



Mettre en œuvre le développement à base de modèles pour l'architecture SOA

Thierry BOURRIER
tbourrier@fr.ibm.com



ON DEMAND BUSINESS™

Agenda

- Pourquoi SOA?
- Qu'est ce que SOA?
- SOA et la plateforme de développement IBM Rational
- Conclusion

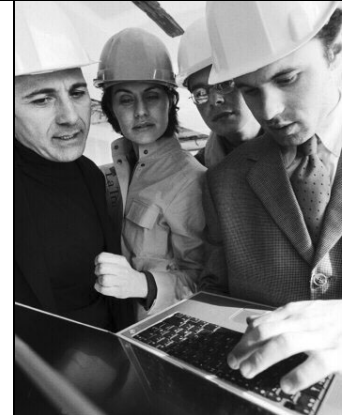
Agenda

- Pourquoi SOA?
- Qu'est ce que SOA?
- SOA et la plateforme de développement IBM Rational
- Conclusion

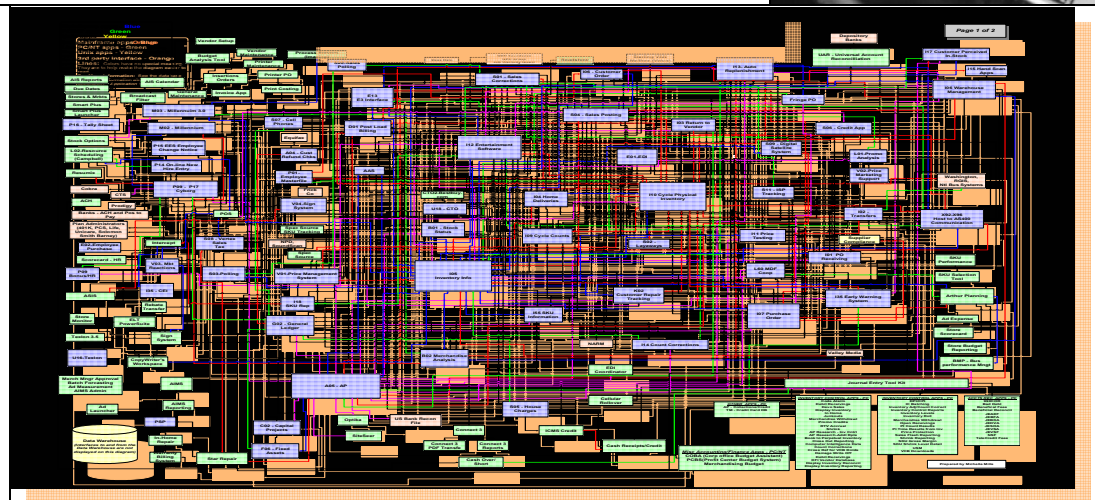
La flexibilité du métier dépend de la flexibilité de l'IT

“Today’s IT architectures, arcane as they may be, are the biggest roadblocks most companies face when making strategic moves.”

–McKinsey
“Flexible IT, Better Strategy”



- Complexité
- Application monolithique ou en silo
- Interfaces cachées
- Codages des Connexions spécifiques
- Pas conçue pour le changement



Architecture d'application pour société d'électronique grand publique

Service Oriented Architecture: the blueprint for change



“Service-oriented architecture deployments are increasing and becoming a mainstream architectural model for developing services, applications and infrastructures.”

*“Effective Web Services and SOBAs Require Management”
Frank Kenney, Gartner
September 28, 2004*

Pourquoi SOA?

- Flexibilité
- Facilite la réutilisation d’applications existantes
- Supporte l’implémentation des processus métier
- Concevoir pour le changement

Aligner objectifs Métier et IT pour augmenter les revenus et maîtriser les coûts

Pourquoi SOA aujourd'hui?

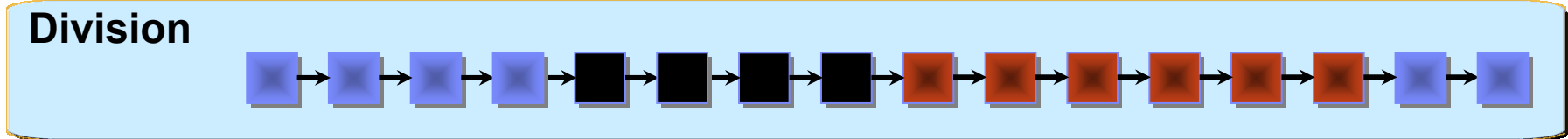
- **Rester dans la course:**
 - “We are taking apart each task and sending it ... to whomever can do it best, ... and then we are reassembling all the pieces”
from Thomas Friedman's 'The World is Flat'
- **Les standards et les technologies sont finalement en place, avec un large support des éditeurs**
- **Les bonnes pratiques sont disponible**
- **Les produits nécessaire pour commencer sont disponible aujourd'hui**



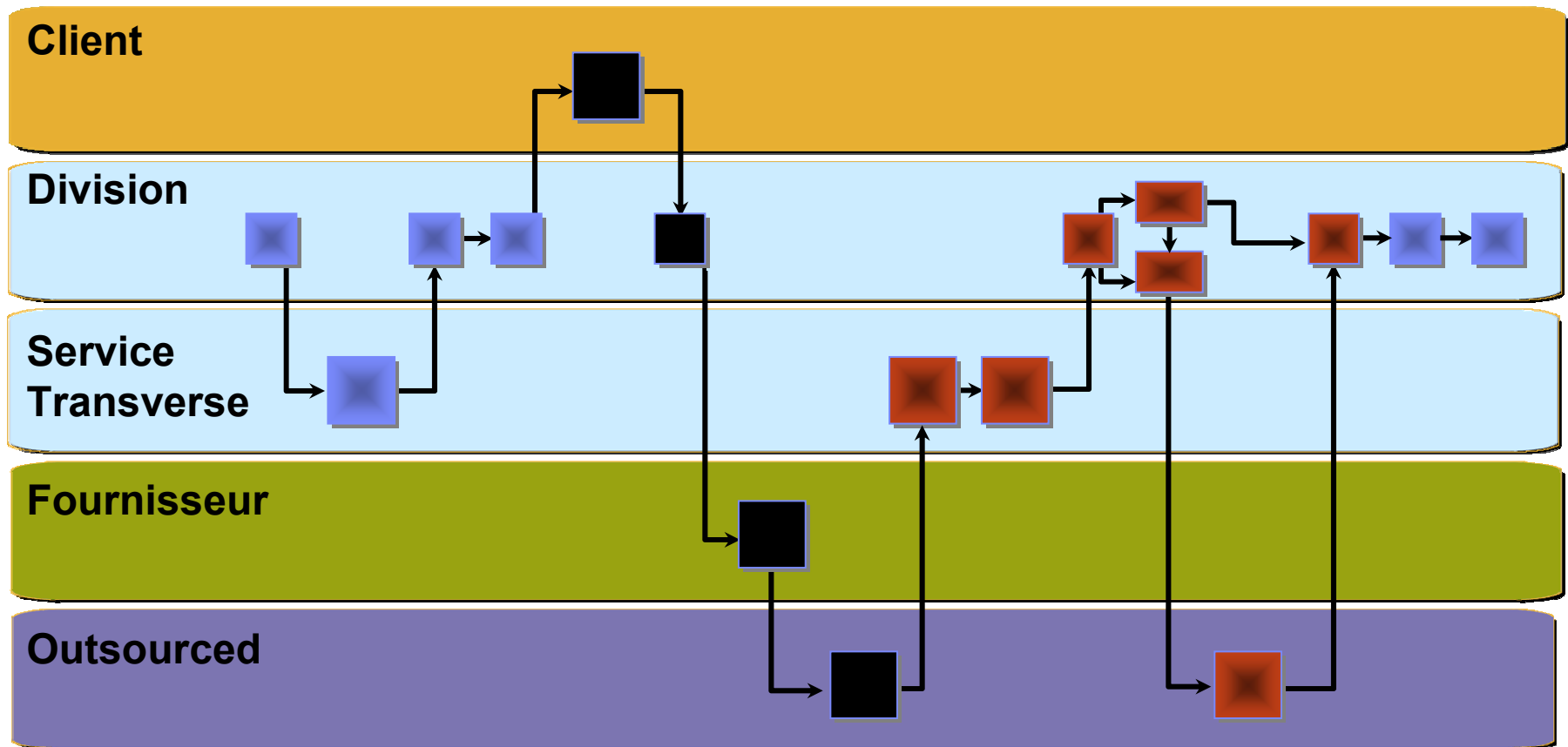
Agenda

- Pourquoi SOA?
- Qu'est ce que SOA?
- SOA et la plateforme de développement IBM Rational
- Conclusion

