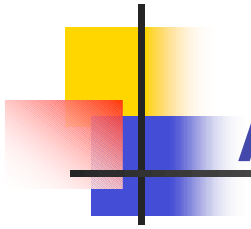
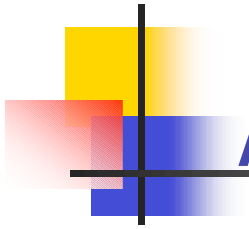
Leveraging Existing Systems into a Service Oriented Architecture

Michel Brassard
Senior Architect
brassardmichel@gmail.com



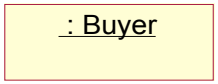
Agenda

- Abstracting SOA with UML and MDA
- Leveraging existing applications with SOA
- Key OMG initiative on SOA



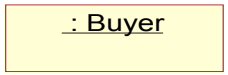
Abstracting SOA with UML and MDA

- First scenario (running example): Credit accepted



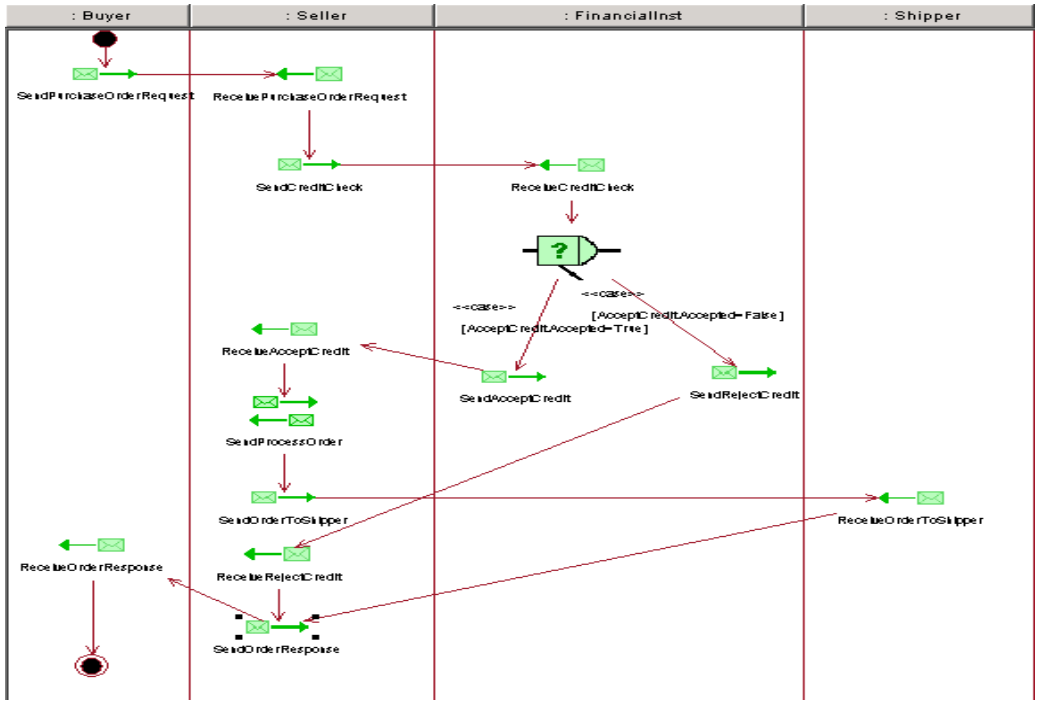
Abstracting SOA with UML and MDA

- Second scenario: Credit rejected



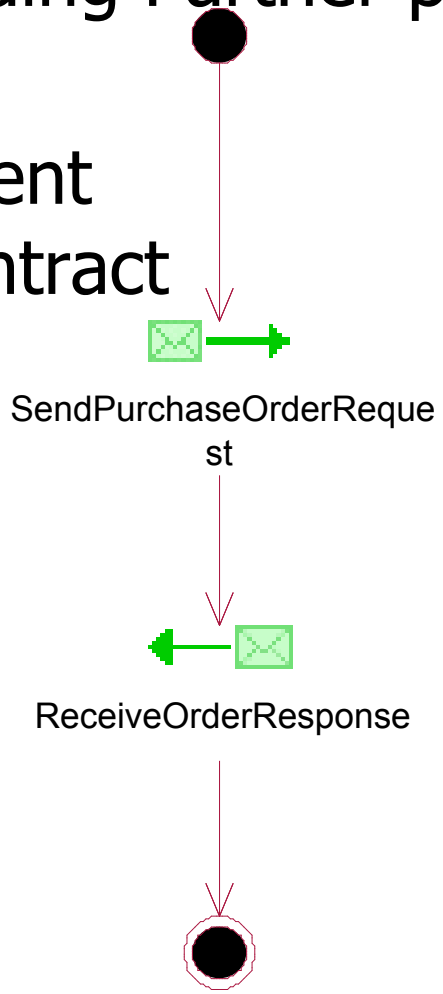
Abstracting SOA with UML and MDA

- Business Scenarios Integration:
  - Automatic synthesis of the global view of the business process:
    - Generates the Behavior of the Global View of the Business Process



