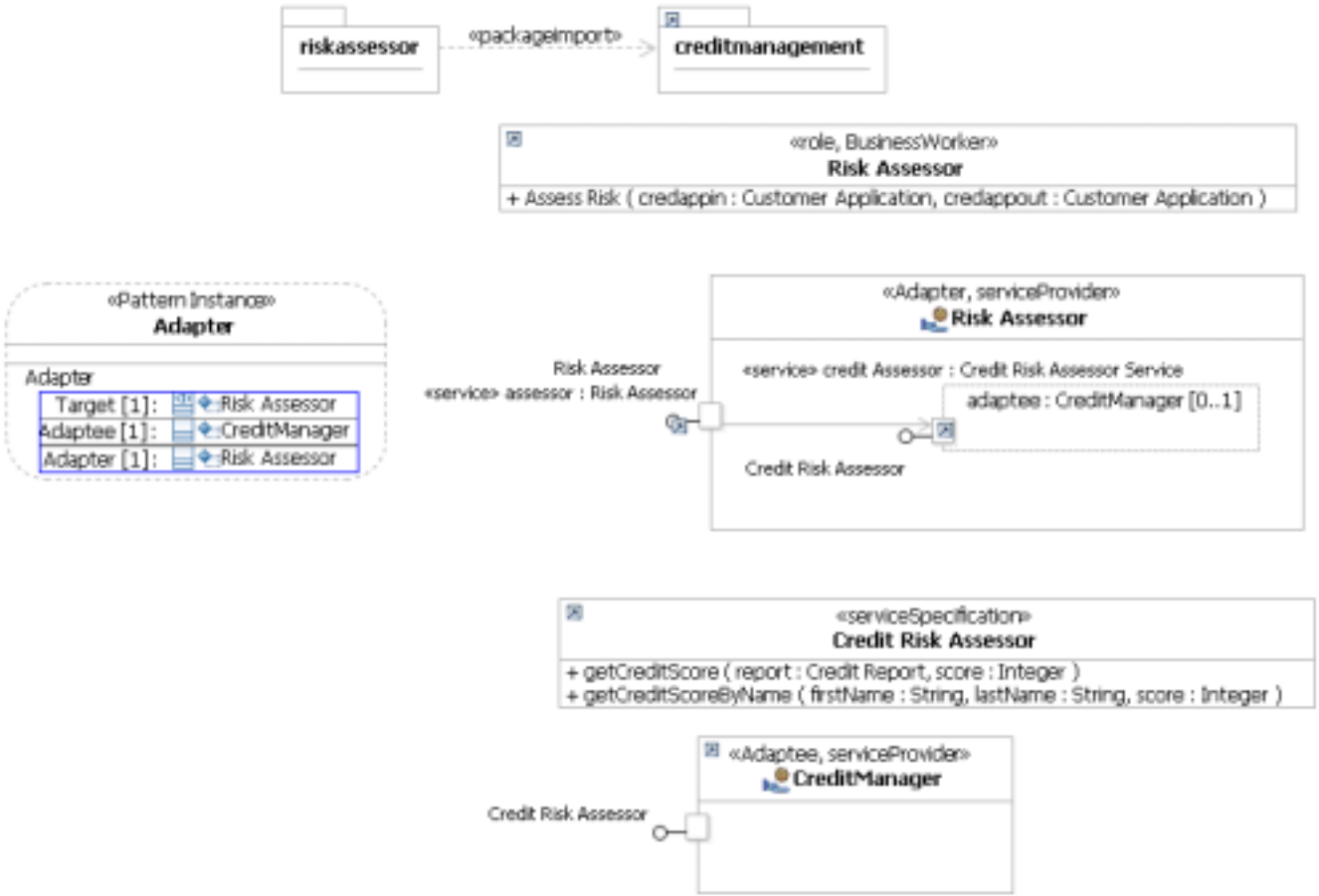
Al may decide later that some domain functions need to be exposed as reusable services. He can do this by selecting the domain class and operations and use the development tools to automatically create s component that exposes that business domain function as an external service.

JK Enterprises Domain Model



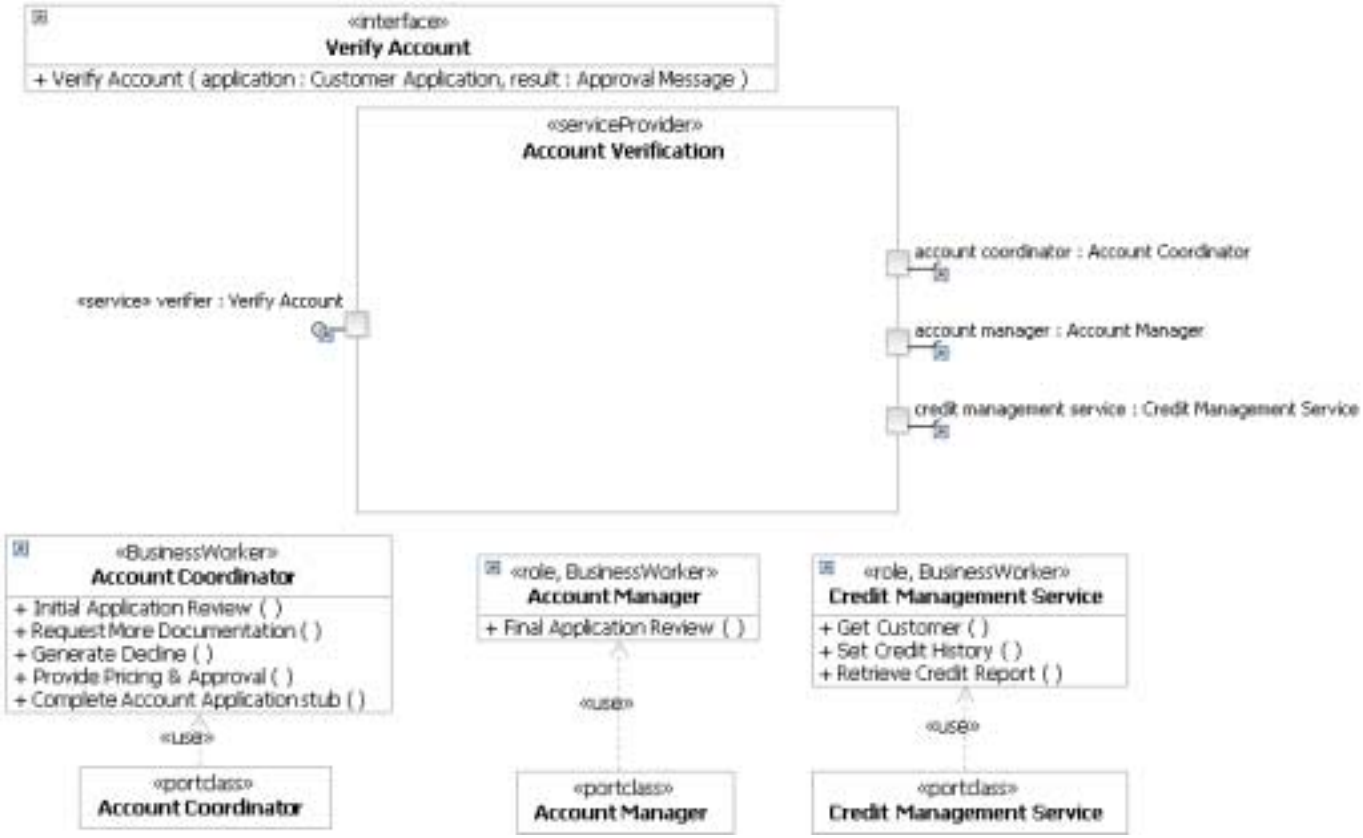
Find and reuse an existing asset

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.



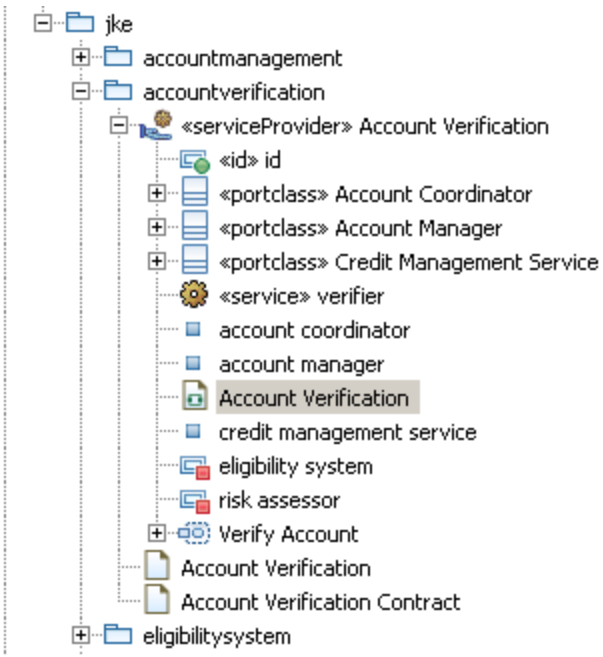
Create a component for Account Verification

Next AI 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. AI 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. AI 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.



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 eligibility system, 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.

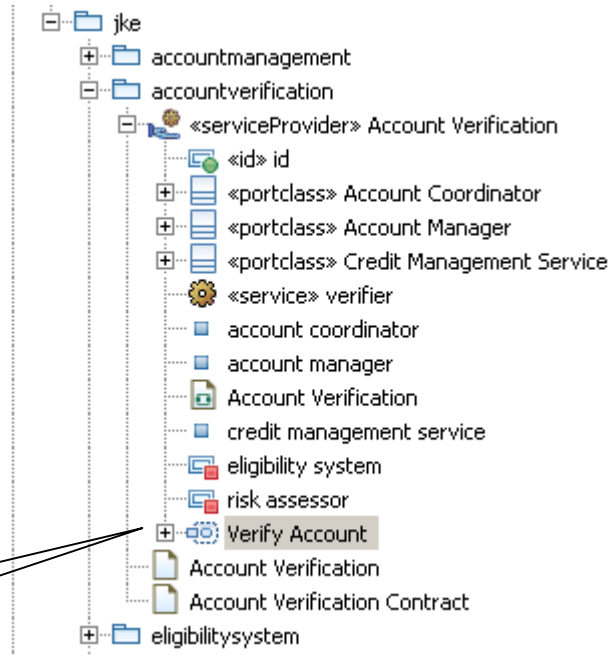


Create the account Verification service implementation

Now Al is ready to deal with the details of implementing the Verify Account service operation provided by the Account Verification component. He begins by creating an Verify Account 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 roles 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.