Besoin de Processus Métier Flexible

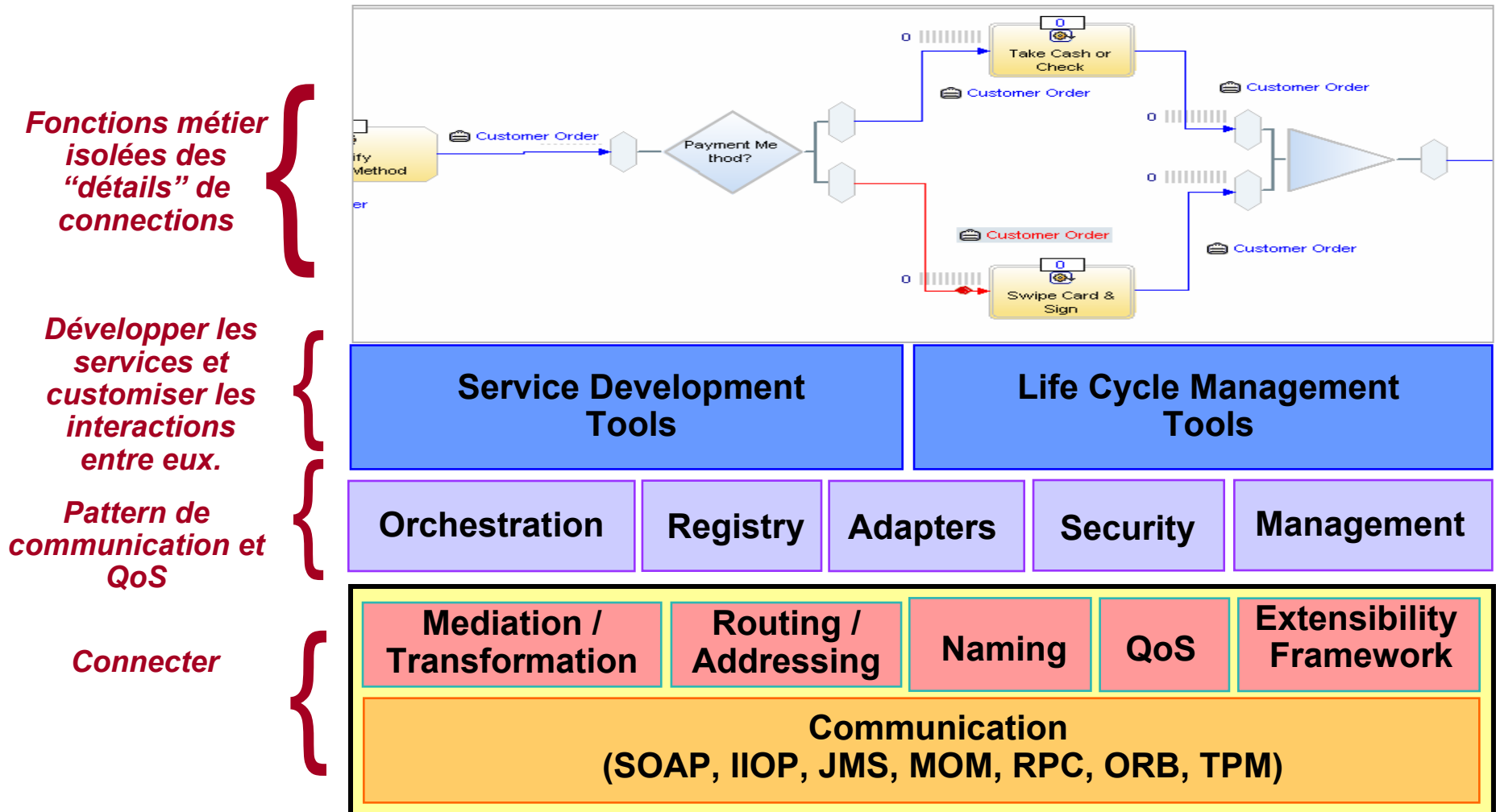


Besoin de Processus Métier Flexible



Changement: Optimisation des Processus, mutualisation, externalisation