Abstracting SOA with UML and MDA

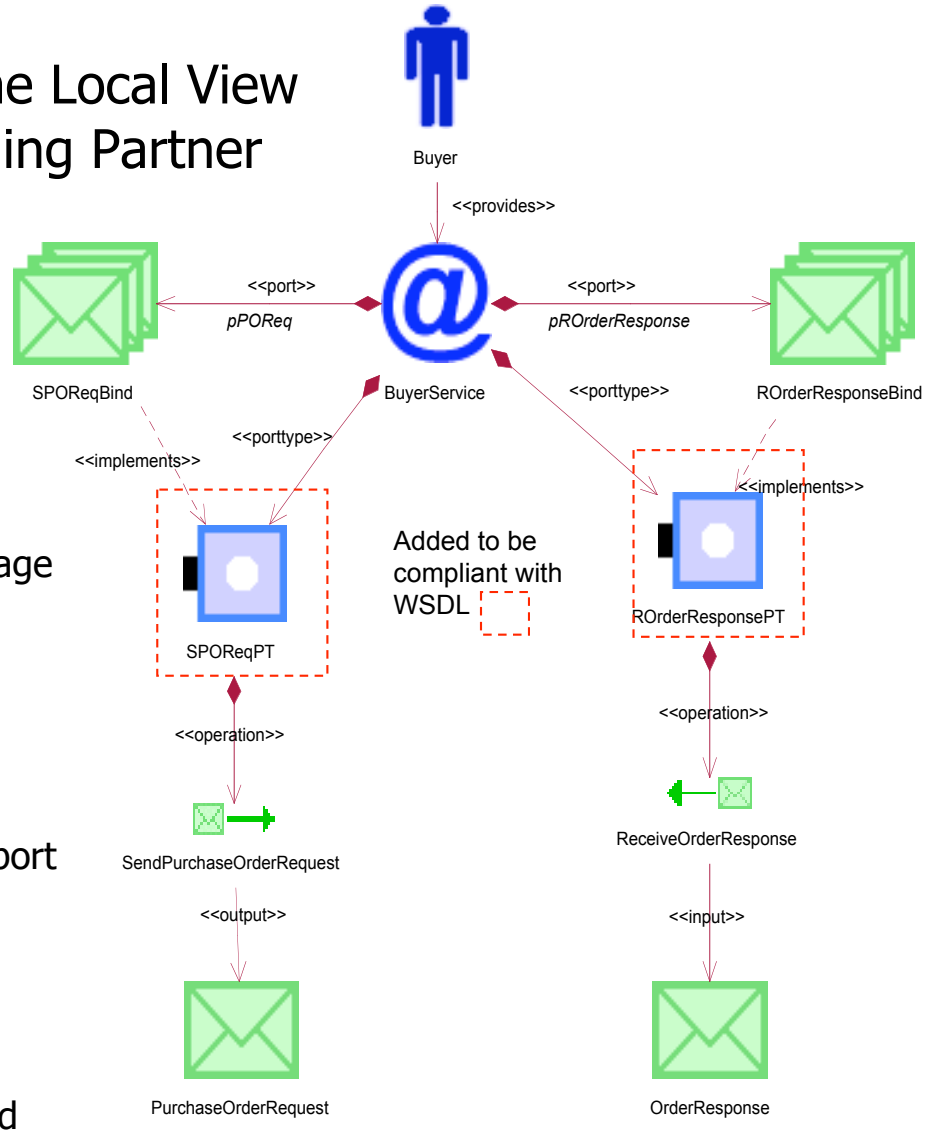
- Generates the Behavior of the Local View (Web Services) of a Trading Partner playing a Role in the Business Process
- Provides a consistent view of the contract among trading partners