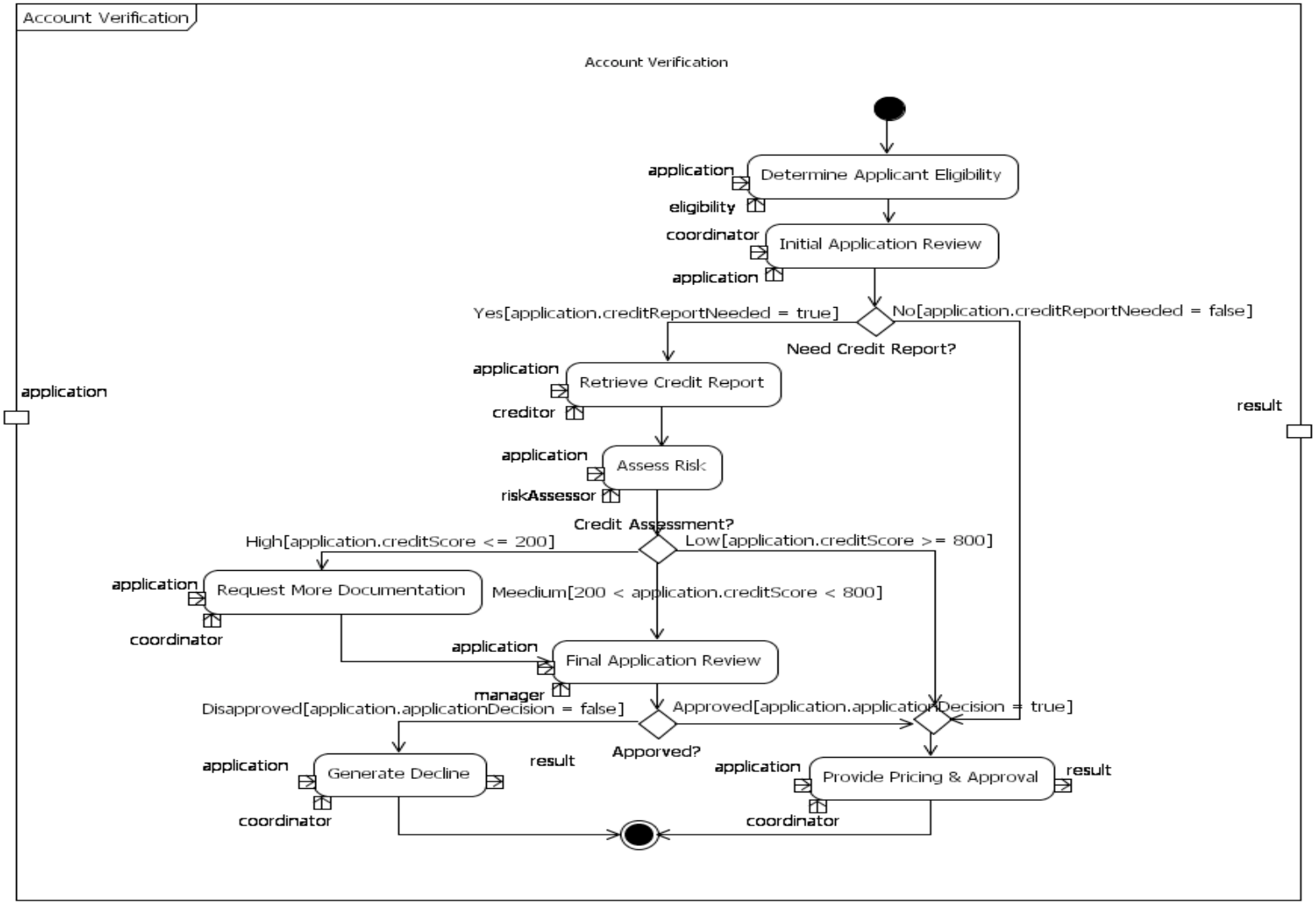
Al also looks at the business service contract and sees that the collaboration contains a number of constraints that realize KPIs for the business goals and objectives. The component must provide detailed constraints that realize each of the constraints in the contract. He adds these constraints. These 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.



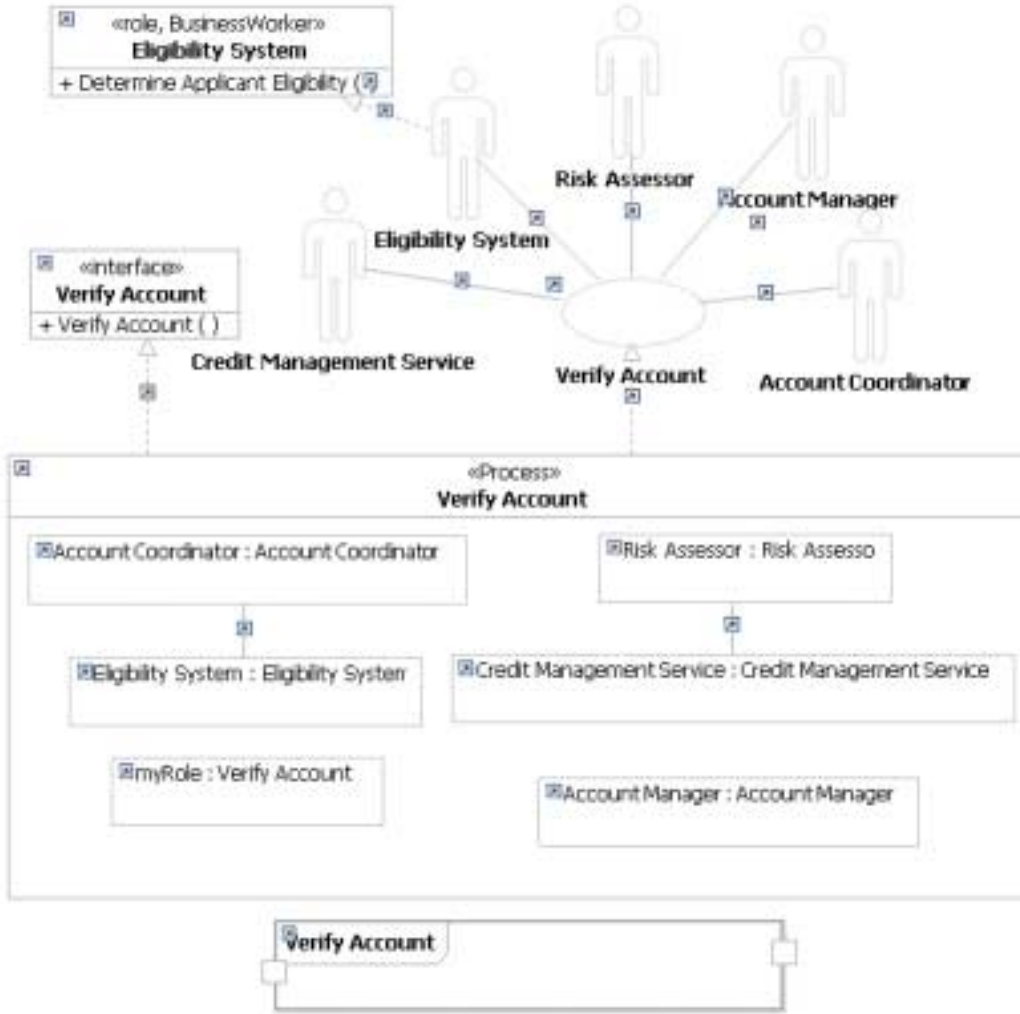
The Verify Account activity is the method for the Verify Account operation provided by the component

Verify Account implementation



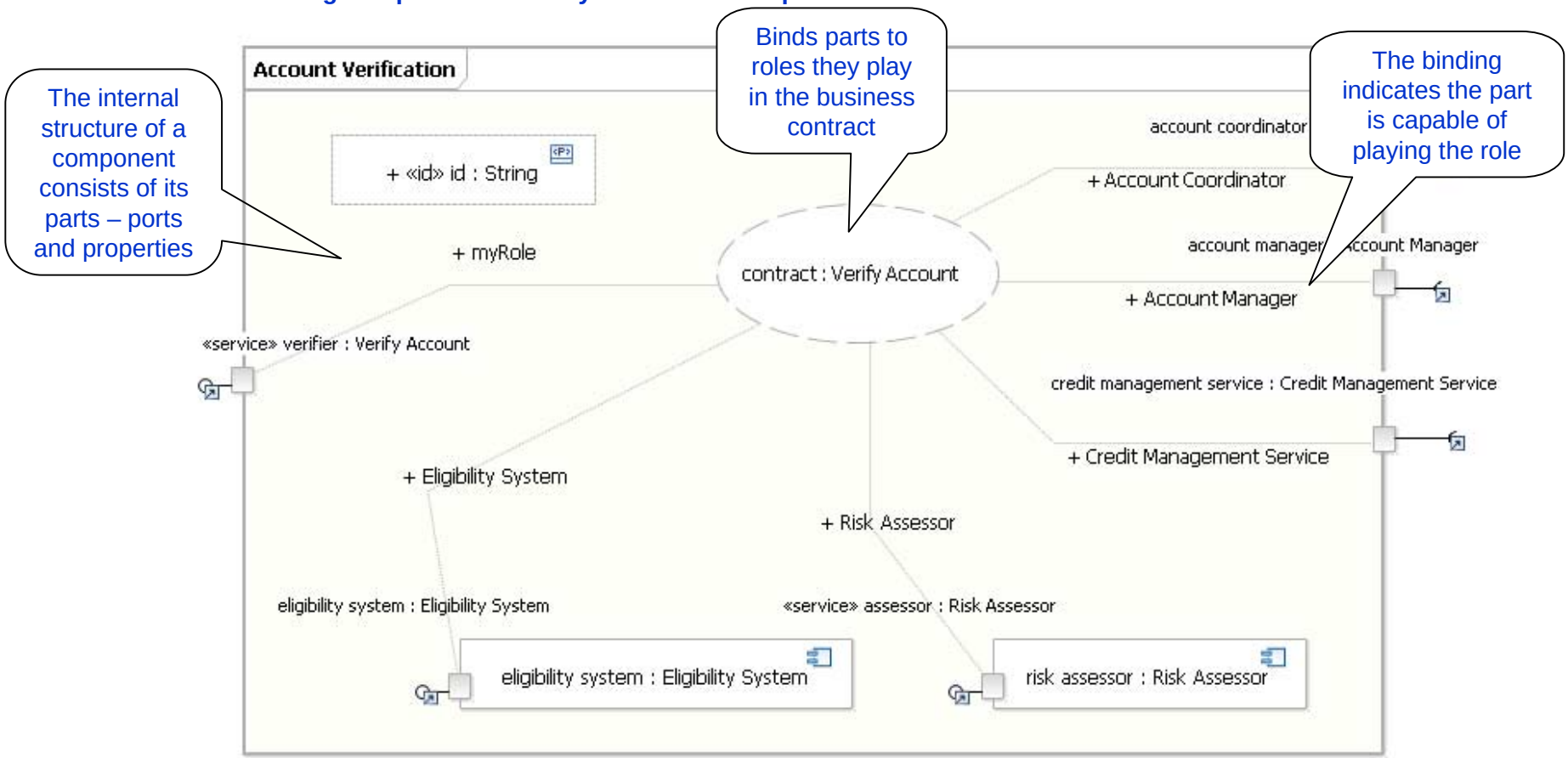
Recall the Business Services Contract

- We found, built, or adapted something to play each of these roles
- Now we need to link the roles to the parts that play them
- This shows how the contract is fulfilled by our services solution

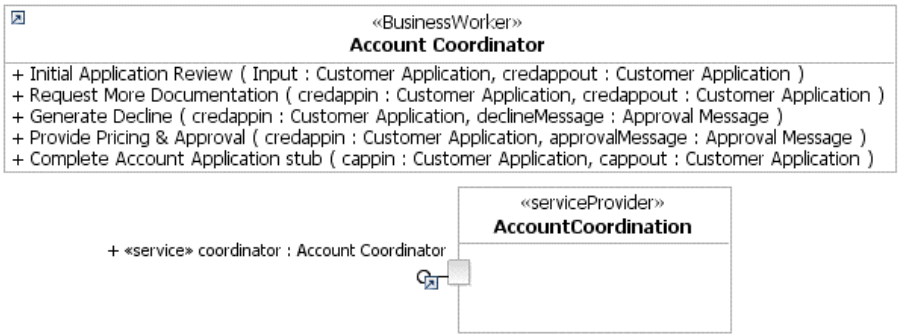


Link the component to the business contract it fulfills

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.