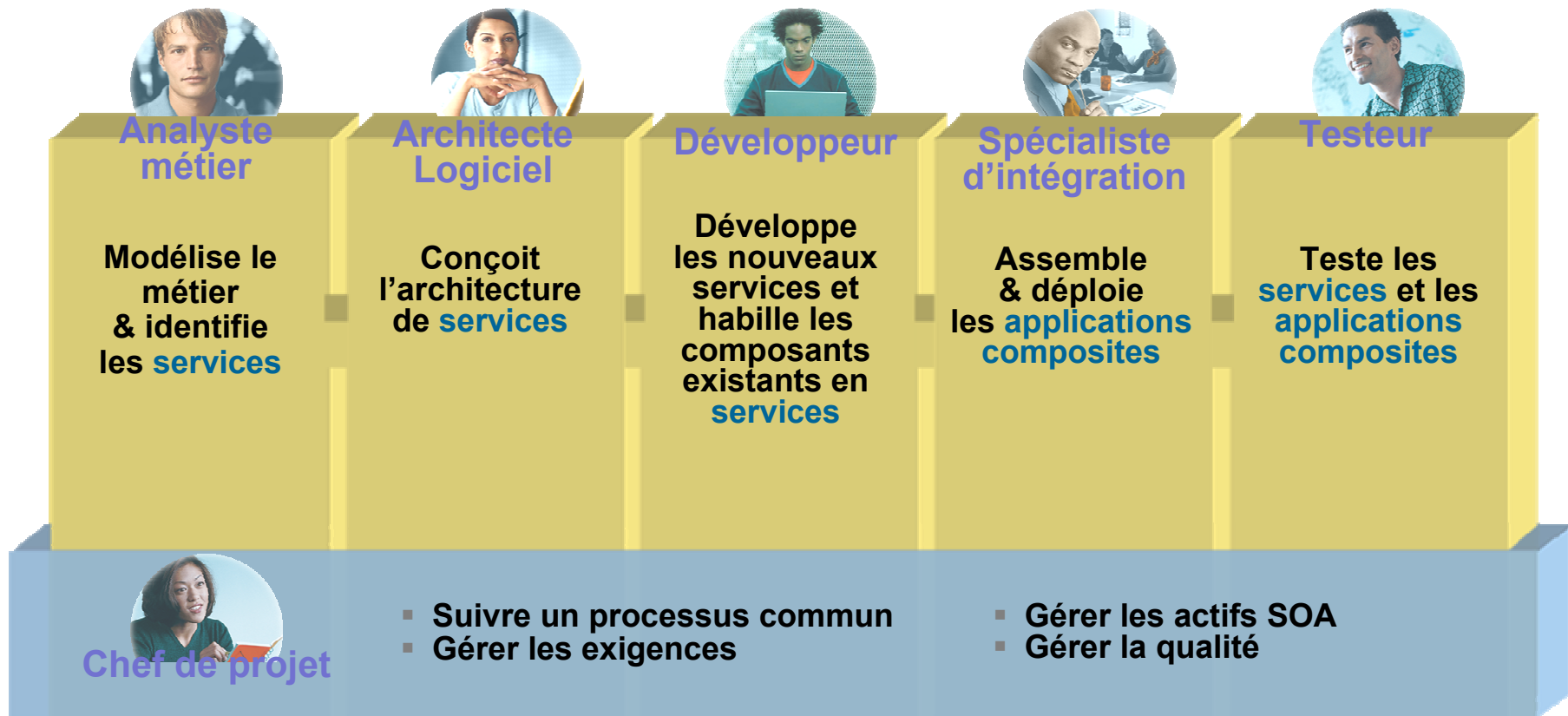
Aspects of the Enterprise Service Bus



Agenda

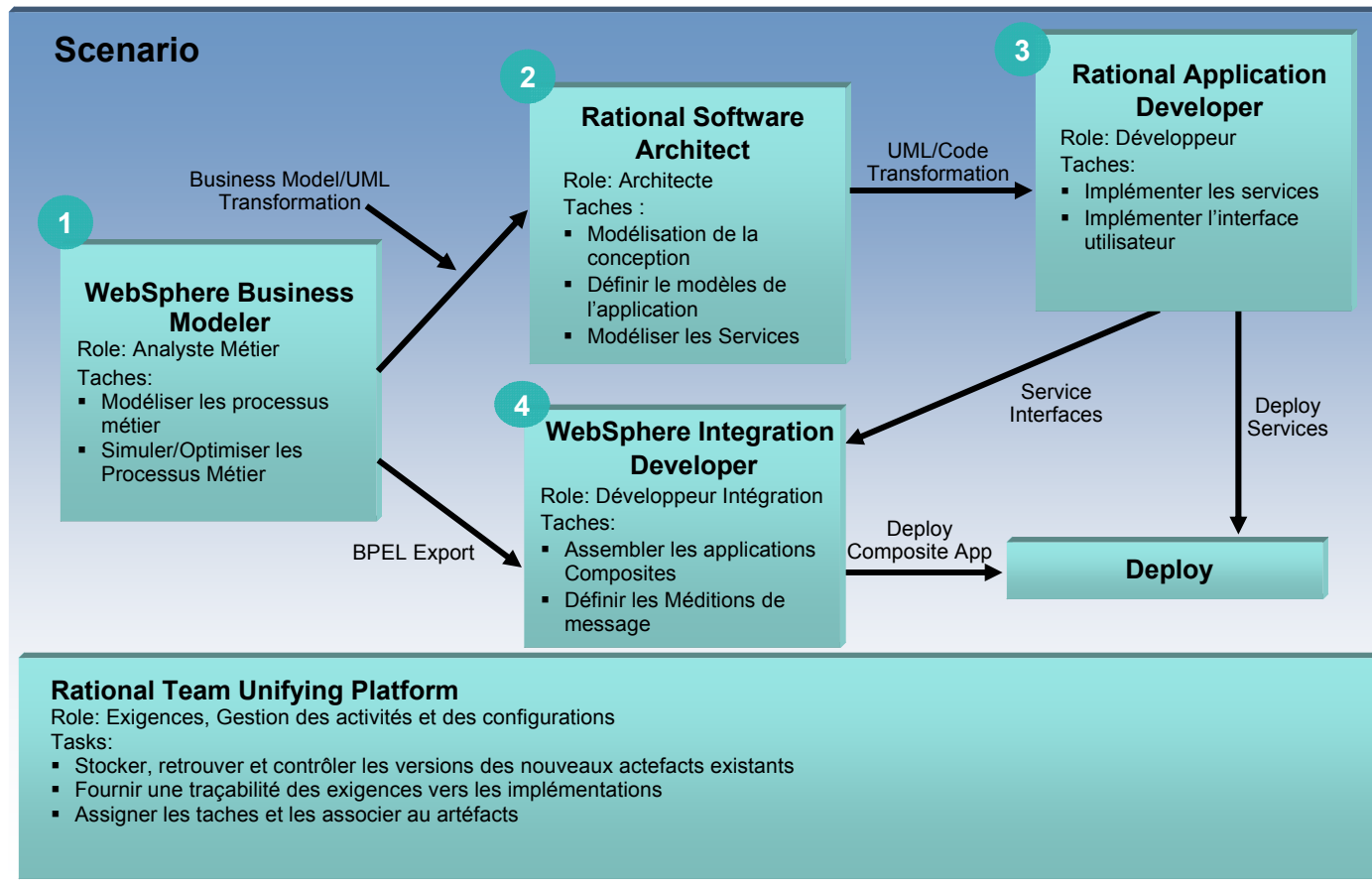
- Pourquoi SOA?
- C'est quoi SOA?
- SOA et la plateforme de développement IBM Rational
- Conclusion