Abstracting SOA with UML and MDA

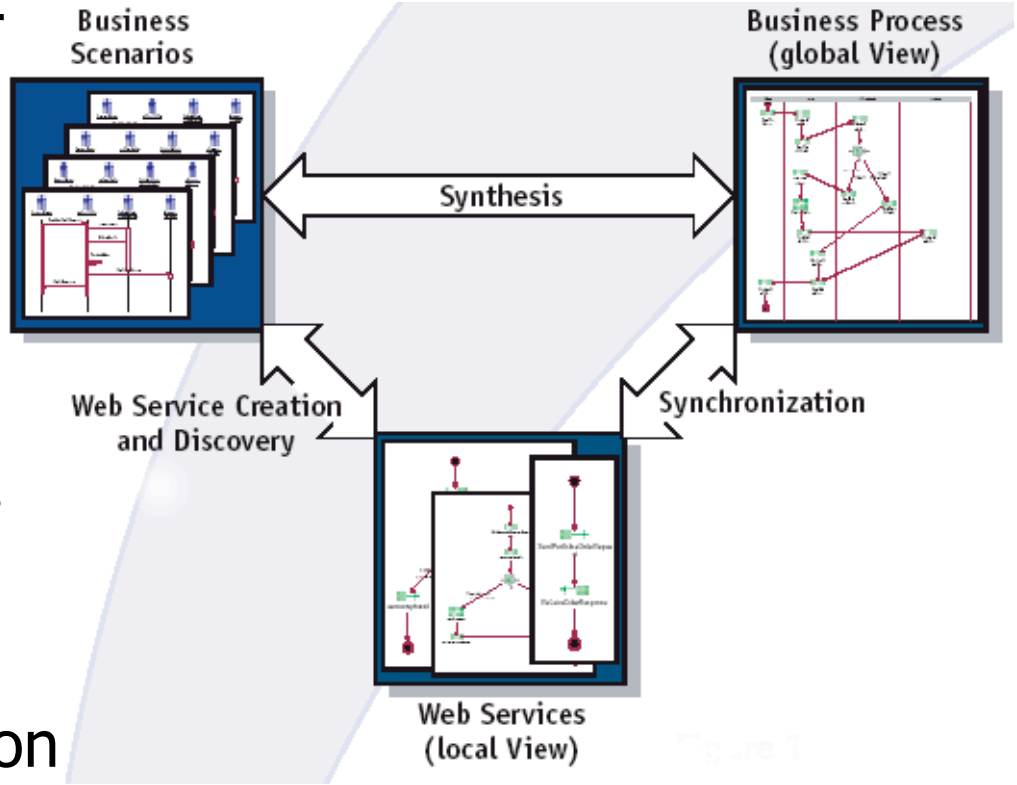
- Generating the Structure of the Local View (Web Services) of a Trading Partner

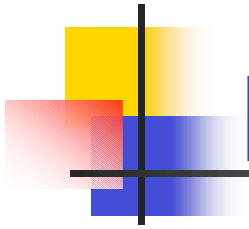
- Binds operations of a business process with messages
  - SendPurchaseOrderRequest with the message PurchaseOrderRequest
  - ReceiveOrderResponse with the message OrderResponse
- Binds operations with a port implementation (manual step)
  - SendPurchaseOrderRequest with the port pPOReq implented by the message queuing SPOReqBind
  - ReceiveOrderResponse with the port pOrderResponse implemented by the message queuing ROrderResponseBind



Abstracting SOA with UML and MDA

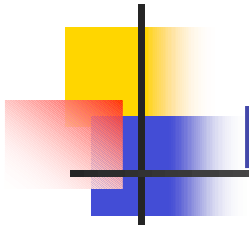
- Provides a UML platform for modeling:
  - Business scenarios
  - Business Processes
  - Business Documents
  - Web Services
- UML profile for
  - Modeling business documents
  - Modeling business document transformation
  - Modeling business processes
- Model-to-Text Transformation for the creation and the reverse engineering of the BizTalk schedule and XML documents





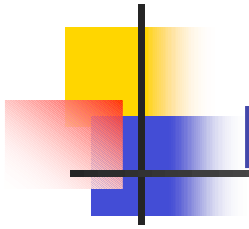
Part 2

- Abstracting SOA with UML and MDA
- Leveraging existing applications with SOA
- Key OMG initiative on SOA



Leveraging existing applications with SOA

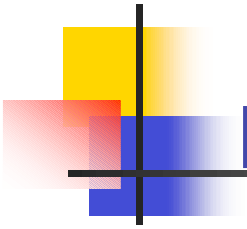
- Major challenges faced by companies:
  - Aligning Business Processes with IT
  - Enforce Compliance
  - Eliminate Duplication in Business Functionalities Among Similar Systems



Leveraging existing applications with SOA

- Functional Architecture will help consolidate your application portfolio
  - System B is the strategic one
  - System Z is obsolete
  - System A and C will be phased out respectively Feature 3 and 1

Business	Existing System				
Feature	System A	System B	System C	...	System Z
Feature 1	x		x		
Feature 2		x	x		x
Feature 3	x	x			
…					
Feature n		x			x

