

# *A Model Based Enterprise Architecture for Web Services and XML*



**E**nterprise  
**C**ollaboration  
**A**rchitecture

**EDOC**

*Applying Model Driven Architecture using the  
OMG Enterprise Collaboration Architecture and  
XML Web Services*

# Introductions



DataAccessTechnologies

*Where Business Meets Technology*

Cory Casanave

cory-c@enterprise-component.com

Primary author of “CCA” in EDOC

# The OMG-Enterprise Collaboration Architecture

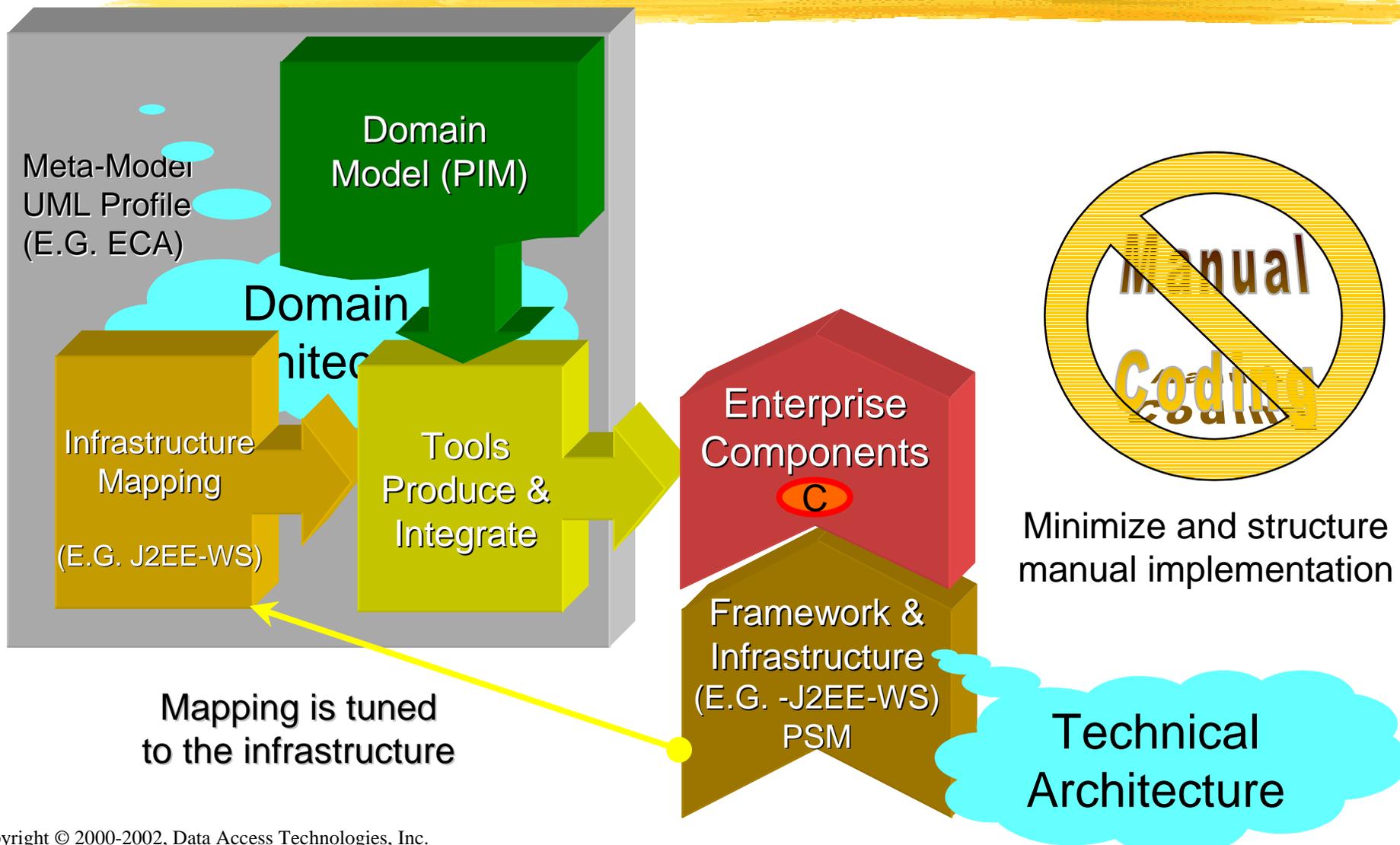
- ECA is a “profile of UML”, a way to use UML for a specific purpose - it is an OMG standard
  - That purpose is *modeling enterprise systems*.
- ECA is part of the “Model Driven Architecture” (MDA) initiative of the OMG
  - Using precise modeling techniques as part of the development lifecycle to speed development and provide technology independence
- ECA has been adopted by the OMG as part of the EDOC Profile for UML specification.
- RFP in process for Web Services Mapping