La plate-forme d'IBM pour le développement SOA

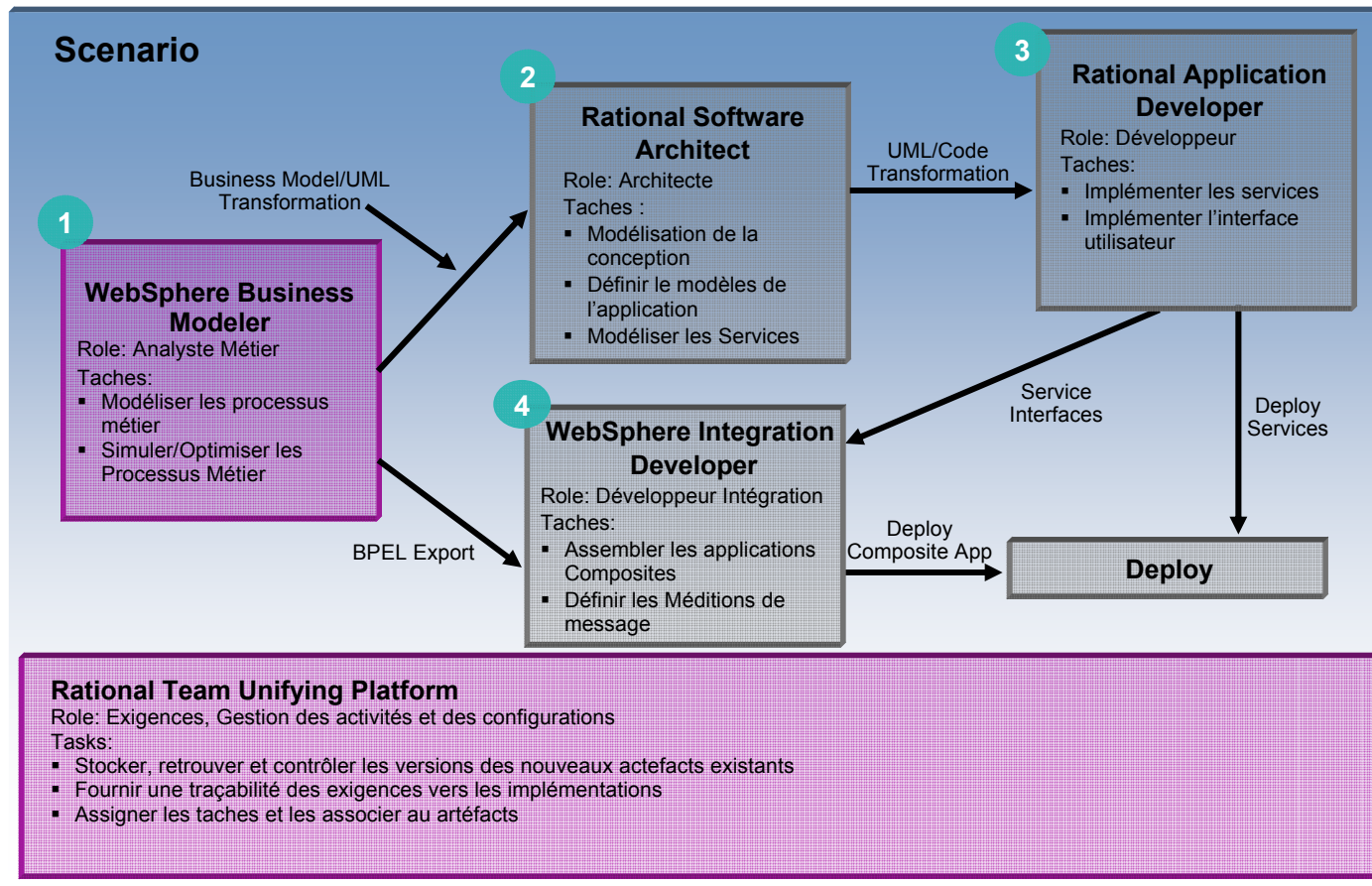


Basée sur IBM Rational Software Development Platform

Cas d'utilisation : Modéliser et Assembler



Cas d'utilisation : Modéliser et Assembler



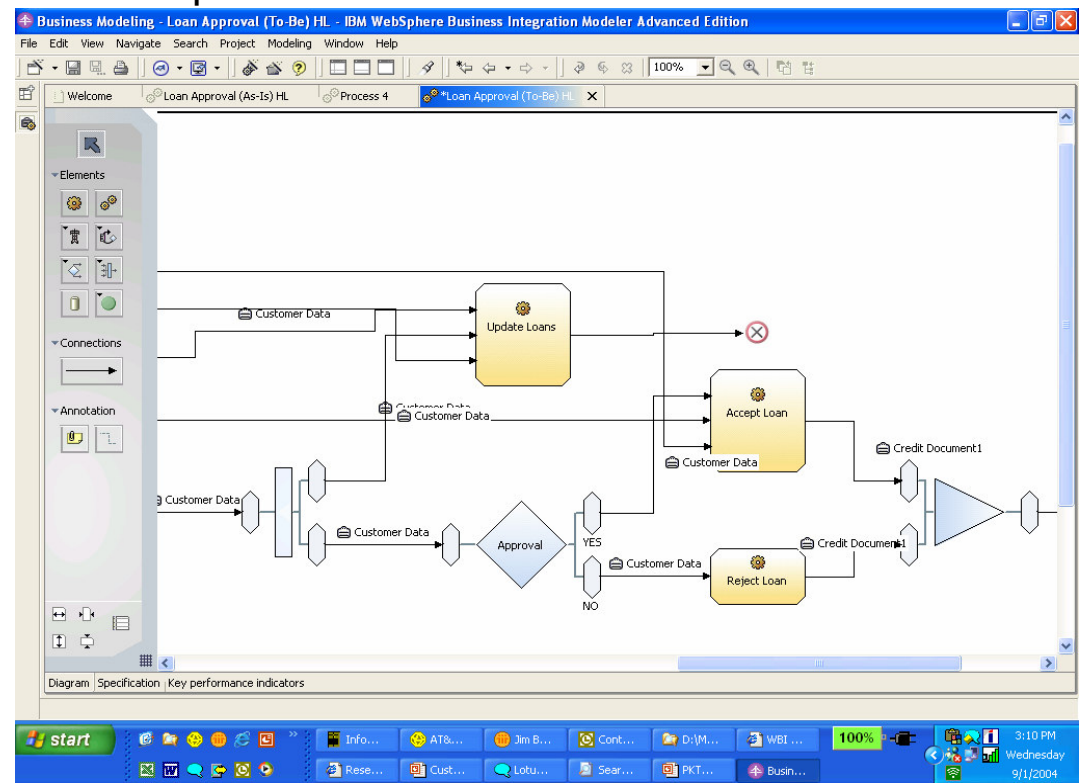
Modéliser, Simuler et Optimiser les Processus Métiers

- Capturer une représentation visuelle et analyser comment l'organisation fonctionne

- Avoir une compréhension claire et documentée des processus
- Evaluer les capacités des processus
- Détecter les problèmes et valider les solutions possibles
- Définir les mesures de performance

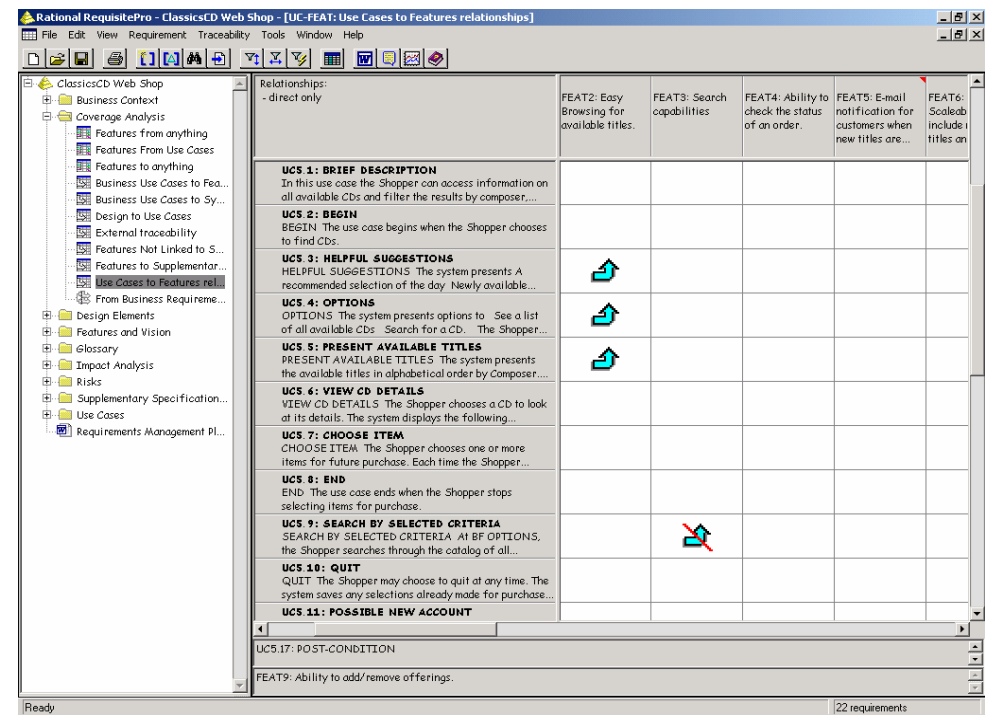
- Simuler

- Réaliser des simulations
- Modéliser les ressources
- Analyser leur niveau d'utilisation



Exigences Fonctionnelles & Non-fonctionnelles

- Mettre en place une hiérarchie et une classification des exigences
- Suivre les liens entre les exigences de haut niveau vers les exigences plus détaillées
- Réaliser une analyse de couverture pour s'assurer que les exigences de haut niveau sont prises en compte dans les exigences plus détaillées
- Faire de l'analyse d'impact lors de l'évolution d'une exigence



Modélisation des processus métier et traçabilité vers les cas d'utilisation

The image displays two screenshots from the IBM Rational Software Development Platform, illustrating the integration of business modeling and UML modeling.