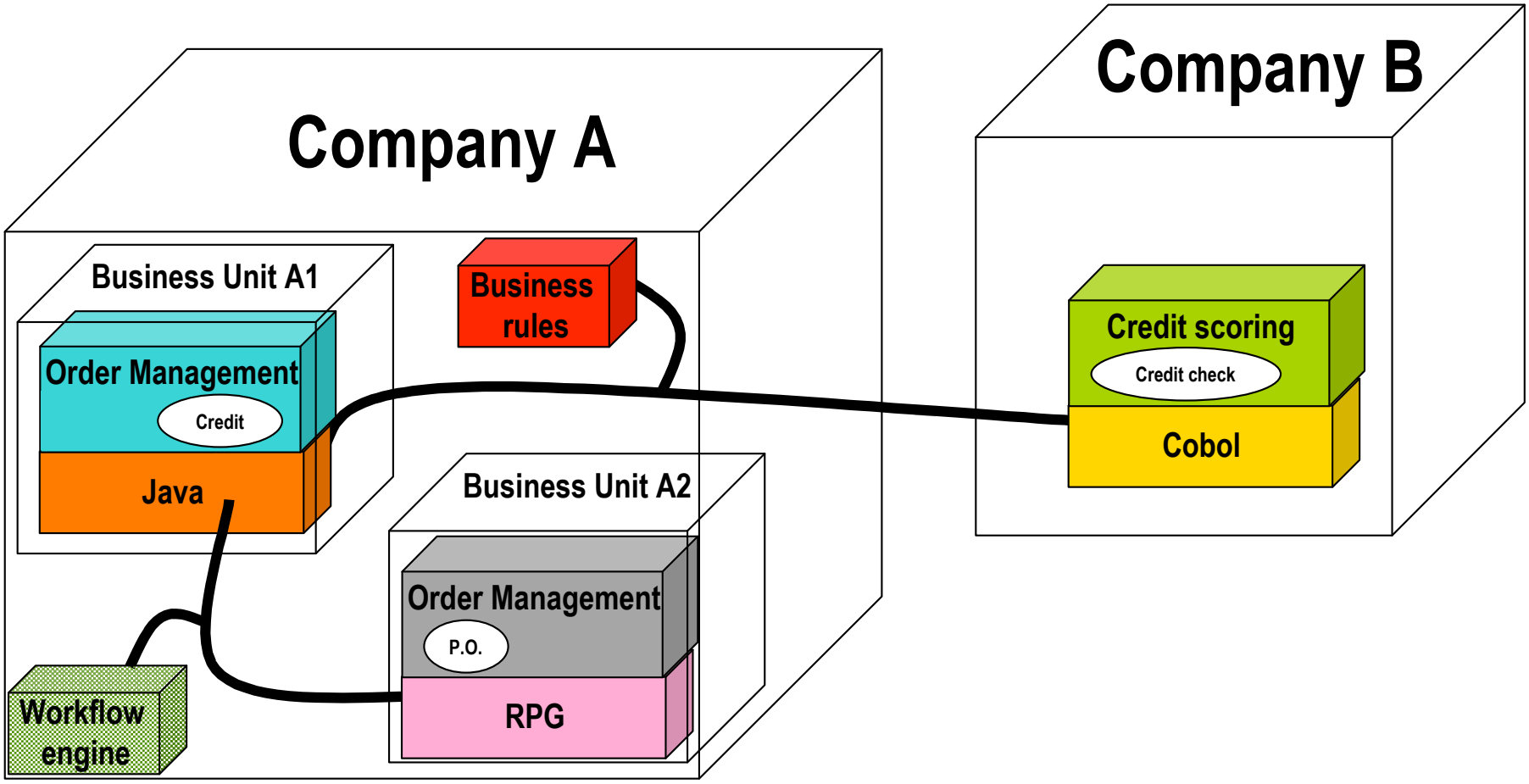


Leveraging existing applications with SOA

- Heterogeneity is permanent
  - No consensus on hardware platforms
  - No consensus on operating systems
  - No consensus on network protocols
  - No consensus on programming languages
  - No consensus on Middleware
  - No consensus on workflow engines
- Infrastructure software is permanently changing

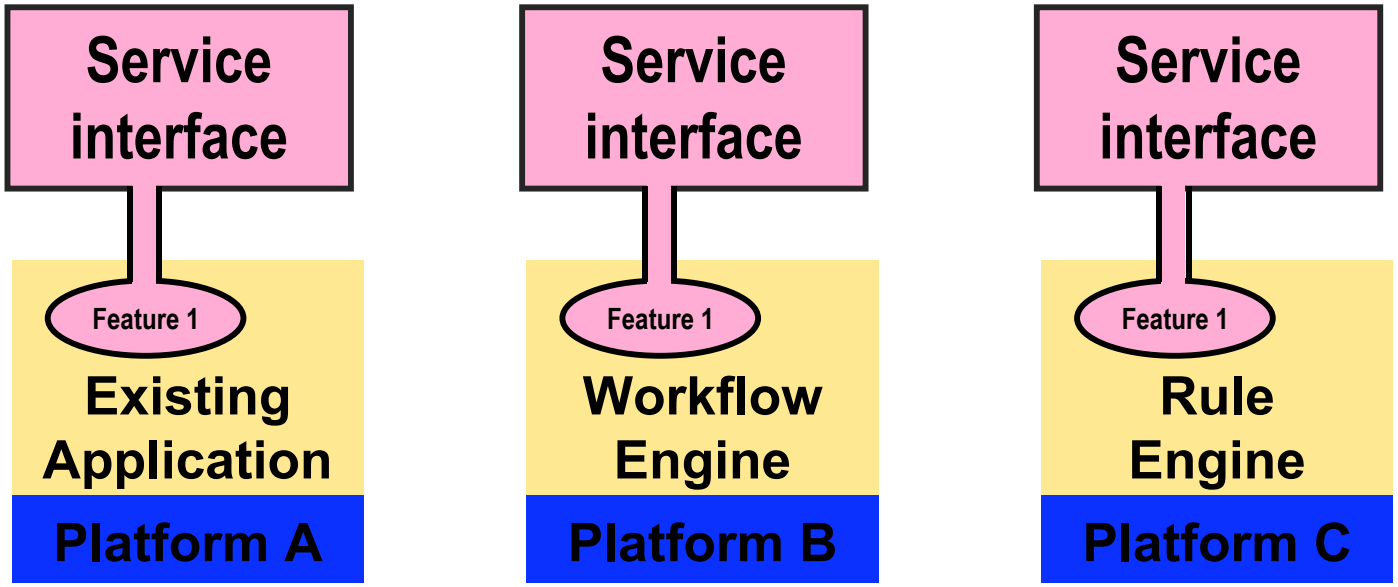
Leveraging existing applications with SOA