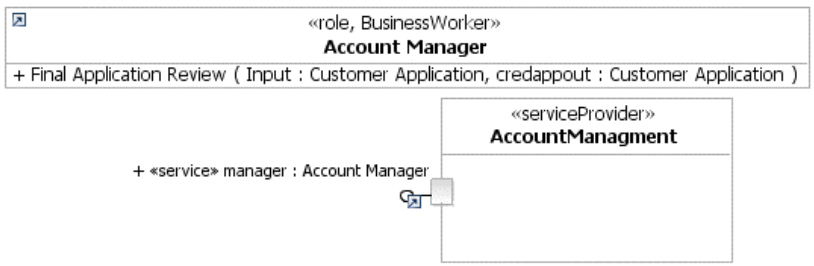


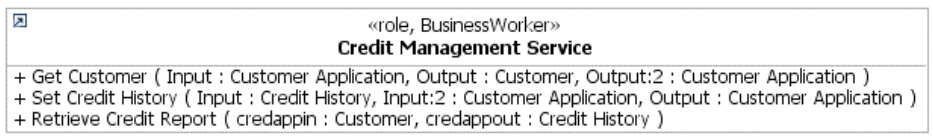
Acme Solutions provides some services we need



Account Coordination services

Account Management services

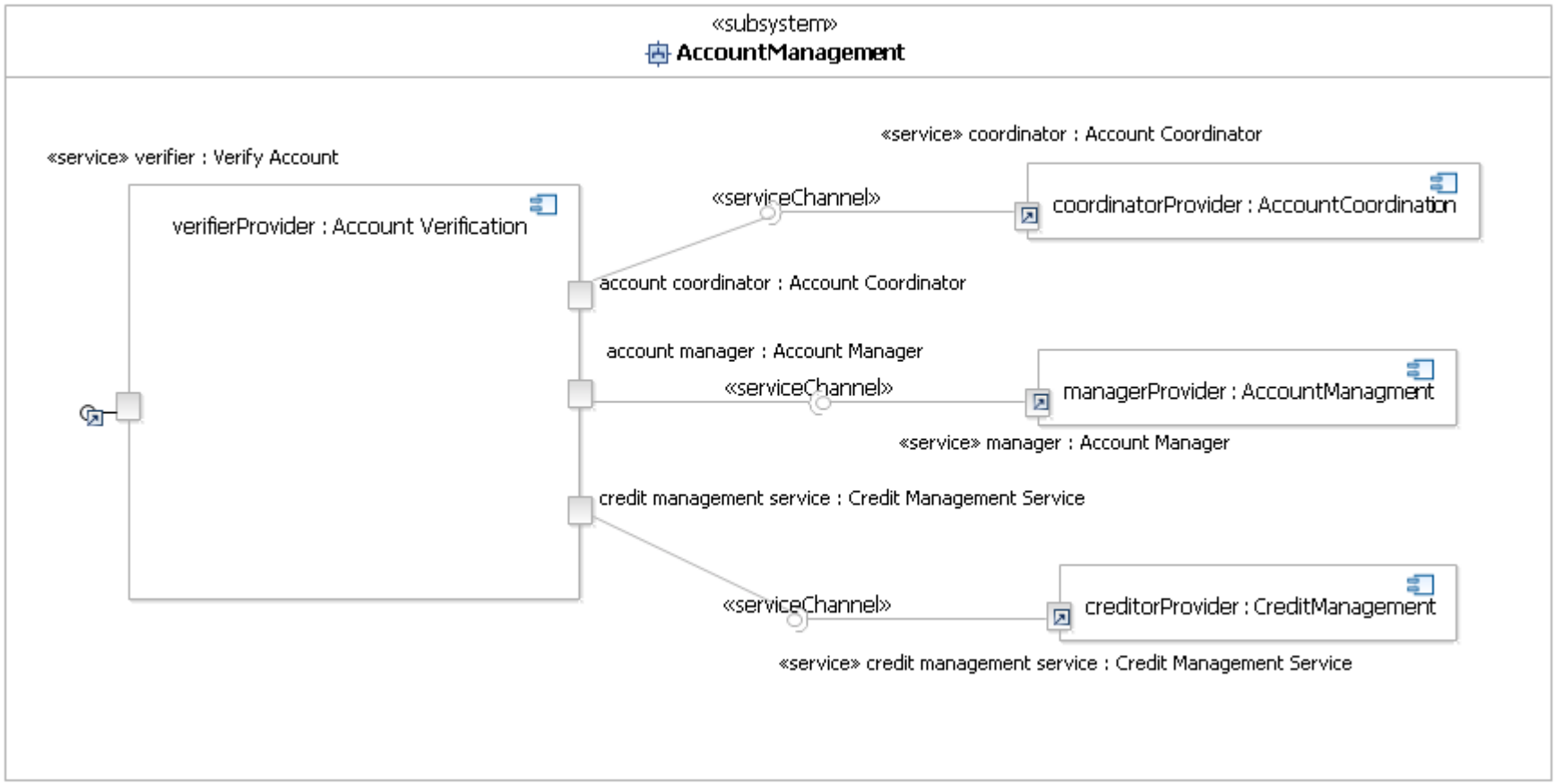




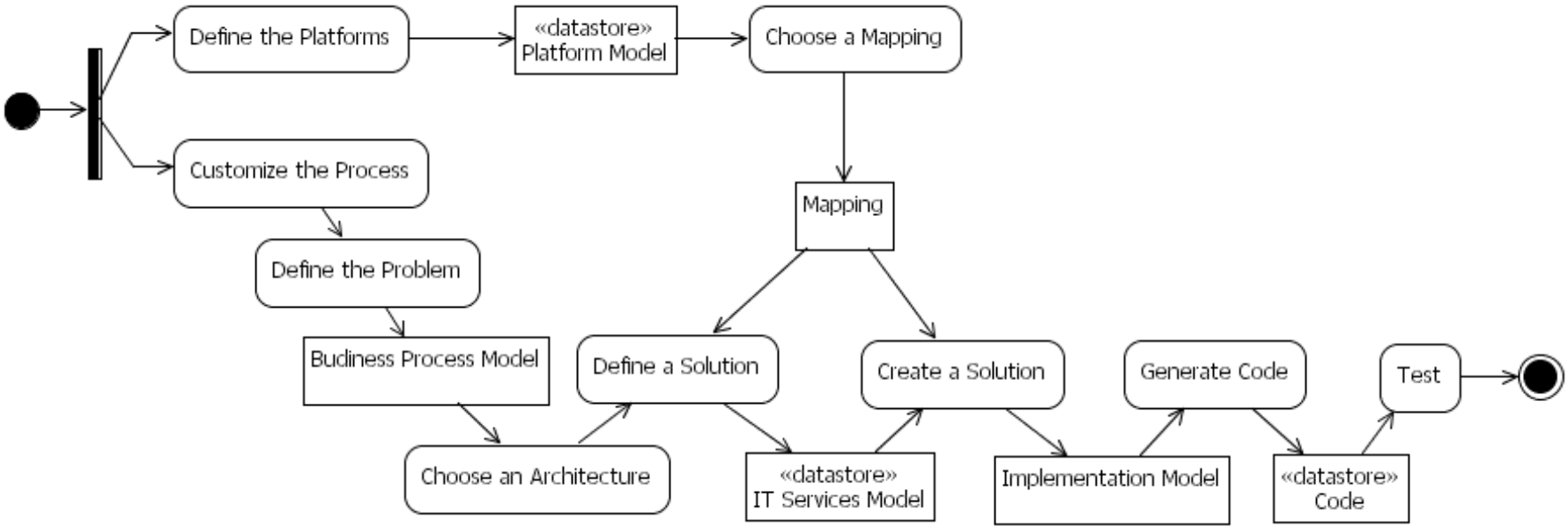
Credit Management services

Assemble the Account Management Solution

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



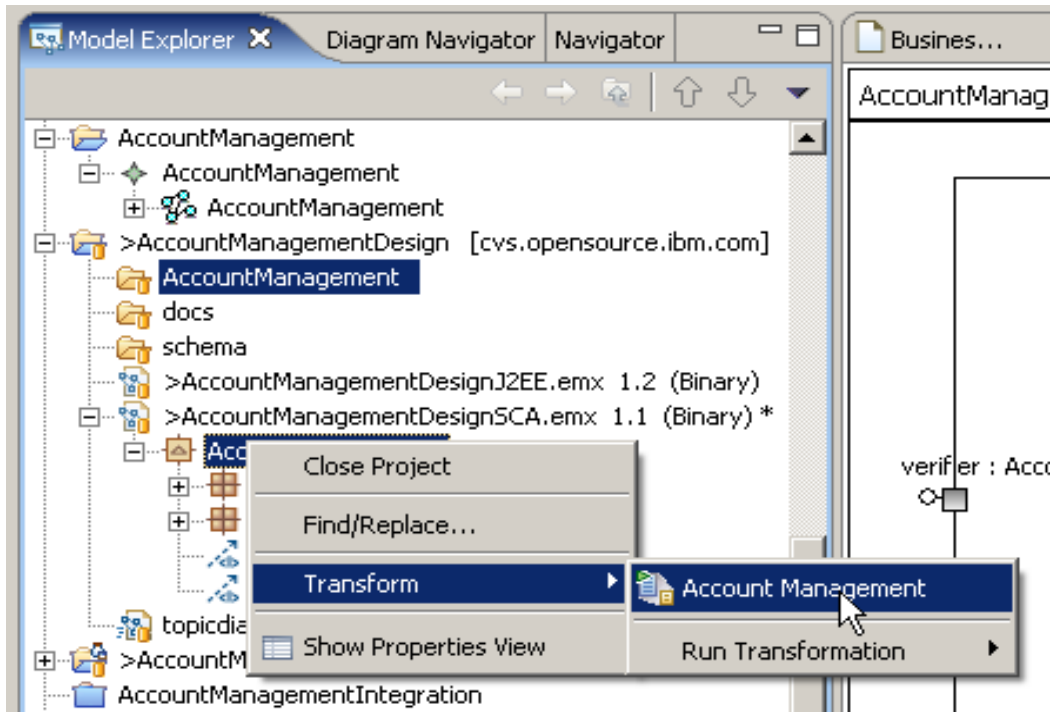
MDA Process in a Nutshell



Generate the implementation based on the chosen architecture

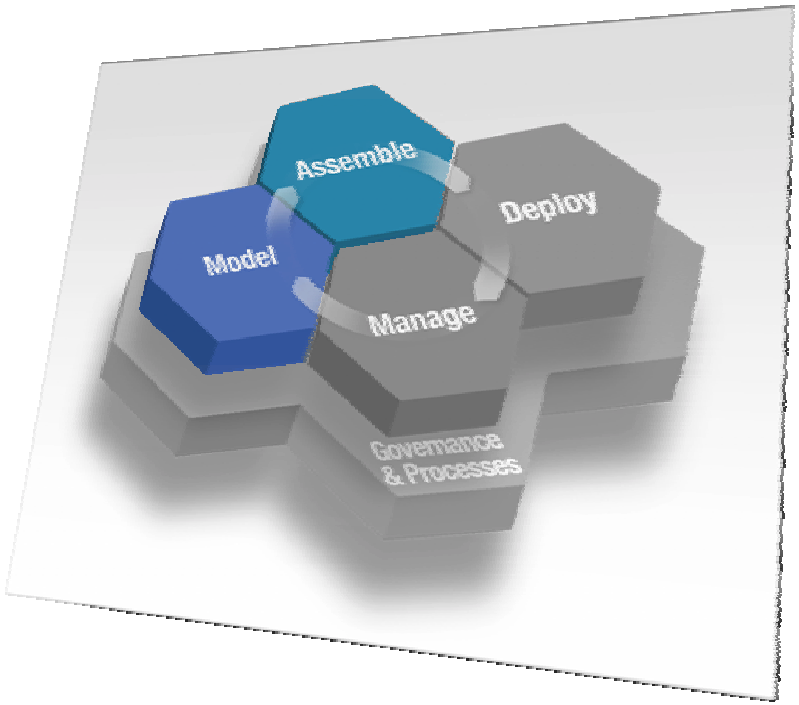
Al has decided to use Web Services to realize his SOA solution. He selects the Web Services transform and configures the transform to generate the Web Services using the desired interchange styles and transformation options.

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

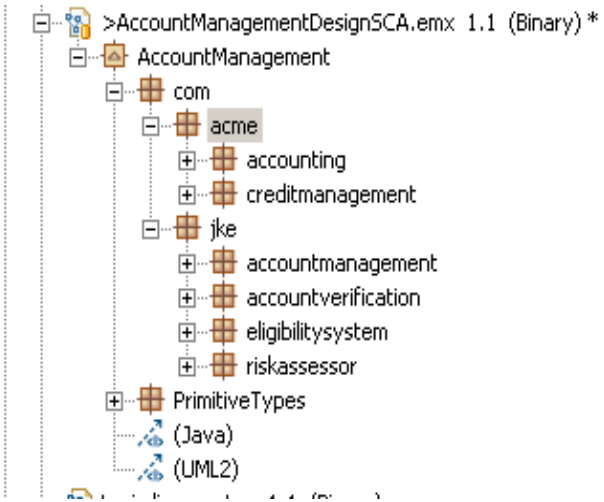
Architect Al has completed a services design model that fulfills the business requirements and has generate the initial solution artifacts using UML to SOA transforms. Developer