Left Screenshot: Business Modeling - Assessor Determination

- Project Tree:** Shows a project structure with folders for 'ABC', 'ECAM', 'Business Items', and 'Processes'. The 'Assessor Determination' process is highlighted.
- Process List:** A detailed list of business processes, including:
 - Auto Claims Handling
 - Auto Claims Process
 - Auto Claims Submission
 - Assign New Auto Claim
 - Cancel New Auto Claim
 - Check Assessment Report
 - Close Auto Claim
 - Create New Auto Claim
 - Identify Available Assessors
 - Identify Claim Hot Spots
 - Identify Suitable Assessors
 - Initiate Payment and Repair
 - Investigate Auto Claim
 - Negotiate Settlement
 - Receive and Register Acceptance
 - Register New Auto Claim

Right Screenshot: Requirement - UML Model::Main - IBM Rational Software Development Platform

- Model Explorer:** Shows a hierarchical view of the UML model structure, including 'resources.XMI', 'ECAM', 'RootInformationModel', 'RootOrganizationModel', 'RootProcessModel', and 'Processes'. The 'Assessor Determination' process is highlighted.
- Requirement Explorer:** Shows a list of requirements and use cases, including:
 - Learning Project - Use Cases
 - Coverage Analysis
 - Eclipse Element Proxies
 - Features and Vision
 - Web Shop Administrative System
 - Web Shop System
 - Vision Document
 - All Features
 - INSTANCESPECIFICATIO1 Assessor
 - INSTANCESPECIFICATIO3 Assessor
 - Glossary
 - Impact Analysis
 - Supplementary Requirements
 - Use Cases
 - Arrange Shipment
 - Check Order Status
 - Purchase CD
 - Shop for CD
 - All Use Cases
 - Use-Case Brief Descriptions
 - UC5 Assessor Determination
 - UC6 Assign New Auto Claim
 - UC7 Auto Claims Process
 - UC8 Auto Claims Handling
 - README
 - Requirements Management Plan
- Diagram:** A UML Use Case diagram showing a relationship between a 'Business Use Case' (Assessor Determination) and a 'Use Case' (Assessor Determination). The 'Business Use Case' is represented by a solid oval, and the 'Use Case' is represented by a dashed oval. A dashed line with an open arrowhead points from the 'Use Case' to the 'Business Use Case', indicating a realization relationship.

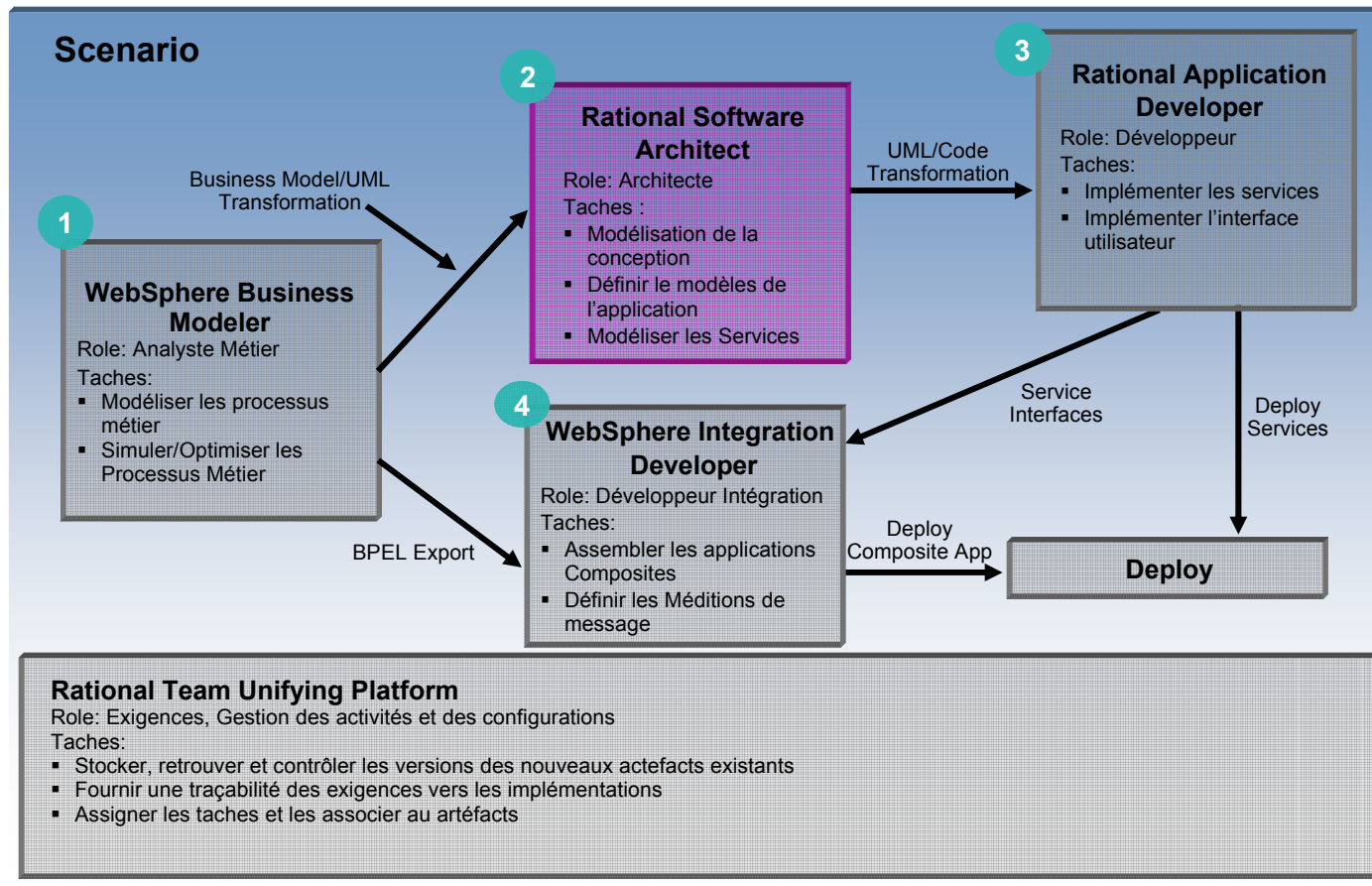


Demo

Documenter les processus Métier
Capturer les exigences fonctionnelles et
non fonctionnelles

WBI Modeler, RequisitePro et RSA

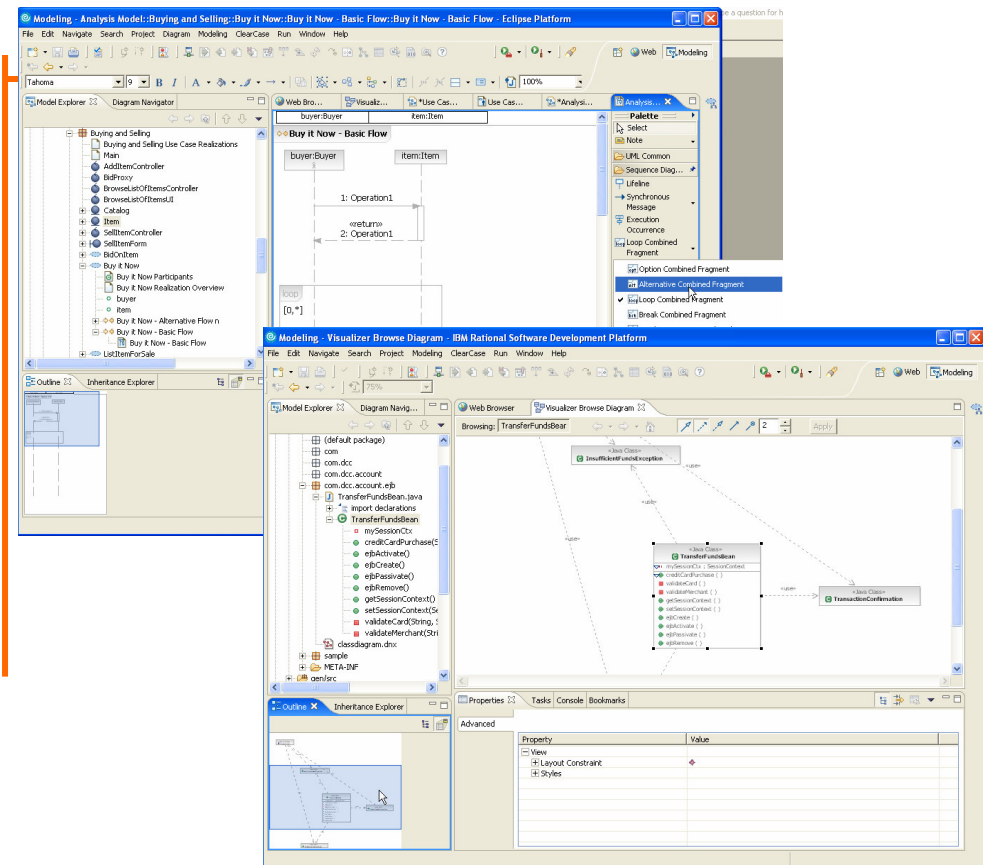
Cas d'utilisation : Modéliser et Assembler



Conception des Services

Rational Software Architecture

- Modélisation UML2 et spécification de l'architecture
- Enrichement de la conception à base de Patterns
- Transformation de modèle et génération de code



Bénéfice:

- Architecture et conception de l'implémentation des services
- Automatisation de l'application des design patterns
- Automatiser les transformations de l'analyse et la conception vers l'implémentation

Qu'est-ce qu'un Pattern ?