- Transition to SOA requires an Enterprise Service Bus



Leveraging existing applications with SOA

- Each node on the Enterprise Service Bus is:
  - Loosely-coupled with other nodes
  - Replaceable by one or more nodes compliant with its service interface and could have:
    - Different physical location
    - Different implemented technology
    - Different execution platform



Leveraging existing applications with SOA

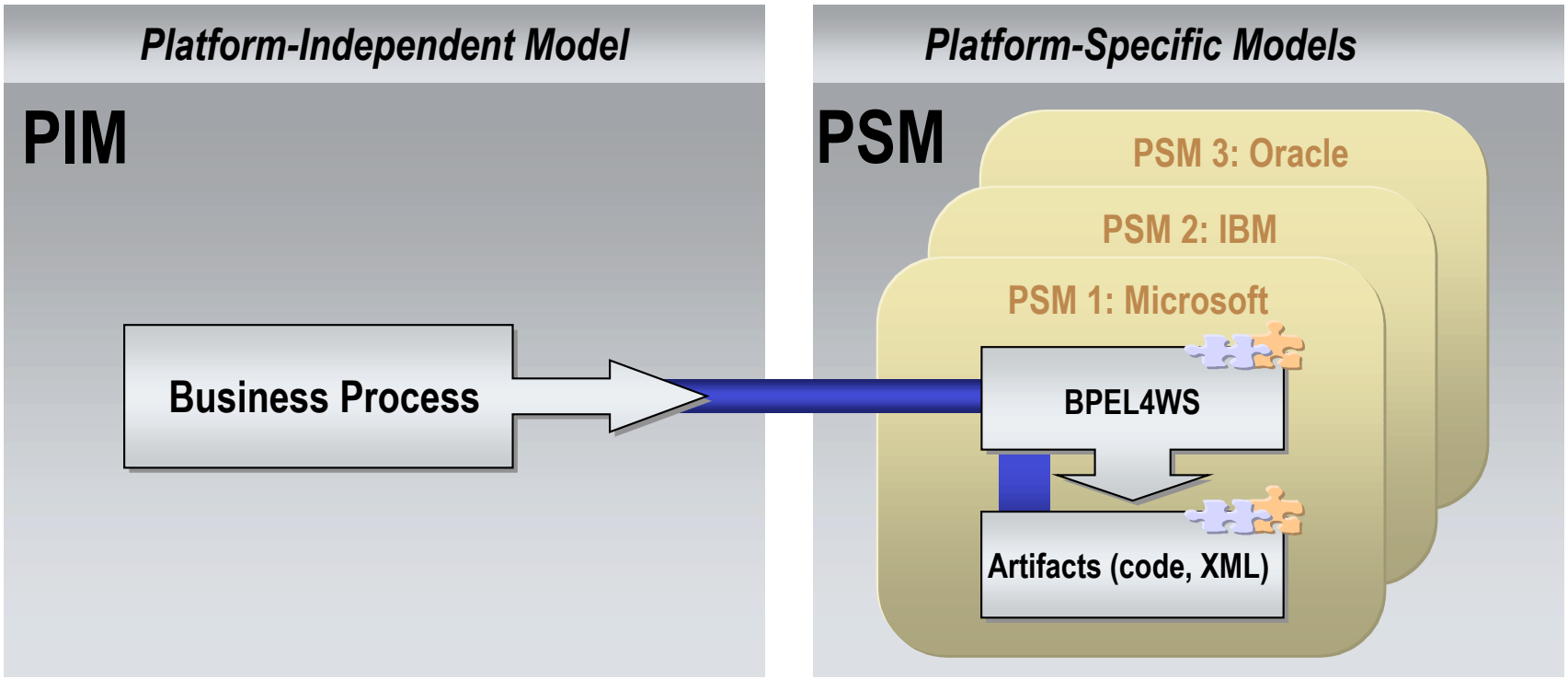
- Web services orchestration can be implemented by:
  - Hard coding it in a programming language
  - BPEL workflow engine

	Workflow Engine	External Trading Partners
Homogeneous	Vendor's offering	Vendor's offering
Heterogeneous	Vendor's offering	Vendor's offering

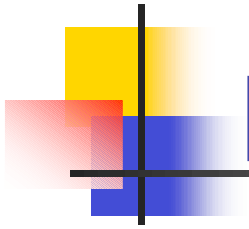
Good fit for UML

Leveraging existing applications with SOA

- Model-to-Model and Model-to-Text are applicable to any SOA environment
- Loosely coupled implementation for a single Business Process

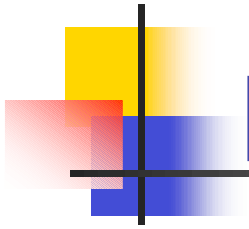


Trading partners build Web services using a .NET or Java platform and different vendors for their workflow engine.