Dan now must complete the solution by assembling the parts, implementing the incomplete components, deploying the solution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

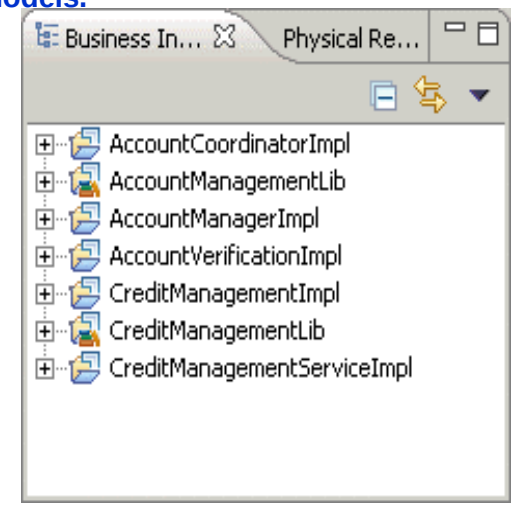
- Review the architected solution generated by UML-to-SOA
- Implement the persistence mechanisms to allow service implementations to access the persistent data sources
- Implement the eligibilityImpl.determineApplicationEligibility method
- Implement the riskAssessorImpl.assessRisk method by invoking 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

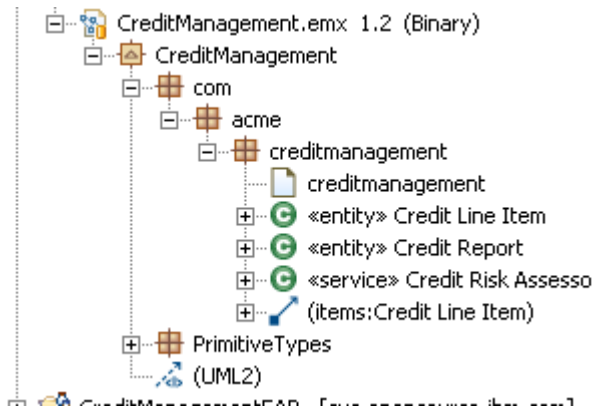
Review the architected solution

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 (the Impl projects) is created for each component in the transformed models.

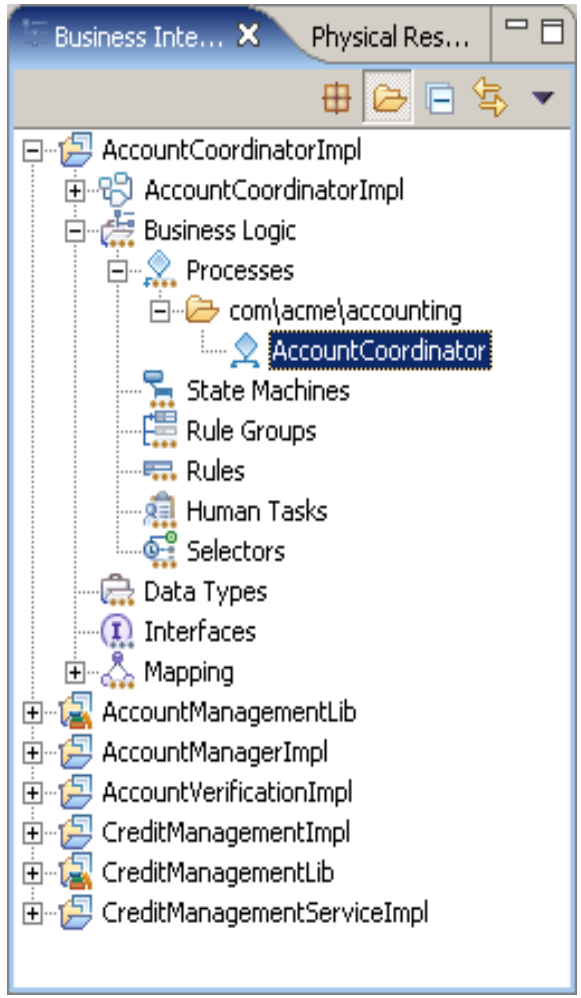






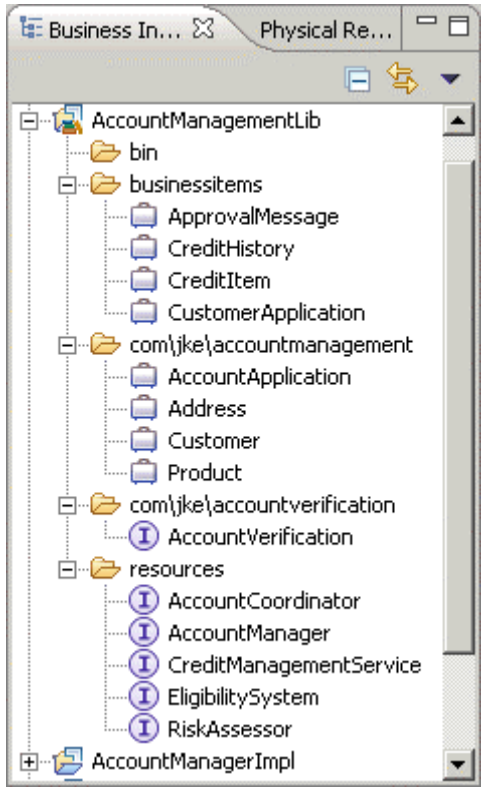
AccountCoordinatorImpl

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.



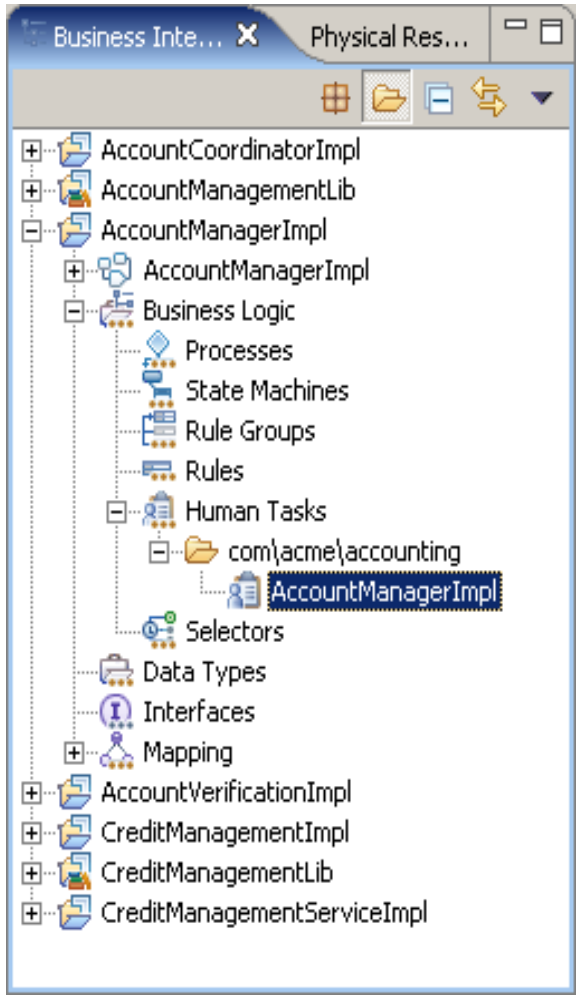
AccountManagementLib

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.