# Typical Problems

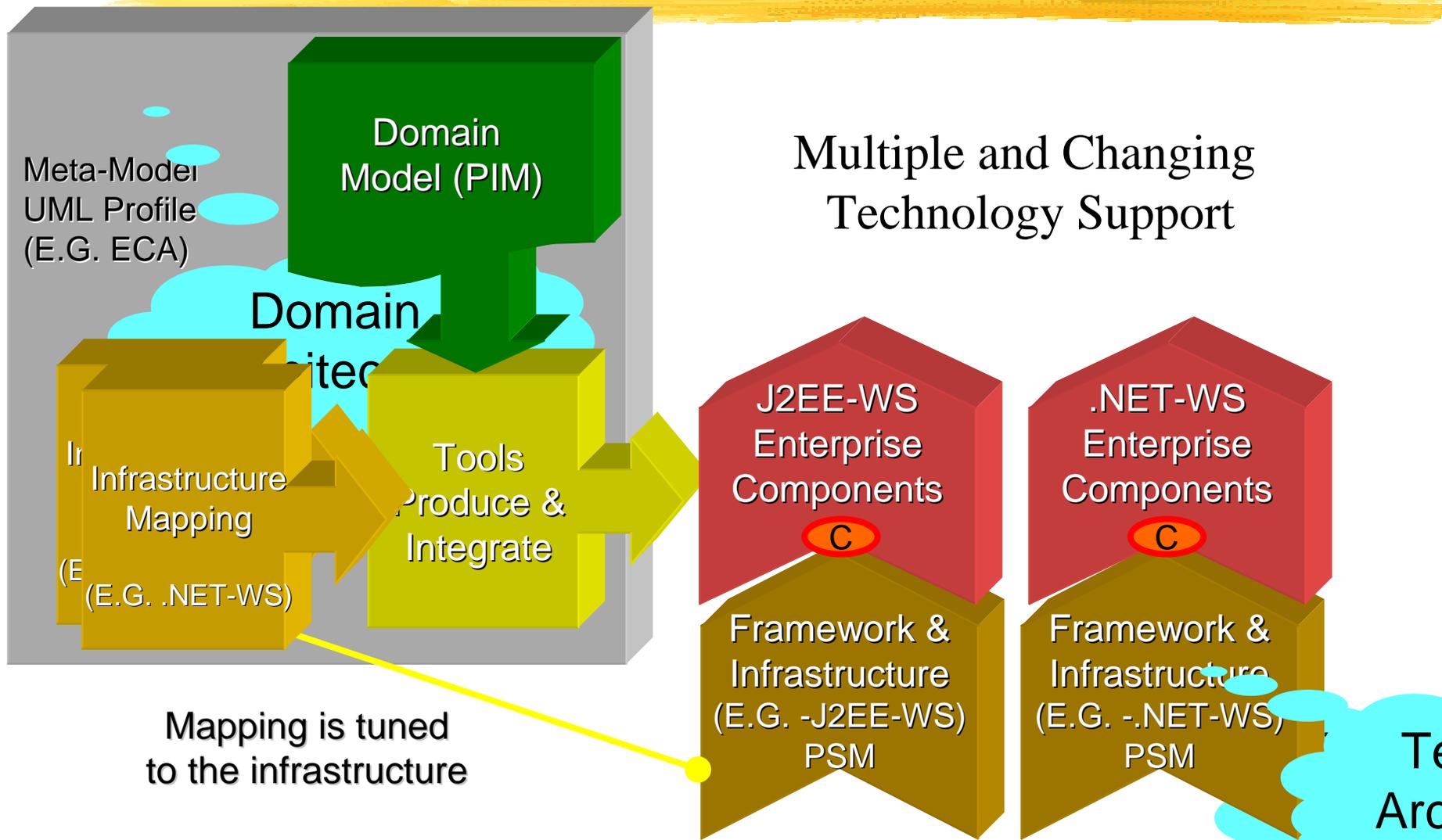


- Integration Nightmare
- Infrastructure, Version & Vendor lock-in
- Complex, divergent and manual development and deployment processes
- *Solutions typically require buy-in (Lock-in) to proprietary tools & infrastructure - your solution must be open*