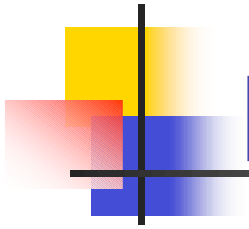
Part 3

- Abstracting SOA with UML and MDA
- Leveraging existing applications with SOA
- Key OMG initiative on SOA



Key OMG initiative on SOA

- Business Process Execution Language for Web Services (BPEL4WS) had a major influence on UML 2.0
- Business Process Definition Metamodel(BPDM) Specification (revision state)
  - Is based on a constrained subset of UML 2.0
  - Mapping to Business Process Modeling Notation(BPMN) enables a friendly user interface over the standard look-and-feel of a UML 2.0 modeling tool to business users

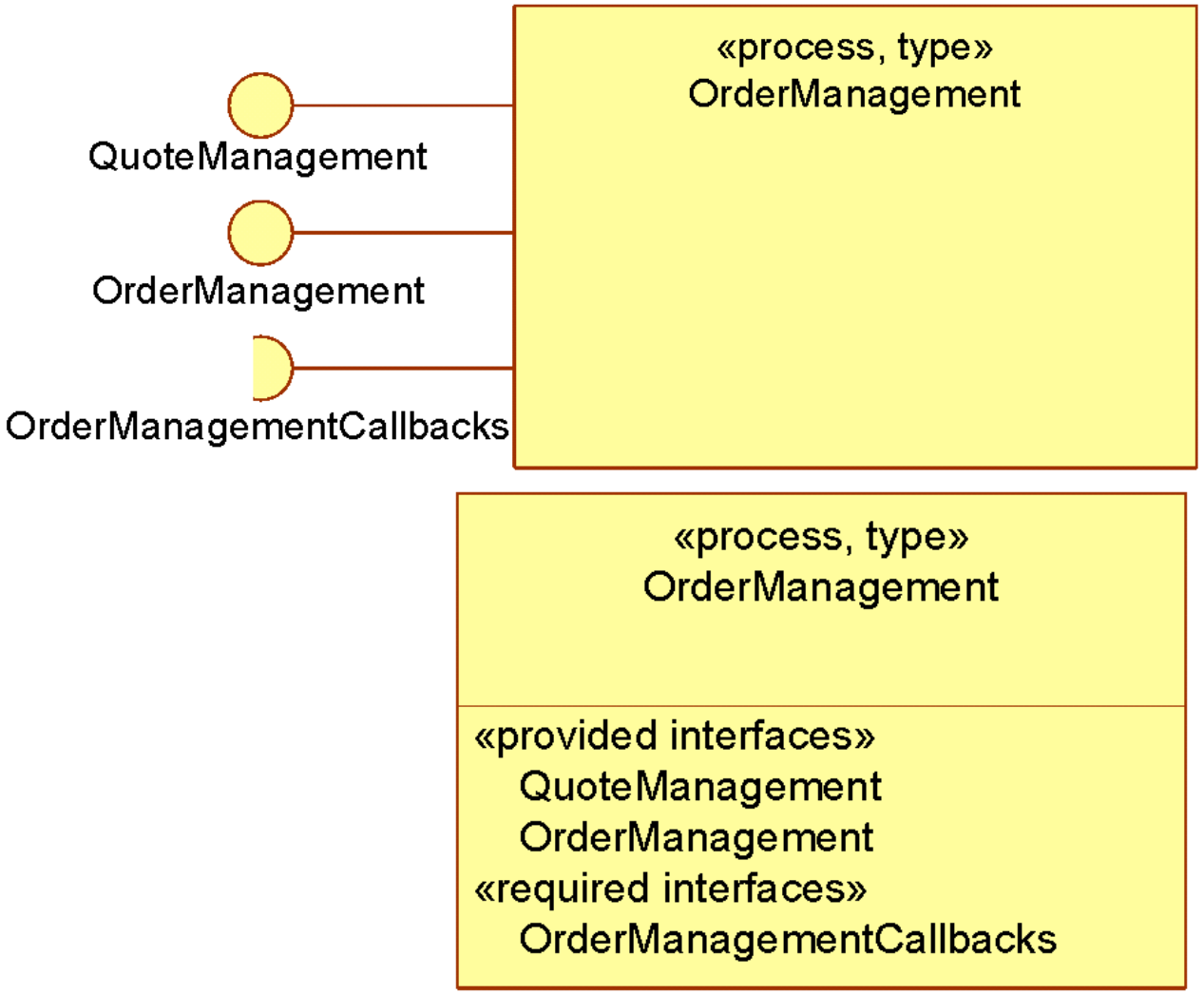


Key OMG initiative on SOA

- BPDM Specification should reuse the following UML 2.0 subset:
  - UML 2.0 Activities and Actions are used for the business process definitions and choreographies.
  - UML 2.0 Collaborations and Roles are used to specify interaction patterns between partners.
  - UML 2.0 Templates are used to permit the use of templates in business process definitions.
  - UML 2.0 Composite Structures and Activities are used to represent composite business processes.