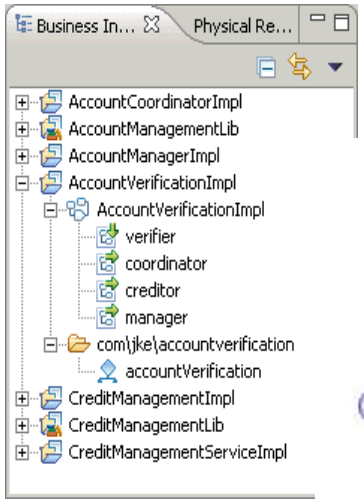
AccountManagerImpl

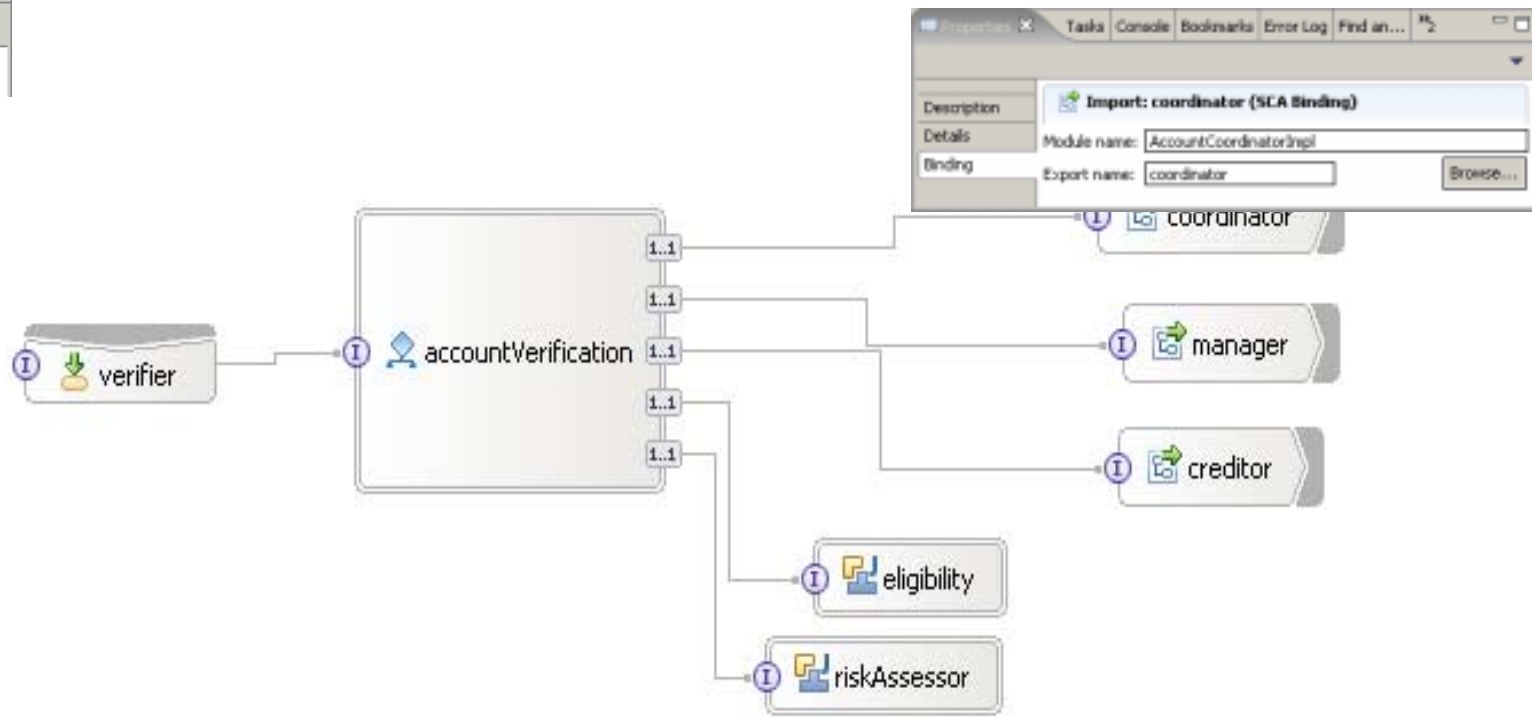
Provides an implementation of the AccountManager interface required by the AccountVerification process. This component is implemented using a human task.



AccountVerificationImpl

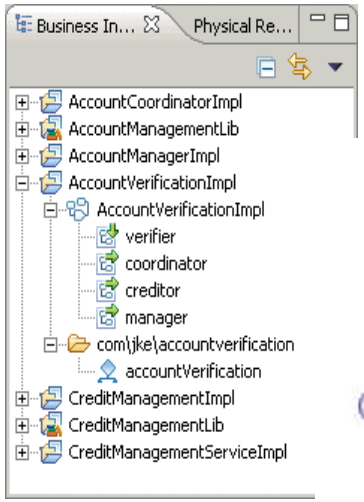
This module provides and implementation of the AccountVerification interface using a BPEL process. It is generated from the AccountVerification component in the solution architecture model.

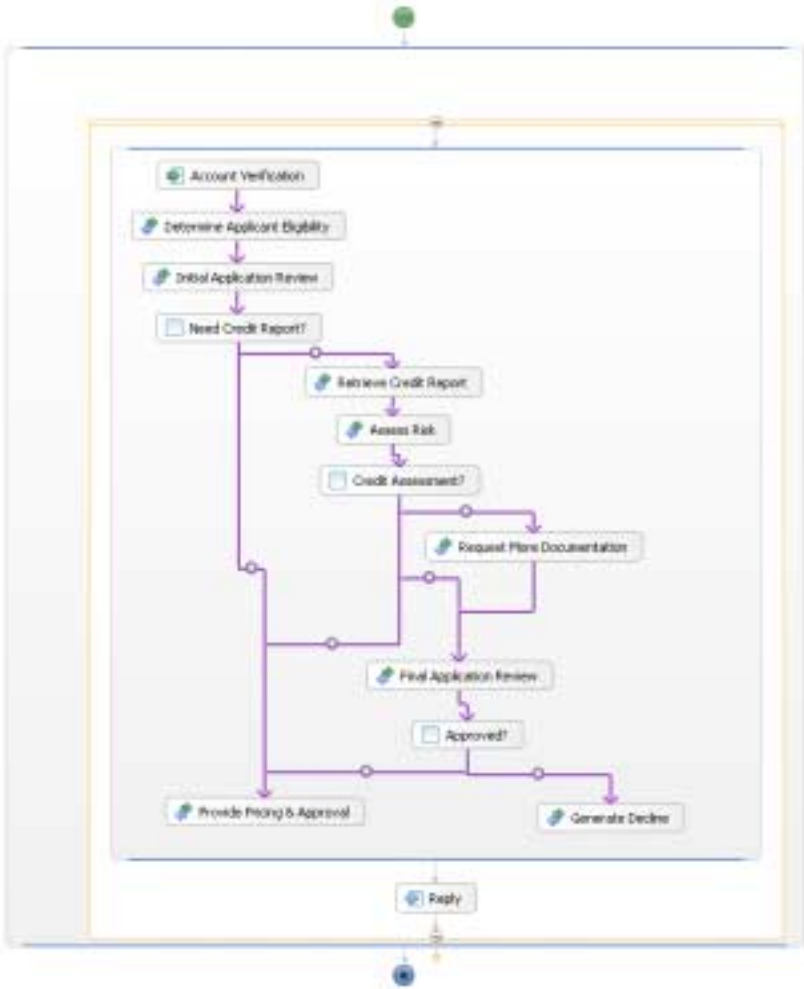


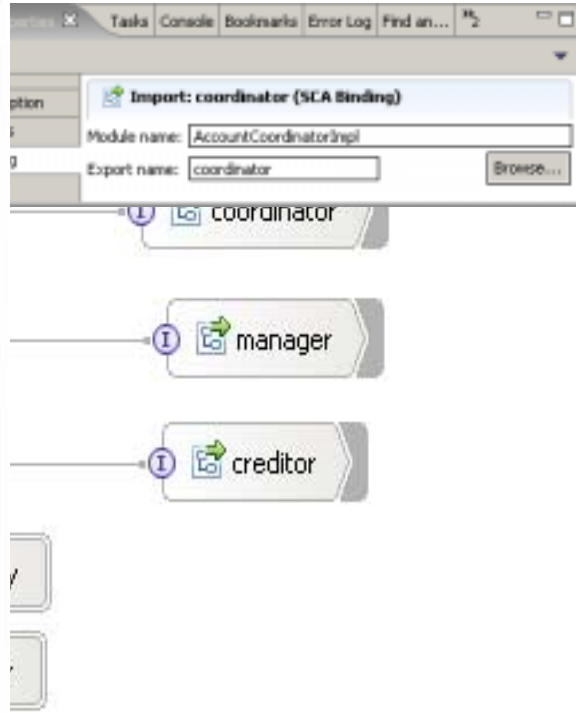


AccountVerificationImpl

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.

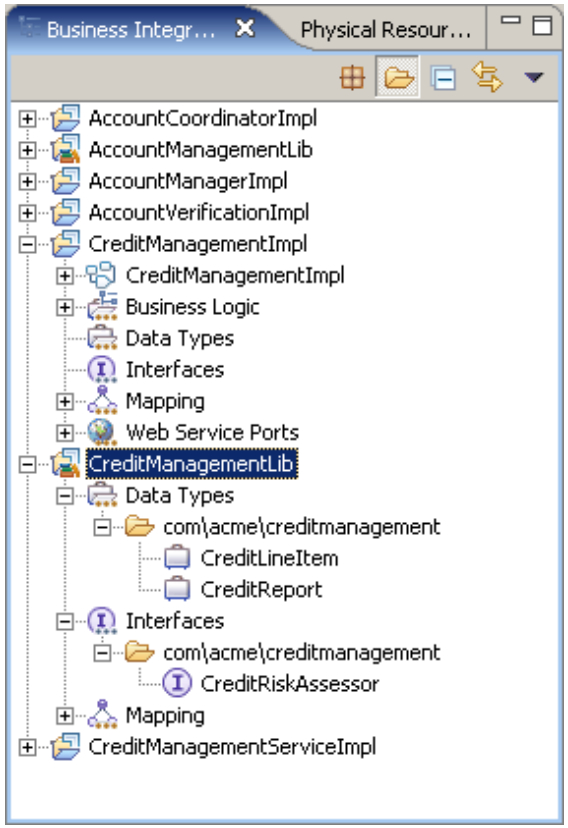






CreditManagementLib, CreditManagementImpl

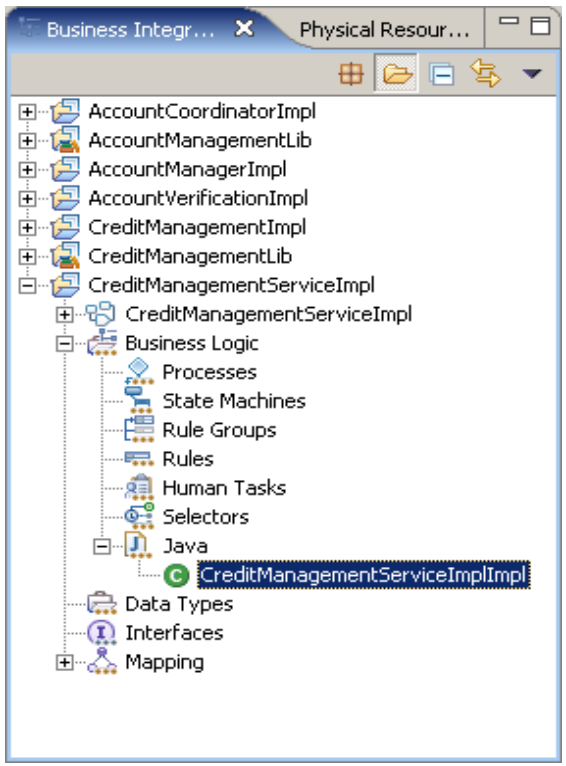
CreditManagementLib is an existing library that AI obtained from RAS and is used to implement risk assessment by adapting an existing service. CreditManagementImpl provides the service implementation using an existing EJB session bean.