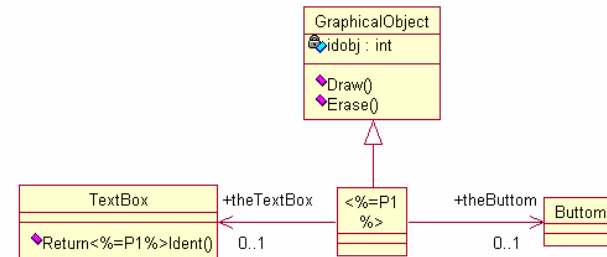
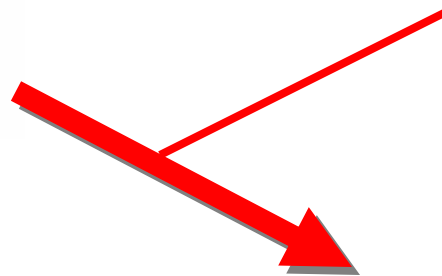
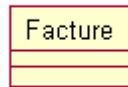
“ A pattern provides a common solution to a problem in a specific context”

Grady Booch

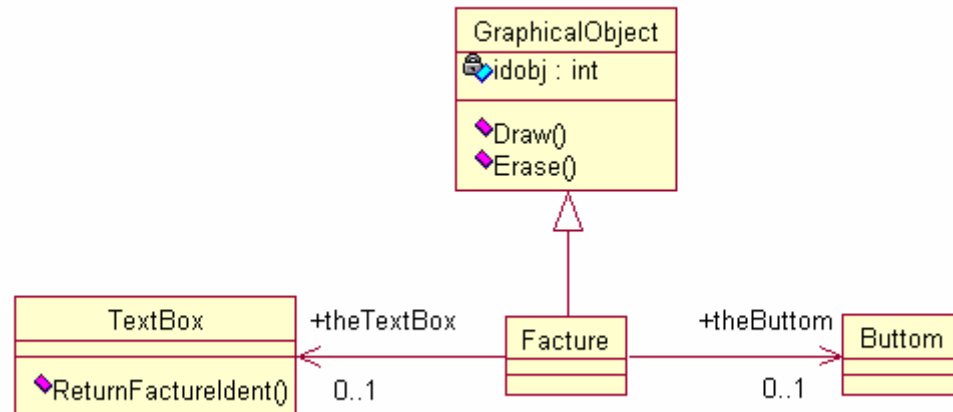
“ A design pattern describes the problem, a solution to the problem consisting of a general arrangement of objects and classes, when to apply the solution, and the consequences of applying the solution” **Gamma et. al.**

Obtenir une architecture évolutive et réutilisable

Pattern



Réutilisation de savoir-faire



Définition et application interactive des Patterns

Rational Software Architect

- Reusable Asset Specification (RAS)
 - Un moyen standard de paquager les assets
 - Décrire la structure et la nature des assets
 - Réduire la friction sur la réutilisation en définissant un standard de packaging
- GOF design pattern
- Patterns supplémentaires fournis via un repository RAS sur developerWorks
 - Exemples: SOA / Web Service design patterns, IBM WebSphere patterns

Supporte le “drag-and-drop” de paramètre de pattern

Apply Pattern "Proxy"

Specify values for pattern parameters.

Parameter Name	Type	Multiplicity	Value
Proxy	Class	1	PrinterServerProxy
Subject	Interface	1	PrinterServer
RealSubject	Class	1	PrinterServer

Proxy parameter description
 Proxy maintains a reference that lets the proxy access the real subject. Proxy may refer to a Subject if the RealSubject and Subject interfaces are the same. It provides an interface identical to Subject's so that a proxy can be substituted for the real subject. It controls access to the real subject and may be responsible for creating and deleting it.

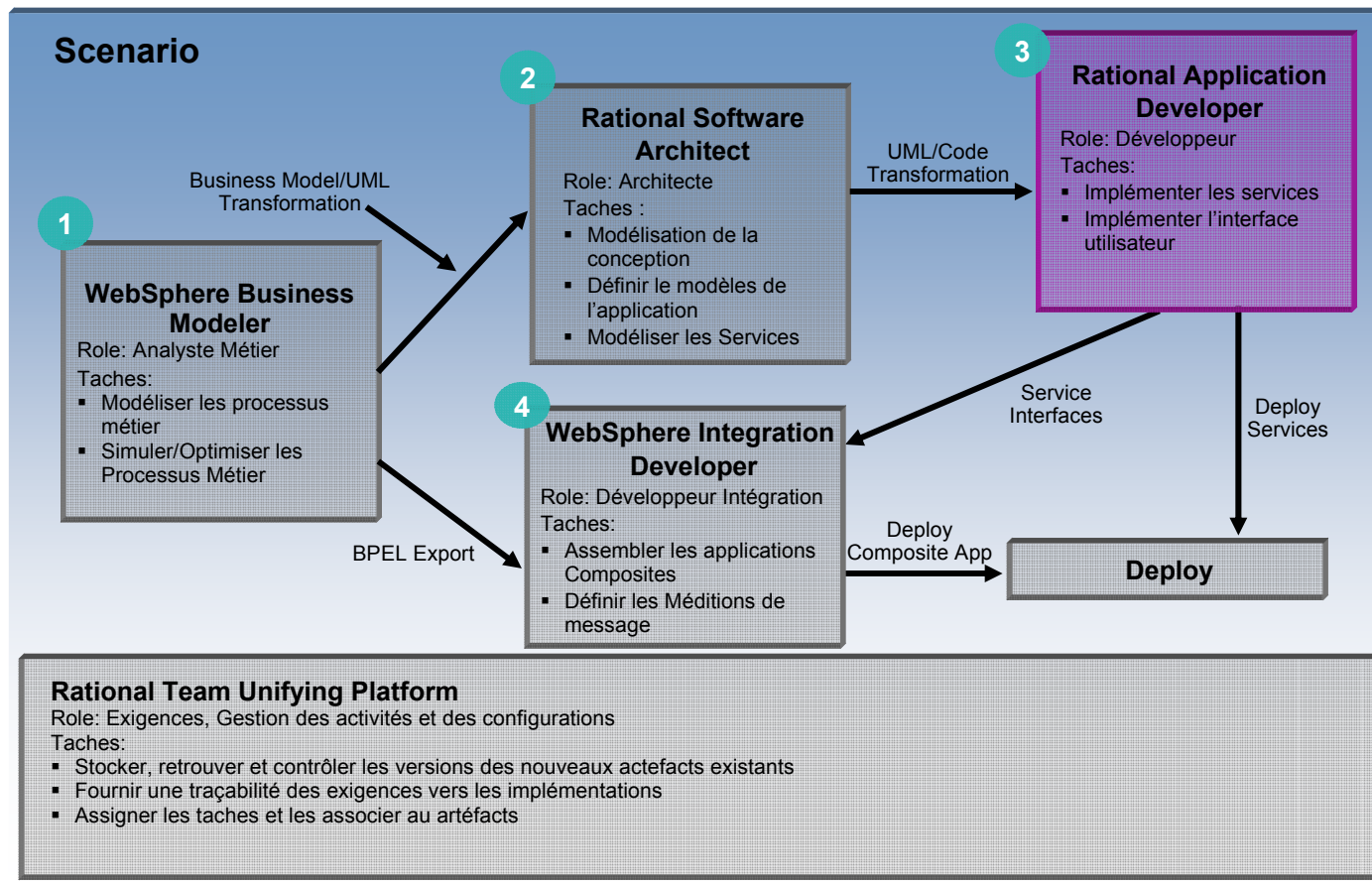
• Nouvelle vue “Patterns” fournissant une description