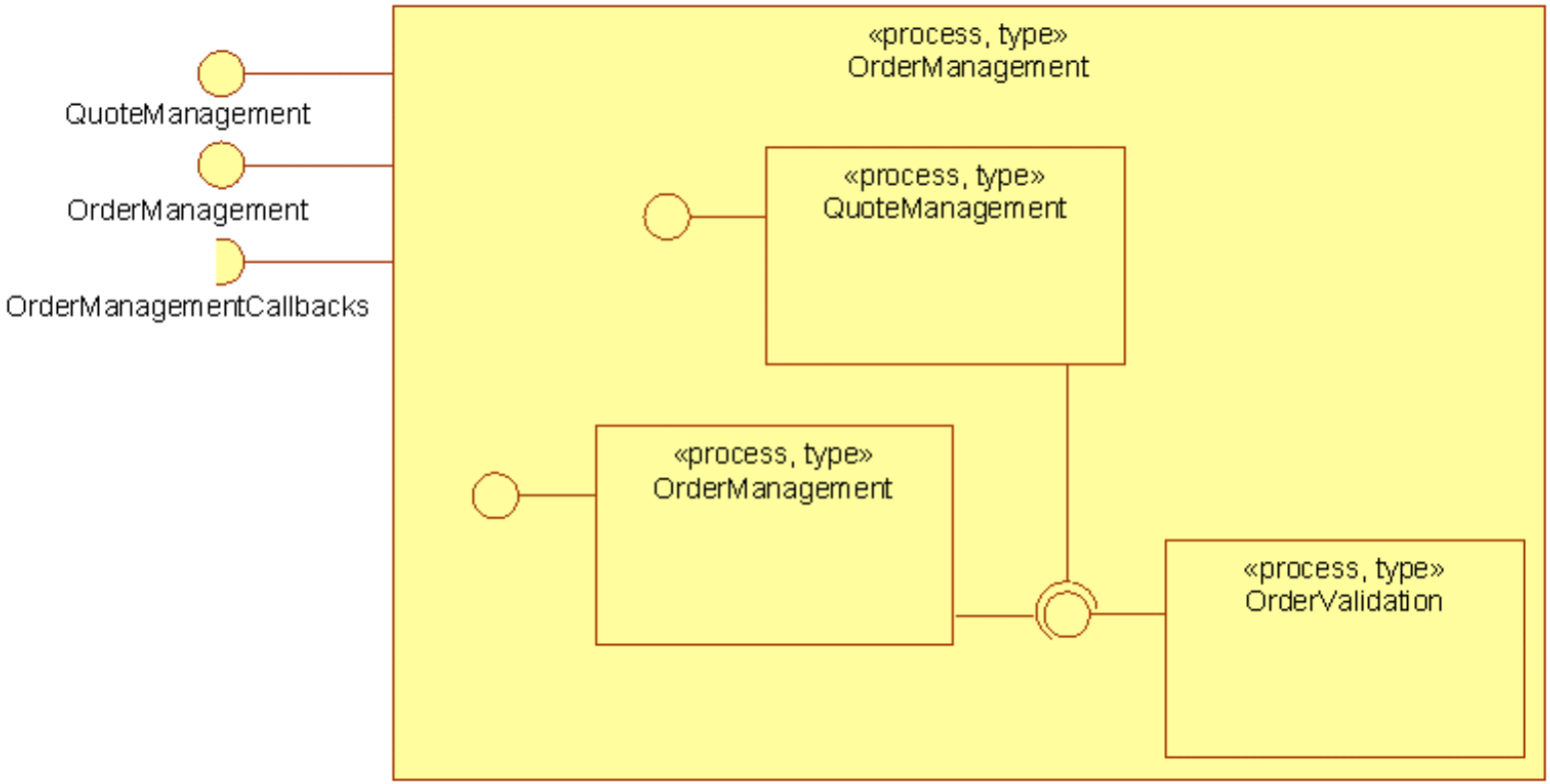
Key OMG initiative on SOA

- BPDM notation for a process (two views)