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.

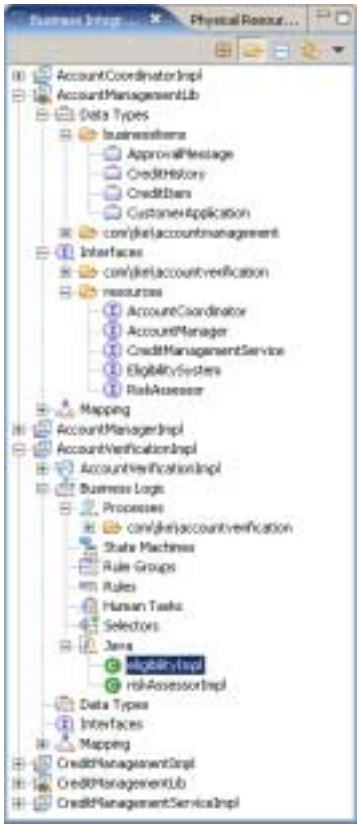
This module provides an implementation of the CreditManagementService interface required by the AccountVerification process. This service provides the access to a credit report while CreditManagementLib provides a service to calculate credit scores. The service is implemented in Java (and currently does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



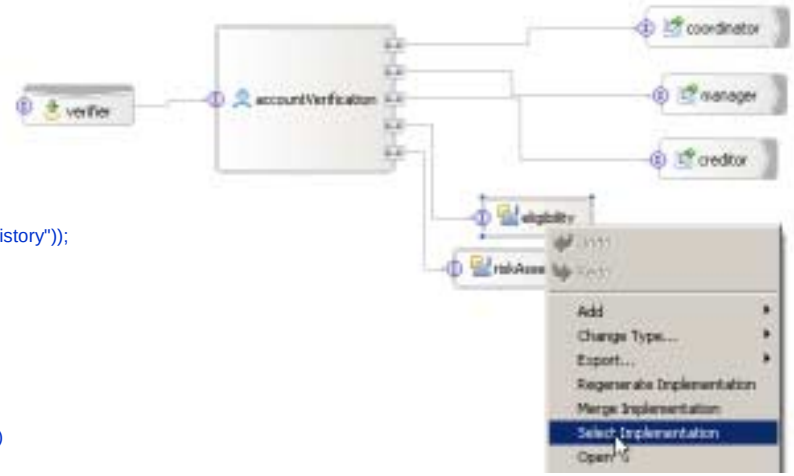

```
/**
 * Method generated to support implemention of operation "retrieveCreditReport" defined for WSDL port type
 * named "interface.CreditManagementService".
 *
 * The presence of commonj.sdo.DataObject as the return type and/or as a parameter
 * type conveys that its a complex type. Please refer 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.

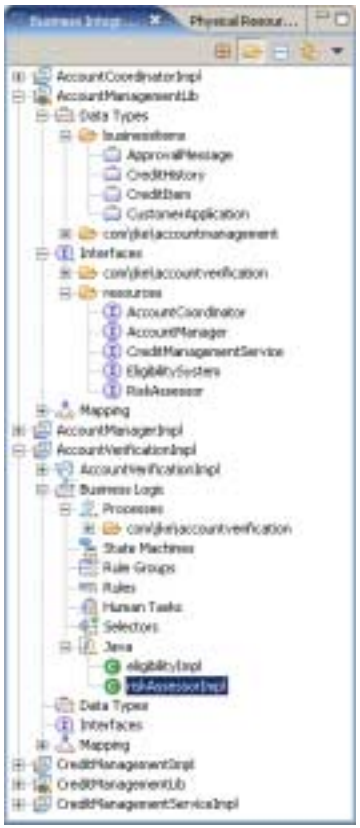


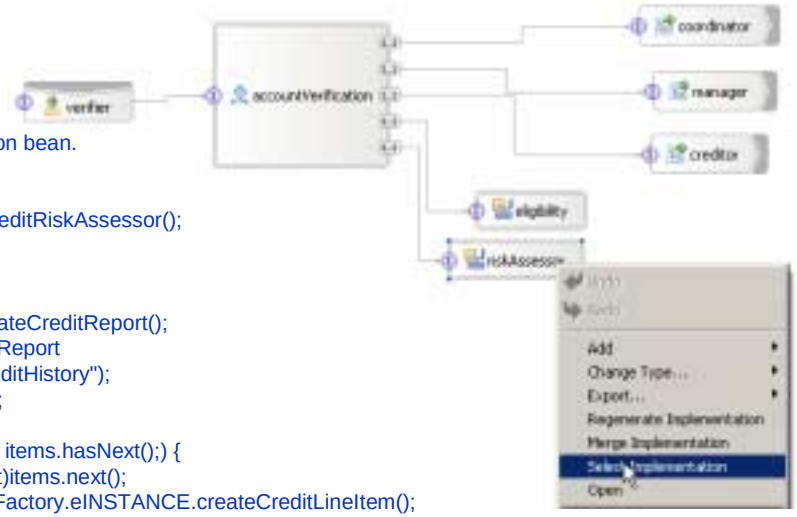

```
public DataObject determineApplicantEligibility(DataObject credappin) {
    boolean isEligible = true;
    // Get the Credit History for the applicant
    CreditReportFacadeRemote creditReportMediator = createCreditReportFacadeRemote();
    CreditReport creditReport = null;
    try {
        creditReport = 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.