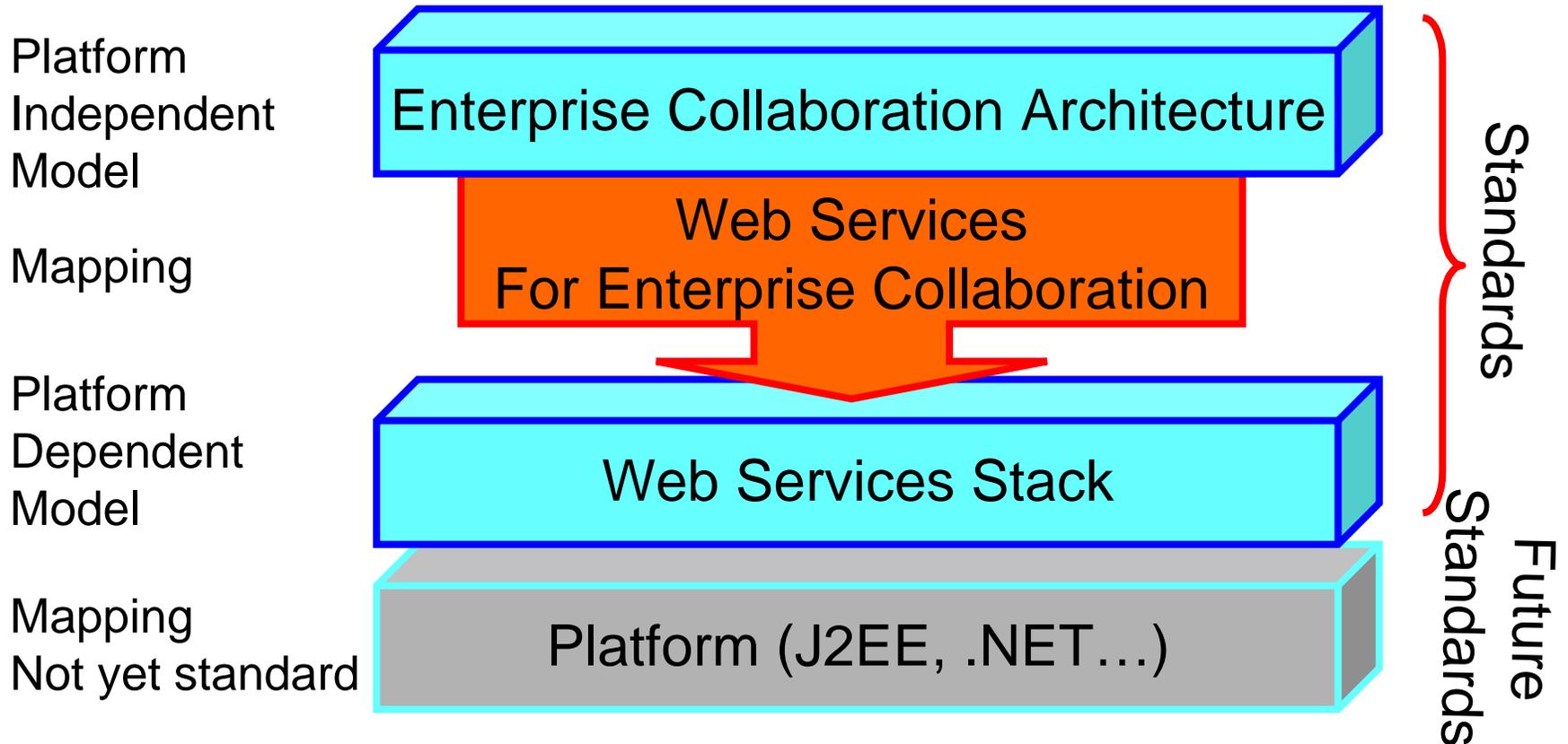
# Automated Model Driven Architecture



# Automated Model Driven Architecture



# MDA Solution for Web Services