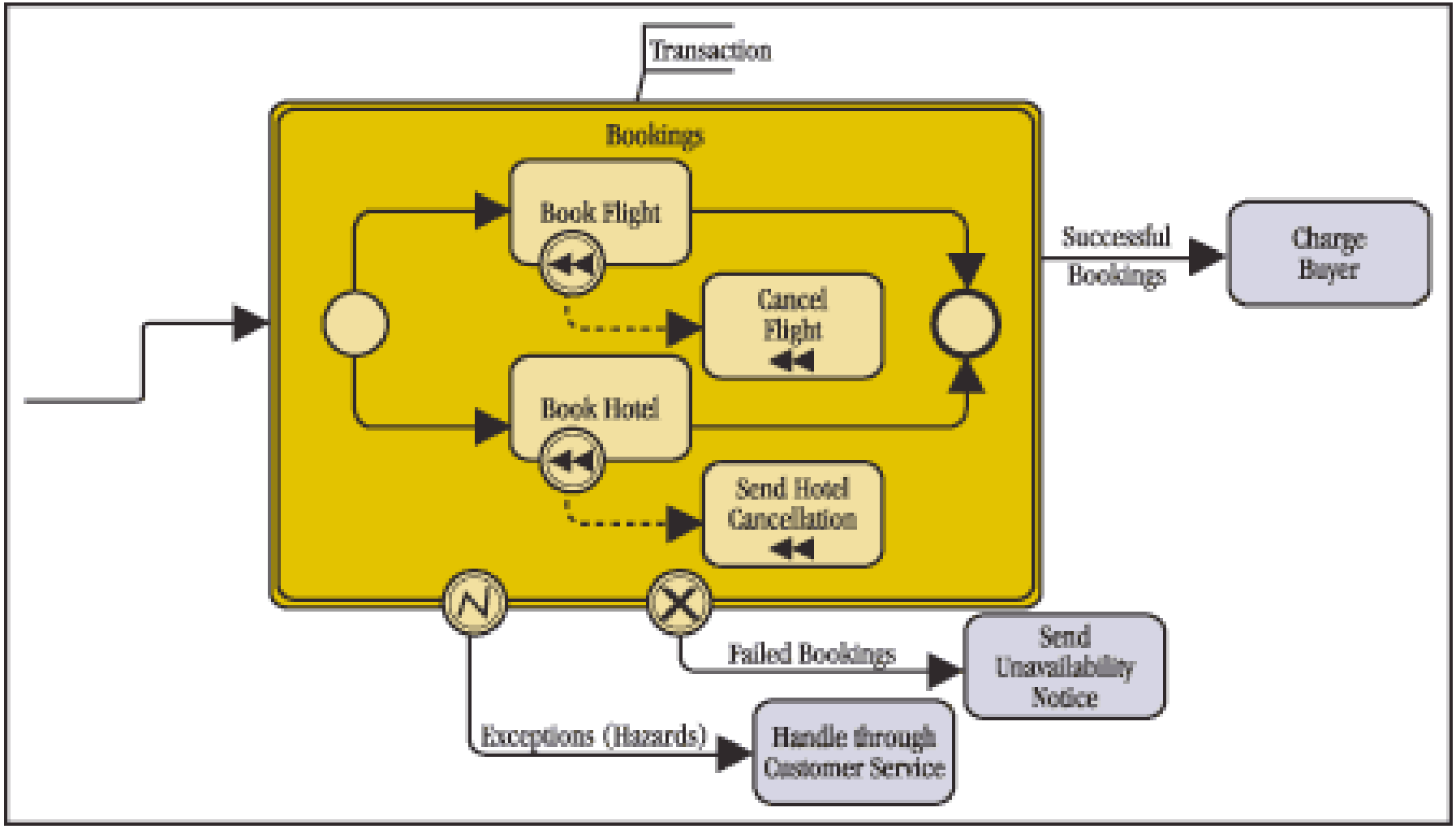
Key OMG initiative on SOA

- BPDM notation for a sub-process



Key OMG initiative on SOA

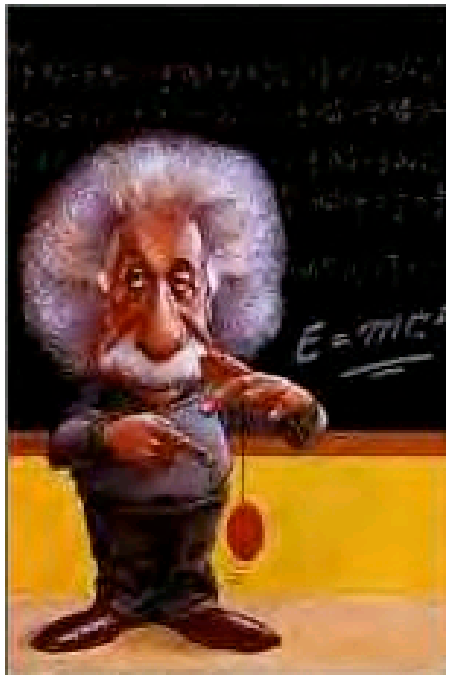
- BPMN notation for a sub-process

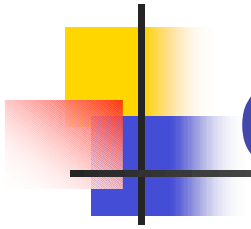


Conclusion

"The significant problems we face cannot be solved at the same level of thinking we were at when we created them"

Albert Einstein





Conclusion

- Start leveraging your existing applications into a Service Oriented Architecture using a higher level of abstraction by:
  - Using a suitable business modeling platform for your environment (with or without BPMN) :
    - UML 2.0 with BPDM support (using a UML profile)
    - BPDM native support
    - UML 2.0 with BPEL4WS support (using a UML profile)
    - BPEL4WS support
  - Derive your service interfaces from your contracts which support your business processes.
  - Consolidate your application portfolio in order to map its business functionalities with your service interfaces.
  - Rely on MOF Query-View-Transformation specification for your Model-to-Model transformations.
  - Rely on MOF Model-to-Text specification for your Model-to-Code transformations.