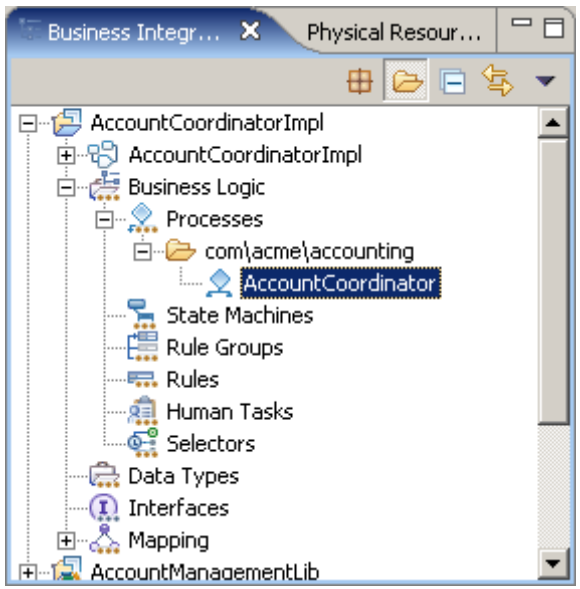



```
public DataObject assessRisk(DataObject credappin) {
        // Get an instance of the CreditRiskAssessor session bean.
        // It is this service we are adapting
        if (creditRiskAssessor == null) {
                creditRiskAssessor = createCreditRiskAssessor();
        }
        // Create 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

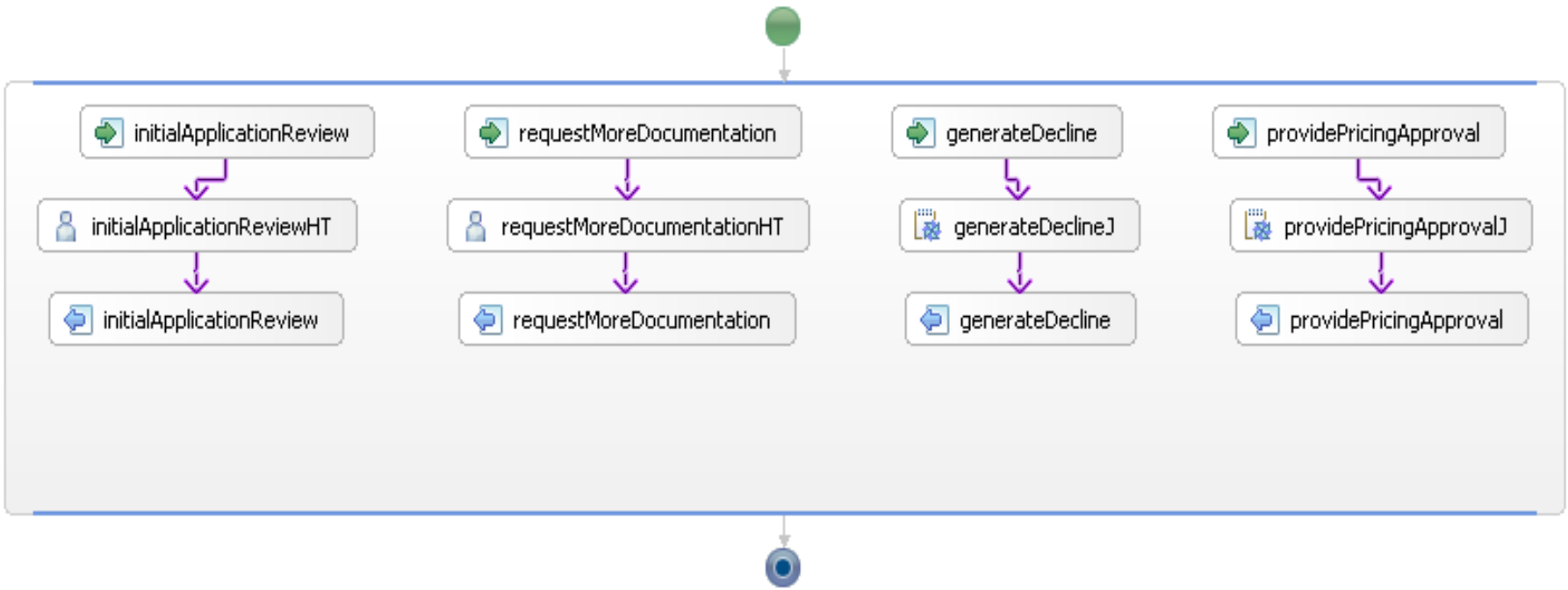
The AccountCoordinator is responsible for a number of tasks. Although these tasks are all assigned to the AccountCoordinator role in the business process, developer Dan has decided that each of these responsibilities may need to be implemented different ways, and that these implementations may change 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



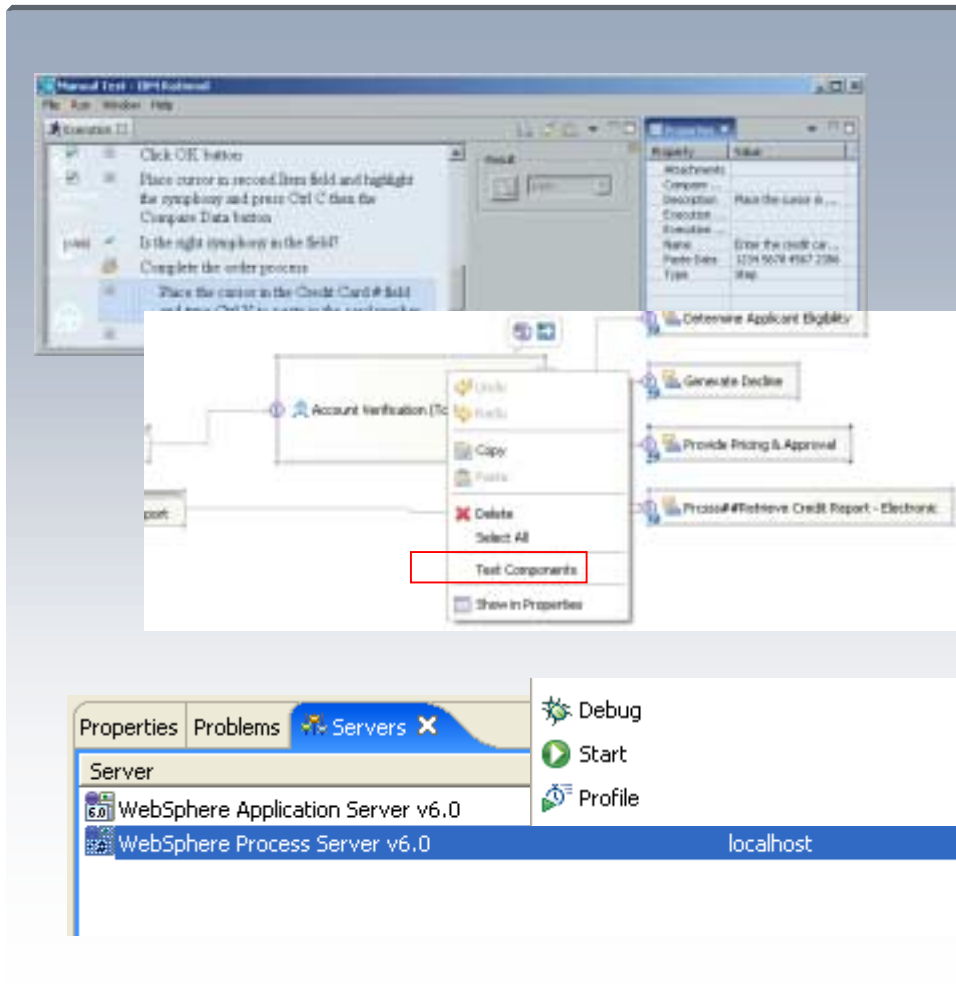
Human Task and User Interface Development

- 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

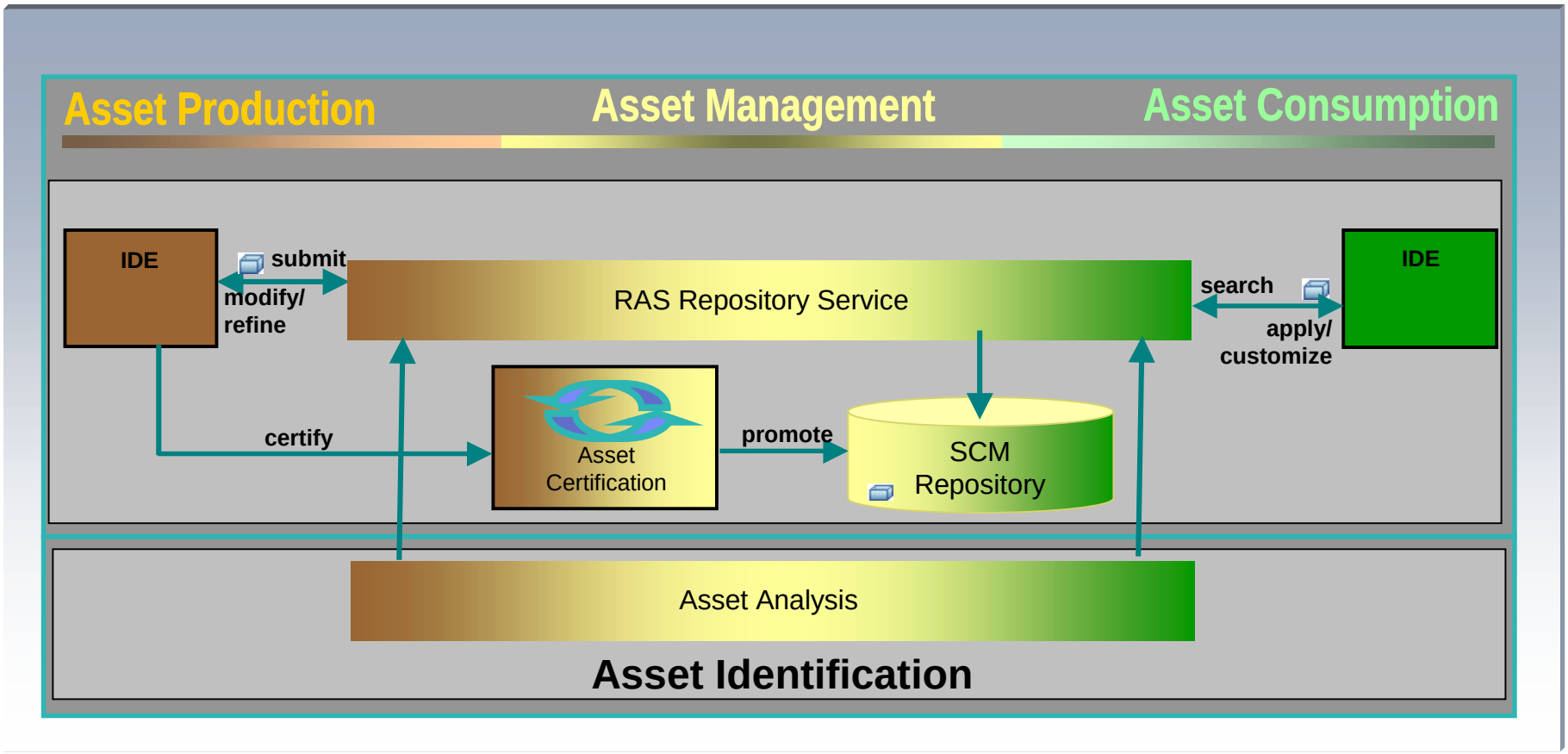
- Testing needs to occur across Business Driven Development:
  - Component
  - Service
  - Business Process
  - Composite Application
  - Functional
  - User Interface
  - Performance
  - Regression
  - System

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



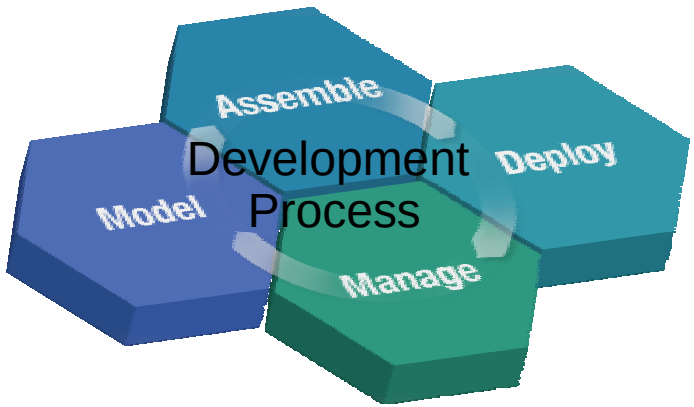
Development-time Service Lifecycle

- 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 :

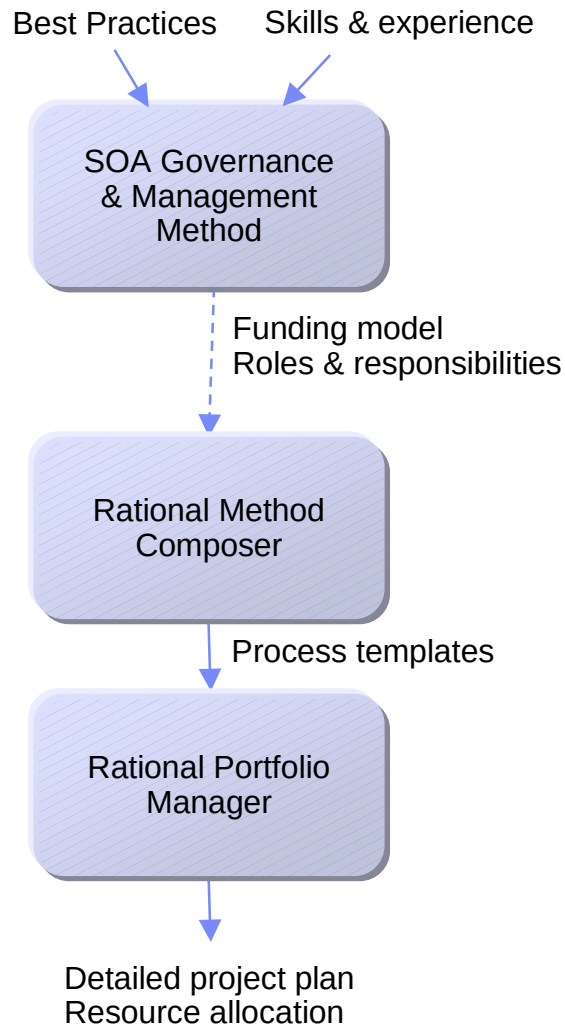
- Determine the business priorities
- Execute development against those priorities
- Measure their effectiveness

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)

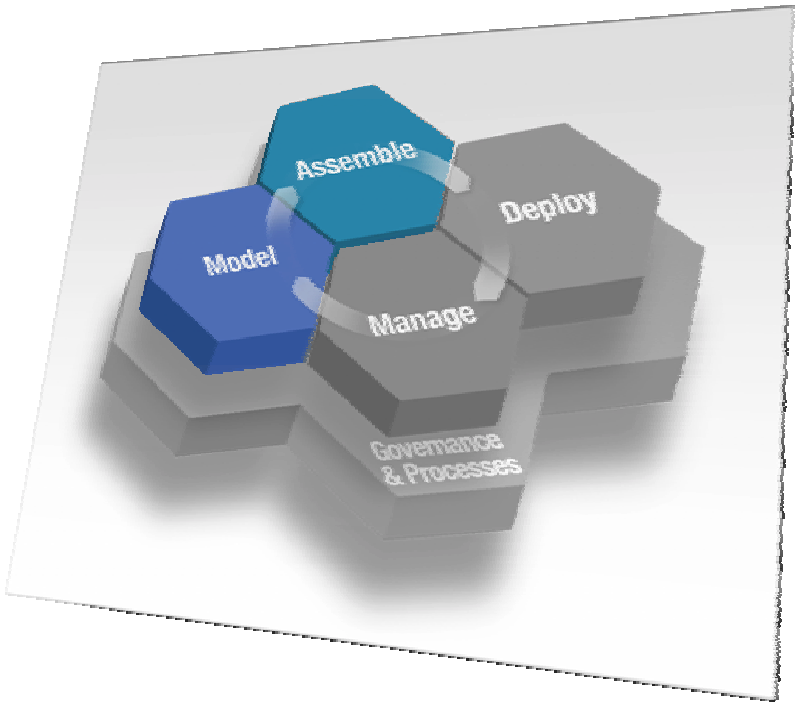
Doing Governance for Business Driven Development

- **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

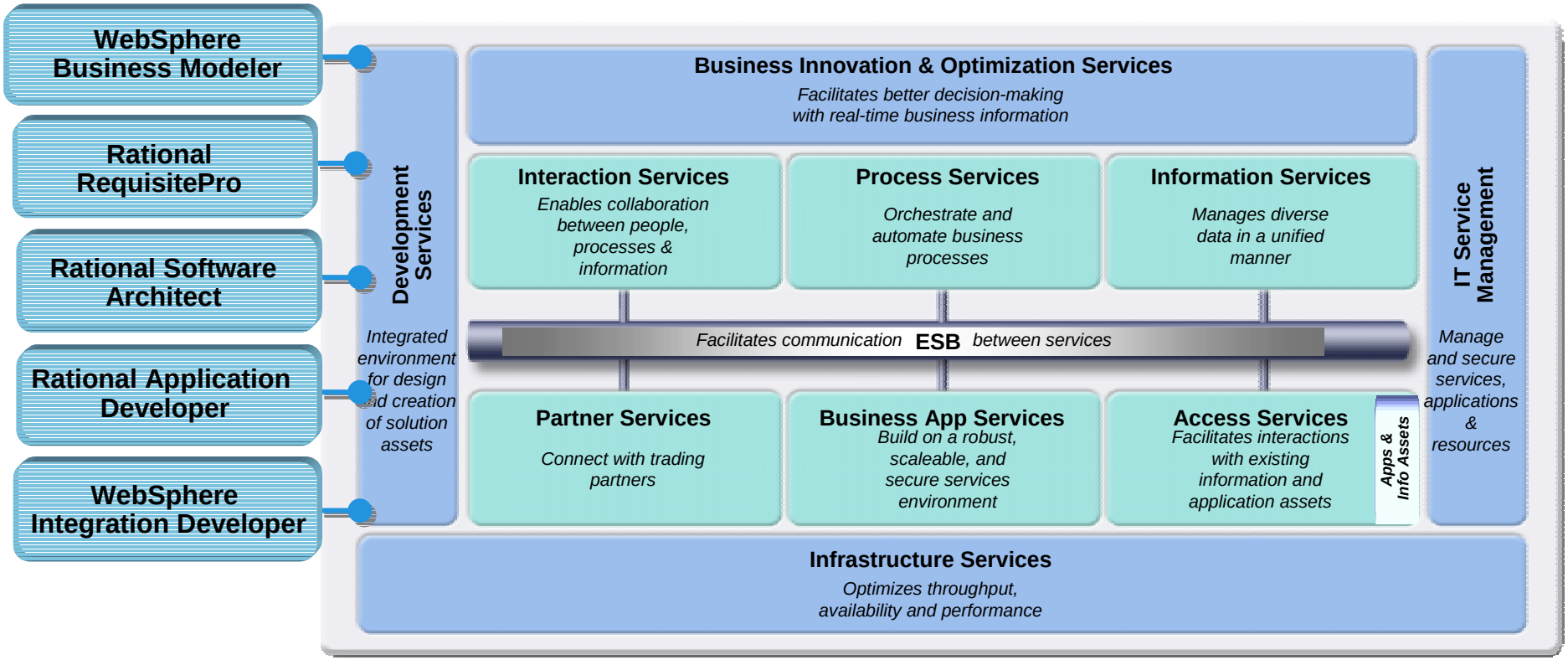


Agenda

- 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

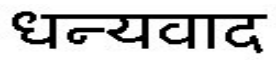


Key Products - Business Driven Development



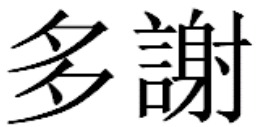
More Information

- 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/