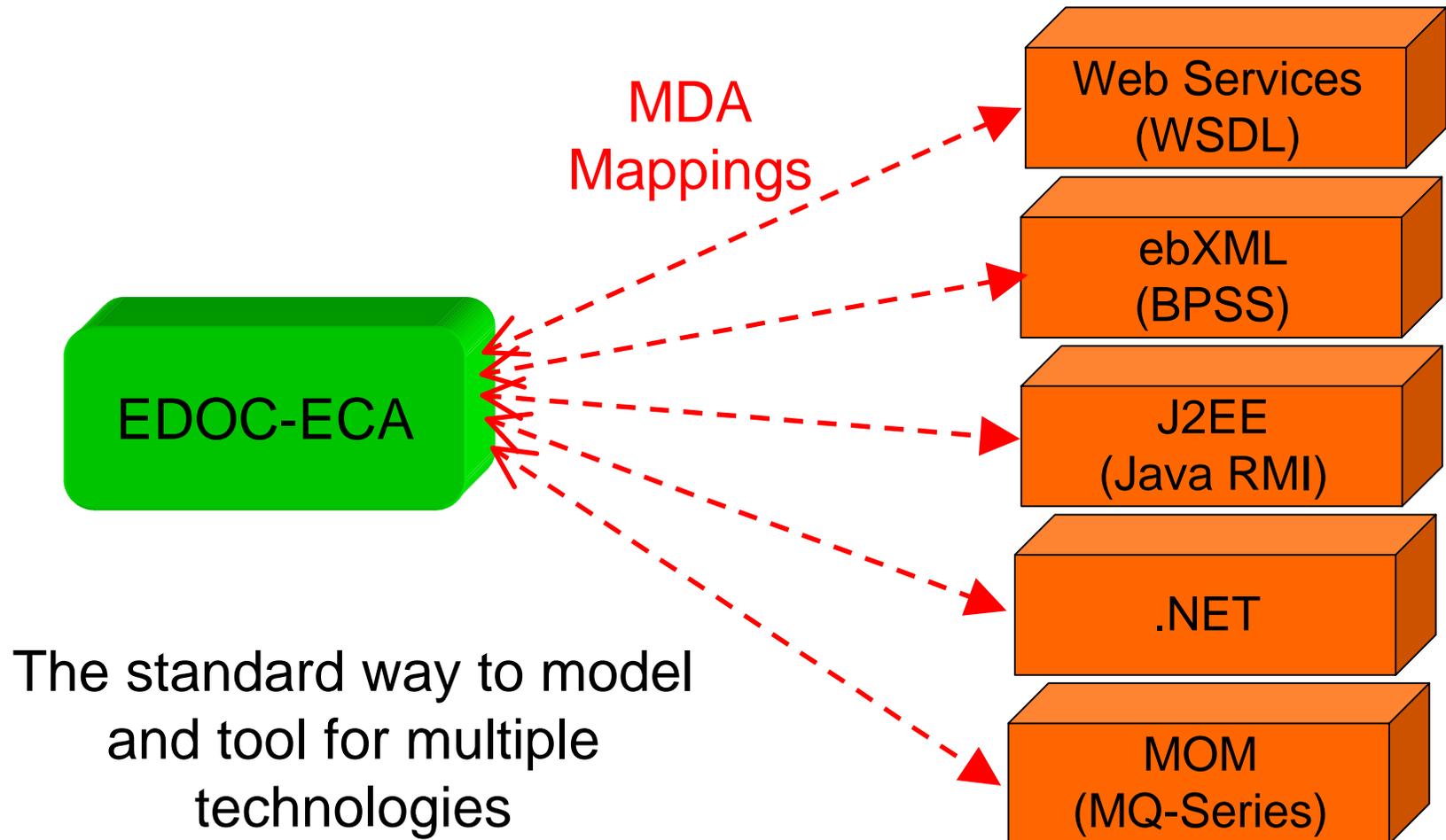


# XML Components



ECA and Web services together provide  
**An XML component architecture**  
*Independent of protocol and platform*