Demo

Modélisation - Architecture et
Conception Rational Software Architect

Cas d'utilisation : Modéliser et Assembler



S'assurer de la qualité du code très tôt dans le cycle

- **Développement à base de diagrammes UML**
- **Revue du code automatique**
 - Par règles de programmation avec proposition de correction
- **Automatisation du test de composant**
 - Exécution cas de test pour classes Java, EJB, Web Services
- **Analyse et profiling à l'exécution**
 - Analyse des performances et couverture du code
 - Graphe d'appel, trace visuelle
- **Test unitaire et mise au point pour WAS, WebSphere Portal, Tomcat et BEA WebLogic.**

Bénéfices:

- Améliorer la qualité du code
- Améliorer les performances et la fiabilité des applications
- Raccourcir les cycles de test

The screenshot displays two windows from the Eclipse IDE. The top window, titled 'Java - ComparisonFailure.java - Eclipse Platform', shows a code review for 'ComparisonFailure.java'. The code includes a method for comparing strings, with a 'Code Review Details' panel on the right showing a 'High' severity problem in 'Globalization:Character (Score 20)'. The bottom window, titled 'Test - BidHelperTestBehavior.java - IBM WebSphere Studio Application Developer', shows the 'testMethods()' method in 'BidHelperTestBehavior.java'. Below the code, a 'Test Data Table' is visible, containing test data for various parameters like 'itemTypeId', 'bidAmount', 'bidIncrement', 'maxBid', 'userId', and 'currency'.

Action	Type	default	IN	OUT
oneBidHelperLocal_1 = oneBidH...				
oneBidHelperLocal_1.bidItem(it...				
itemTypeId	Integer		100546	
bidAmount	Long		100	
bidIncrement	Integer		1	10
maxBid	Long		
userId	Integer		83301	
currency	int		3	
ExpectedException	Throwable			

Construire facilement des interfaces Web et portail

Pointer/cliquer pour la connexion aux services

Glisser/Déposer des composants riches

Conception visuelle WYSIWYG

Édition de propriétés

Page Designer avec support Java Server Faces et SDO

Bénéfices:

- Efficacité accrue de part la simplicité de l'approche.
- Prise en main rapide pour des non-initiés.
- Réduction du risque d'erreur de codage.

Découvrir, Créer et Consommer les Web Services

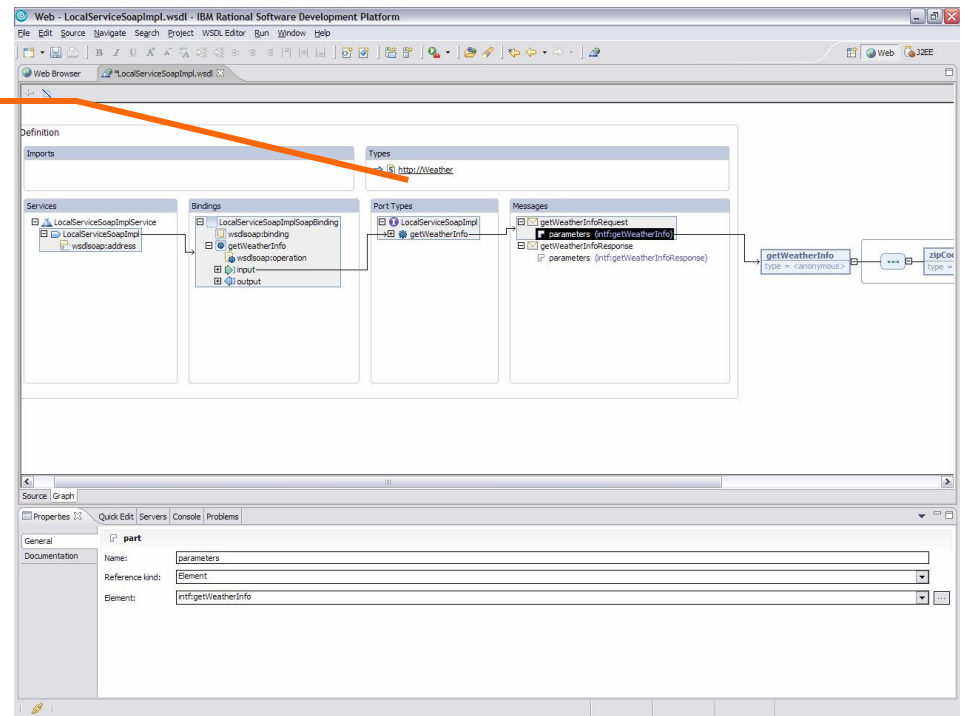
Rational Application Developer

- Outils pour faciliter la découverte, la création, la génération, le teste, le déploiement et la publication des services Web

- Construire de nouveaux services Web à partir de rien ou à partir d'applications existantes.

- Conformité à WS-I

- Découvrir et consommer des services Web existants



WSDL Editor

Bénéfice:

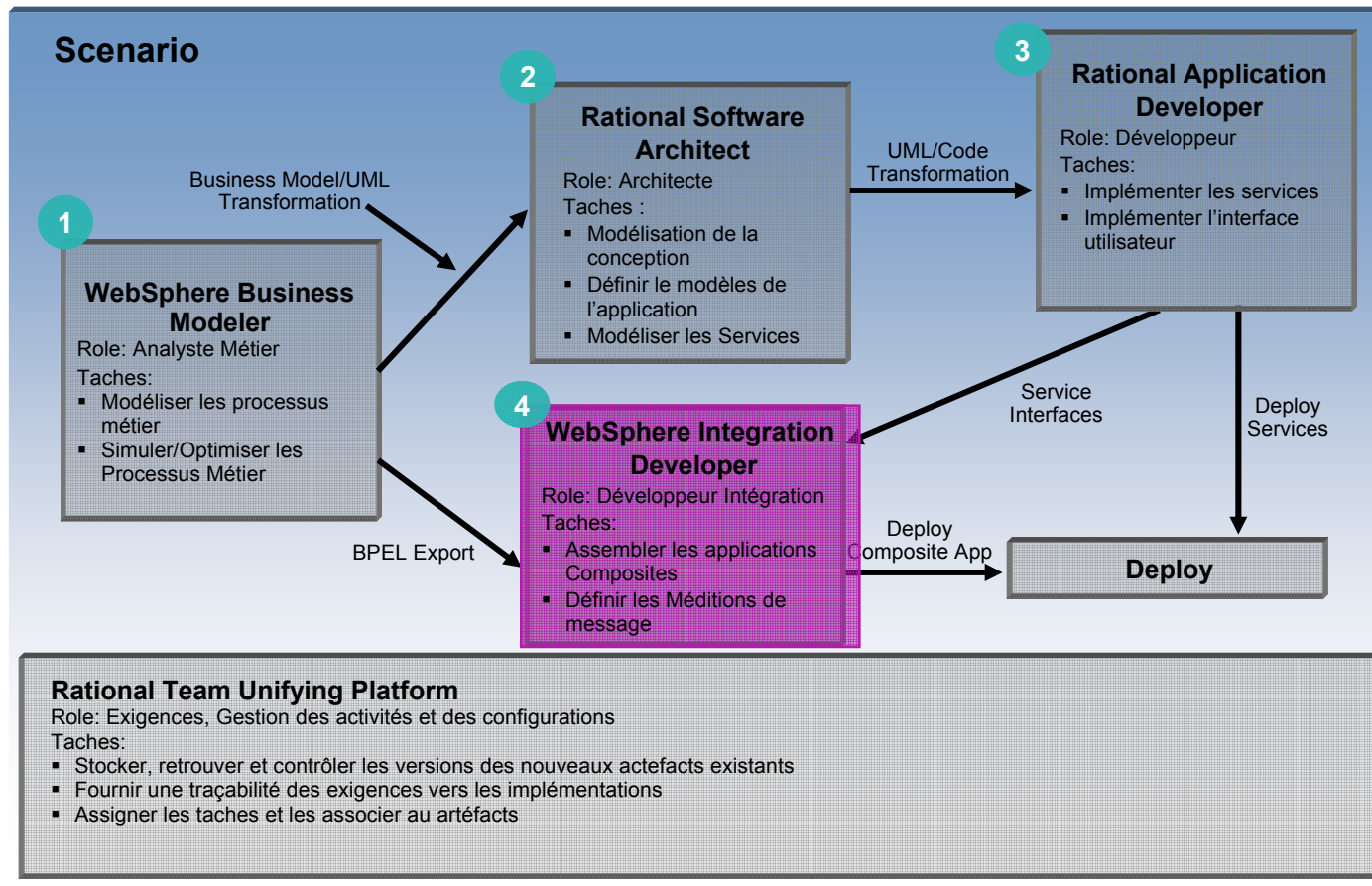
- Requièrè peu voir pas de codage pour les assets existants
- Faible courbe d'apprentissage
- Forte productivié