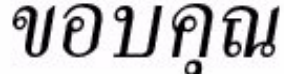
धन्यवाद

Hindi



多謝

Traditional Chinese



ขอบคุณ

Thai

Спасибо

Russian

Gracias

Spanish

Thank You

شكرا

Arabic

Obrigado

Brazilian Portuguese

Danke

German

Grazie

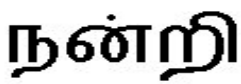
Italian

多谢

Simplified Chinese

Merci

French



நன்றி

Tamil

감사합니다

Korean

ありがとうございました

Japanese

Questions

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

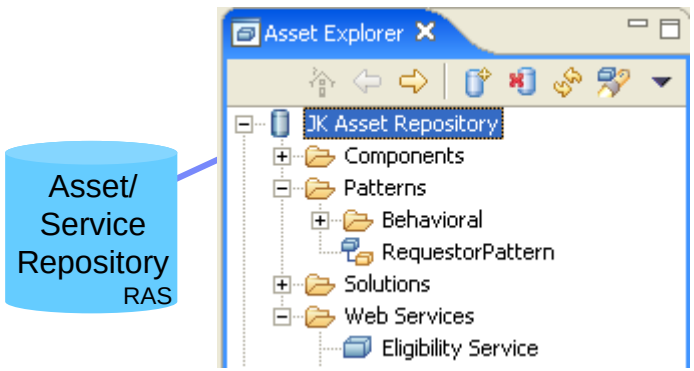
- IBM Pattern Solution Assets currently available:
  - Enterprise Patterns
    - J2EE Patterns
      - Session Façade, Business Delegate, Message Façade, etc.
  - WebSphere Platform Messaging Patterns
  - State Oriented Portlet Patterns
  - Tivoli System Automation Failover Configuration Pattern

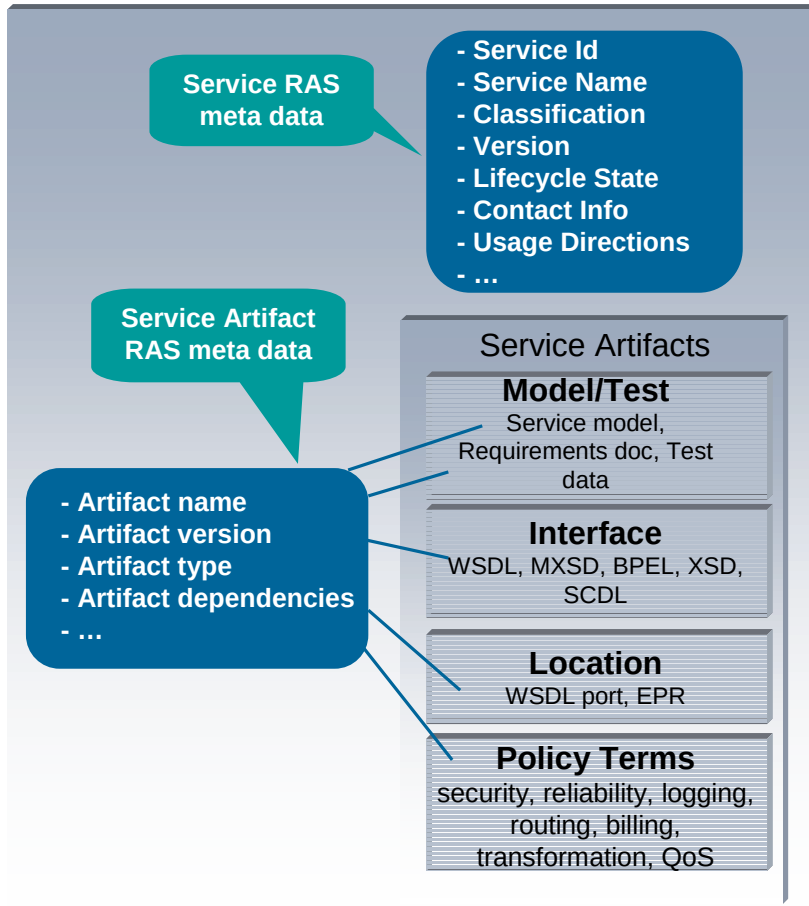
- New patterns are being developed

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

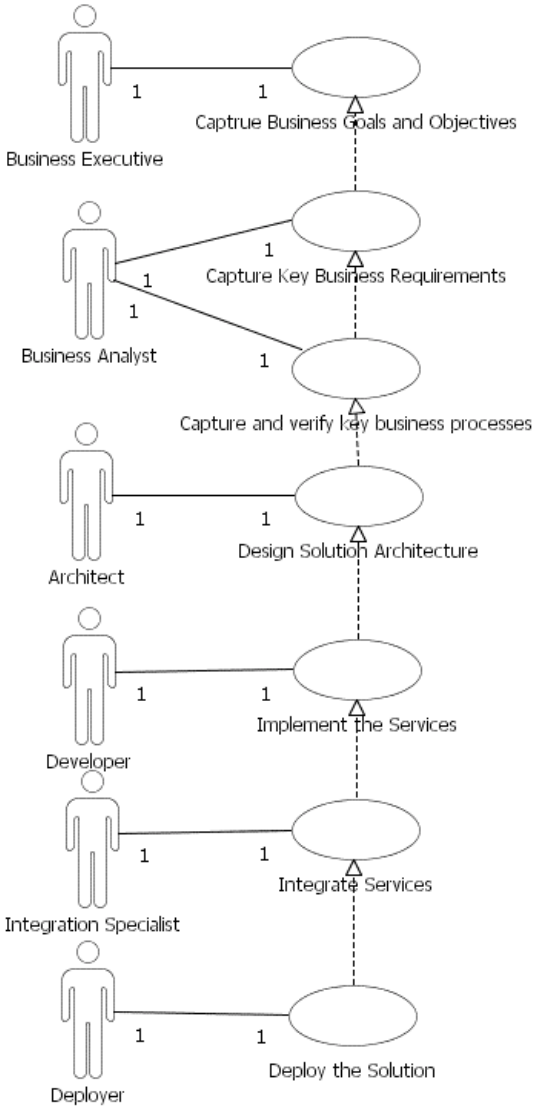
Reusable Asset Specification (RAS)

- 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)





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 purchasec into a completed solution.

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

BDD and BSM formalizes some parts of RUP

- Business use cases model requirements necessary to meet business goals and objectives
  - From the perspective of the key external stakeholders
- BDD uses process models to realize business use cases
  - An additional level of formalism that can be validated through simulation
- Then view process models as Business Services Models
  - From the perspective of collaborating roles instead of sequence of tasks
- SOA solutions then fulfill the Business Services Models
  - Through collaboration usages which show how the solution fulfills the contract
- This is a refinement of RUP that uses concepts of CBD to formalize the relationships between the parts
  - Better validation, more repeatable
  - Better traceability, easier to manage change, easier governance

Business Driven Development in the Larger Context