# ECA as the normal form



# EDOC Component Collaboration Architecture



The model of  
collaborative  
work

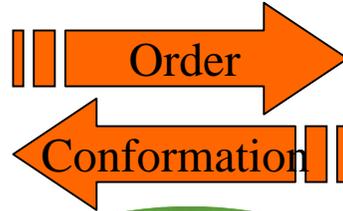
# The Marketplace Example



Mechanics Are Us  
Buyer



Physical  
Delivery

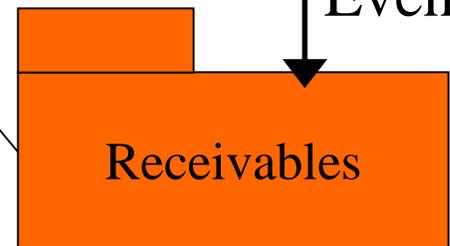
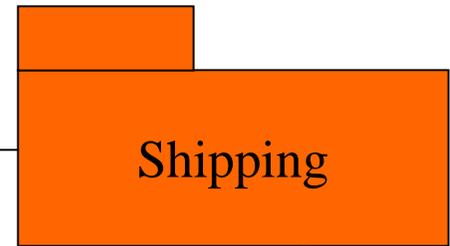
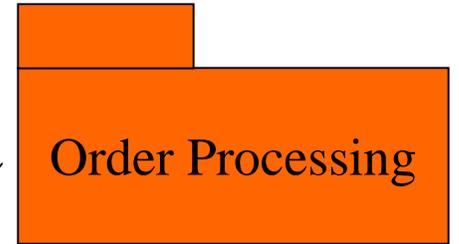
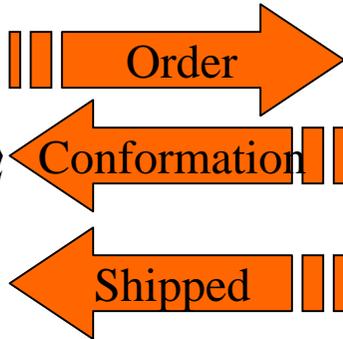