Demo

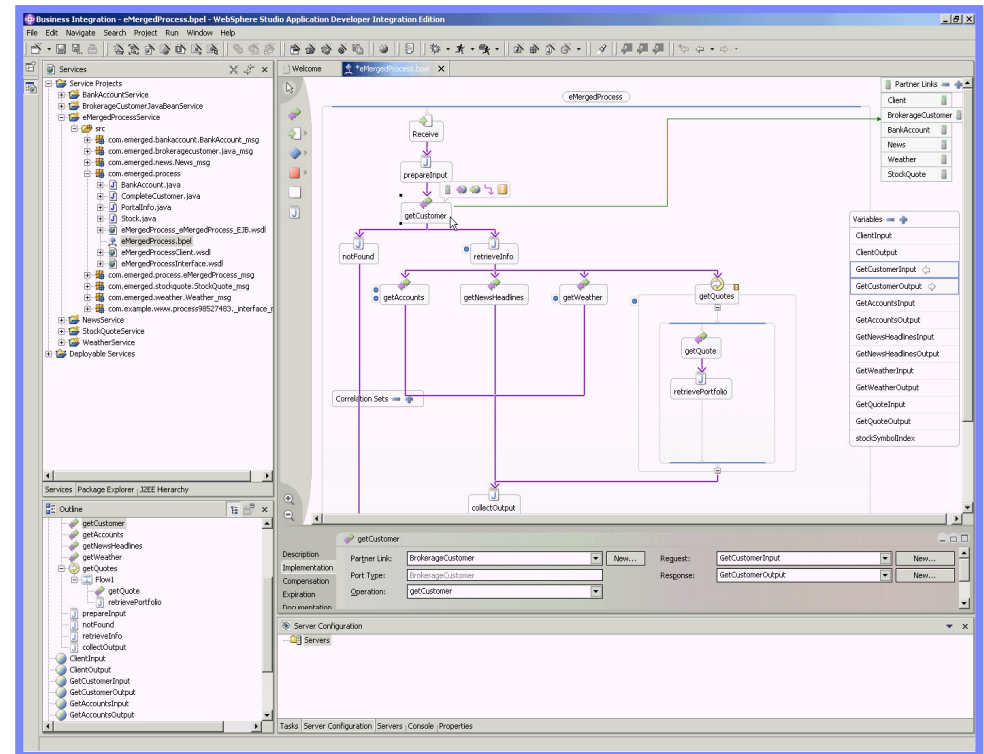
Implementer les Services et les
interfaces homme machine
Rational Application Developer

Cas d'utilisation : Modéliser et Assembler



Process Choreography

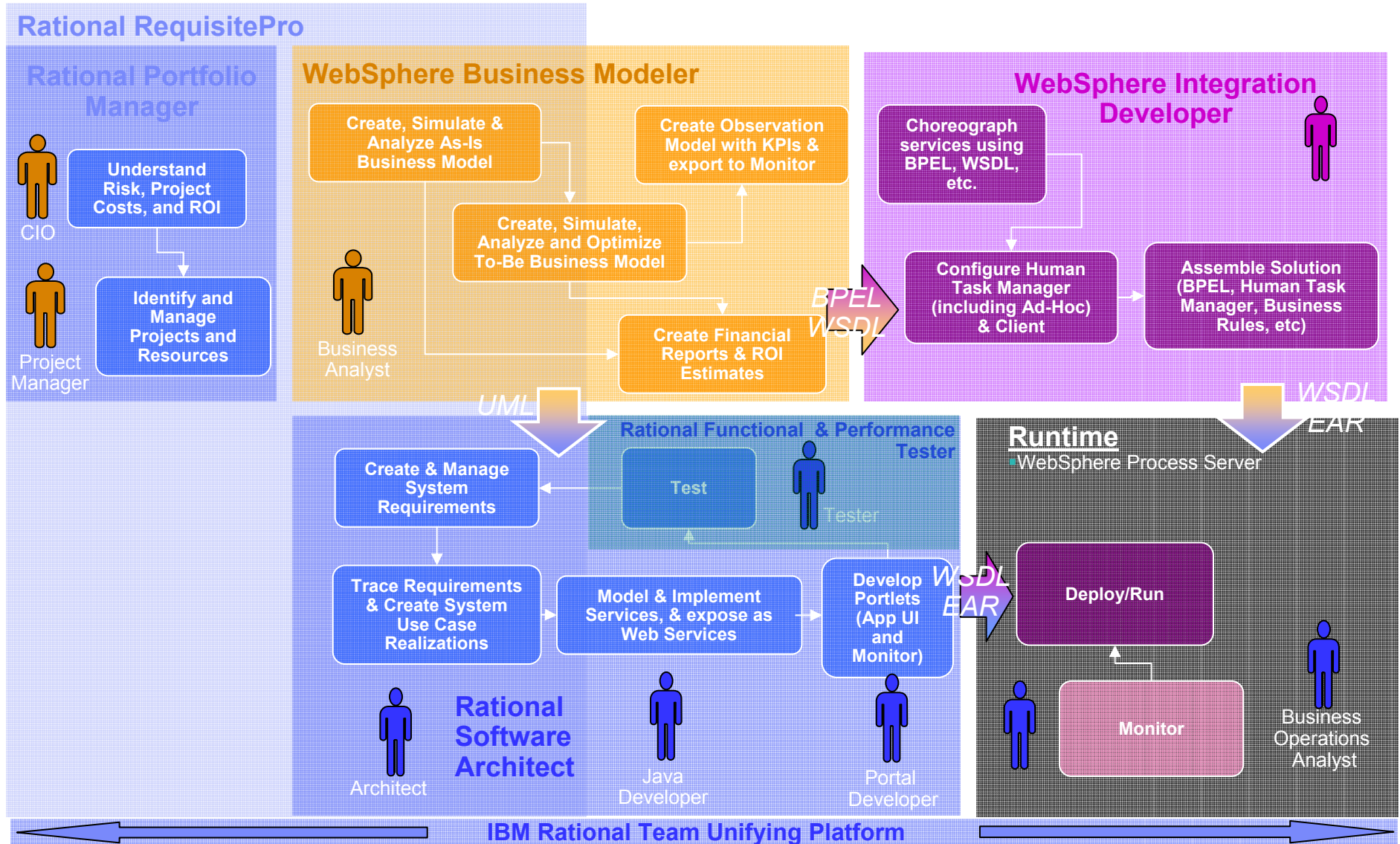
- Utiliser le BPEL exporté par l'activité de modélisation des processus
- Inclure le support des workflows humain
- Découvrir les services web publiés vers l'annuaire UDDI
- Mettre en place le mapping entre les processus



Agenda

- Pourquoi SOA?
- Qu'est ce que SOA?
- SOA et la plateforme de développement IBM Rational
- Conclusion

Comment développer



Questions