Acme Industries  
Seller

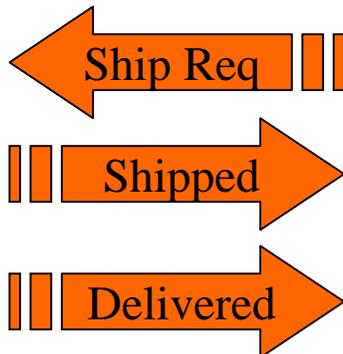


GetItThere Freight  
Shipper

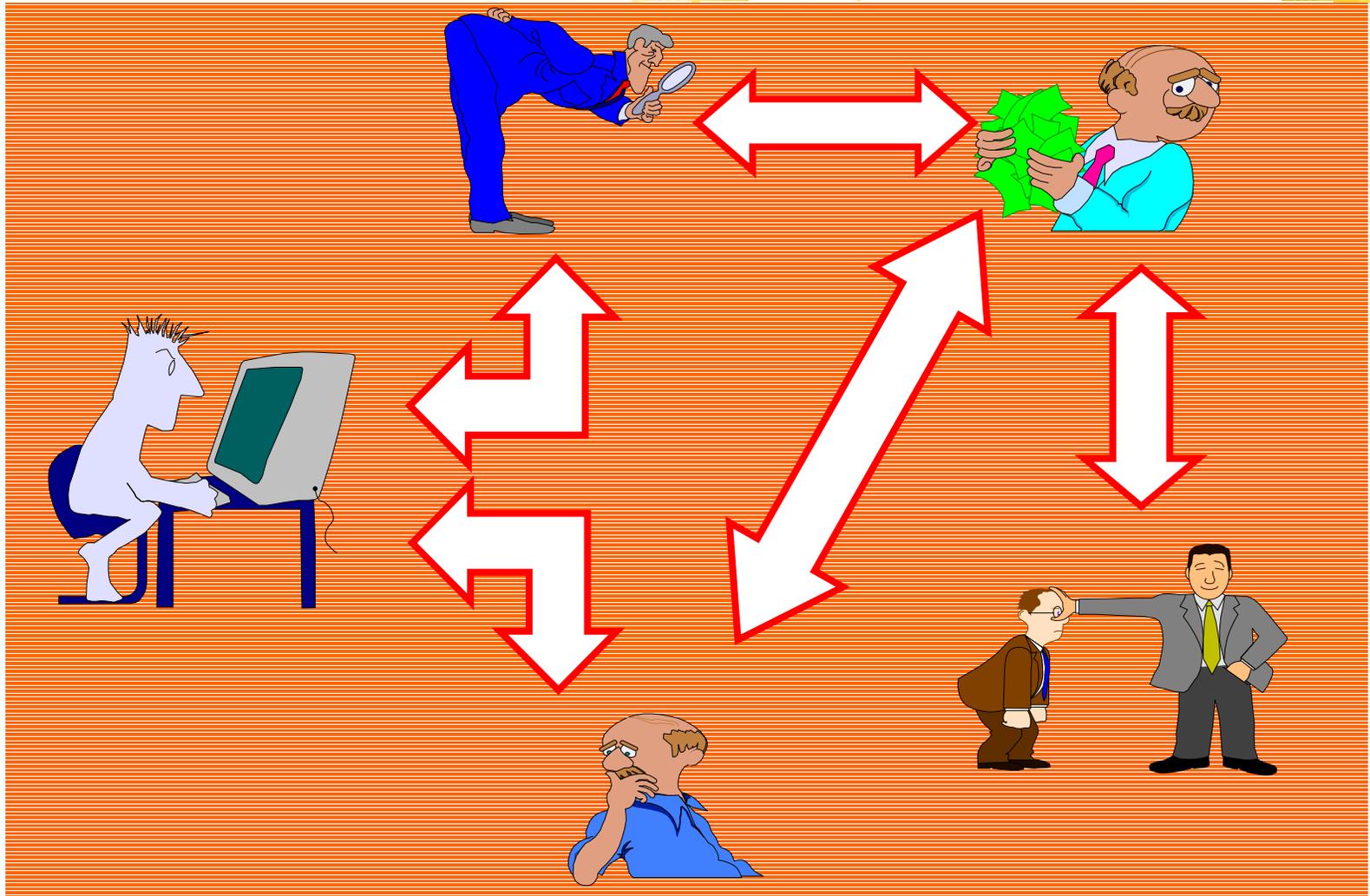
# The Seller's Detail



Event

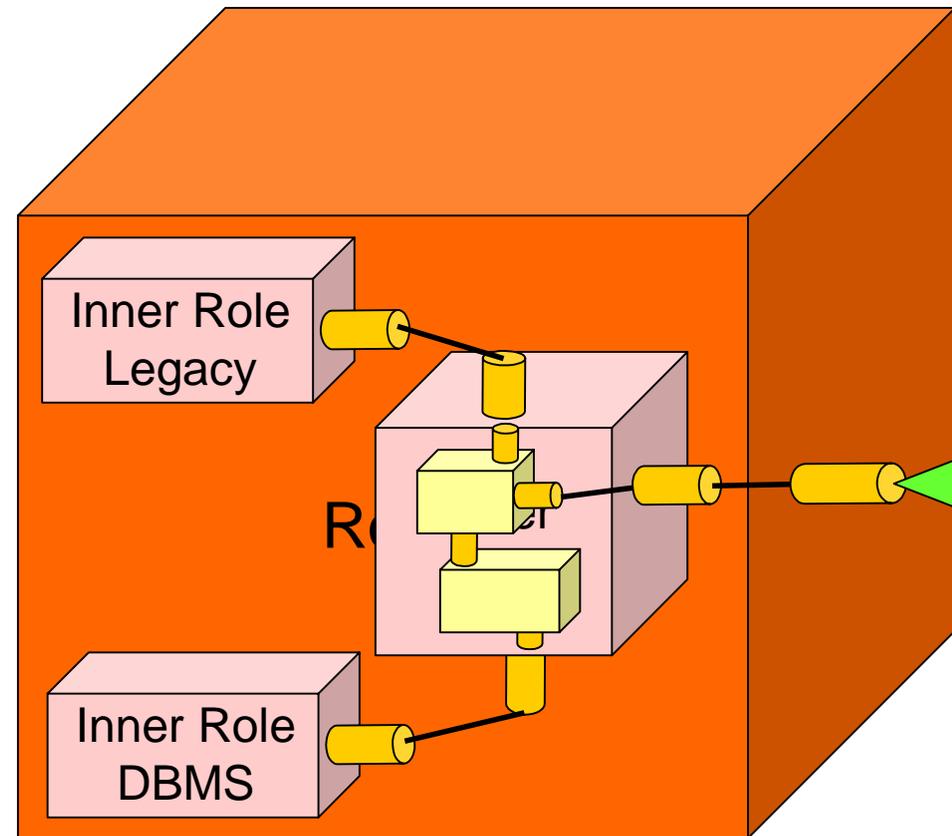


# Multiple roles in a collaboration

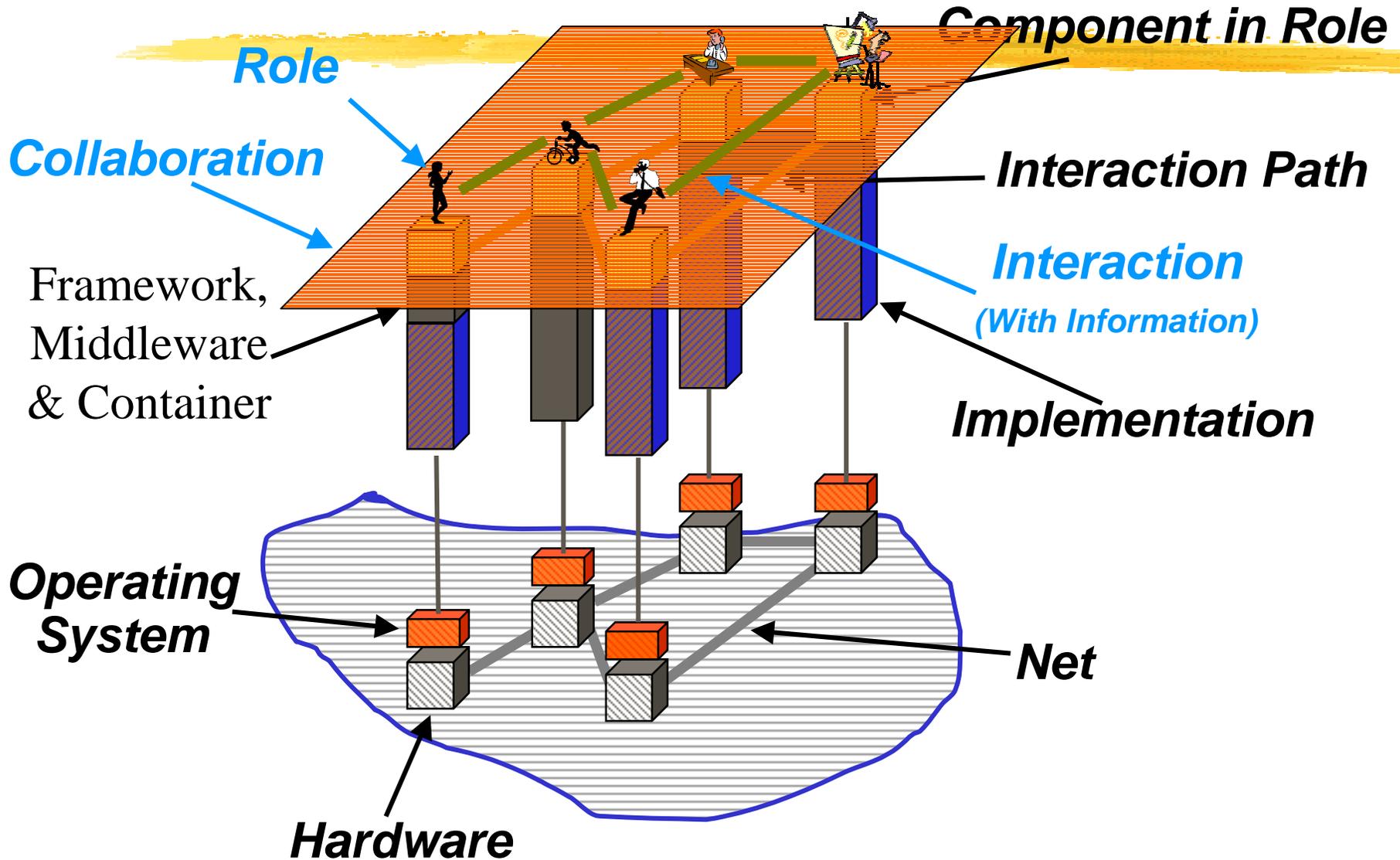


# Drilling down – inside a role

- The open domain should make no assumptions about the “inside” of a role.
- Inside one role you *frequently* find more collaborating “parts” of the enterprise - the same model *may* be used
- Until you get to system inside a managed domain
  - Shared resources (DBMS)
  - Common Management
  - Frequently a legacy system



# Roles to Systems



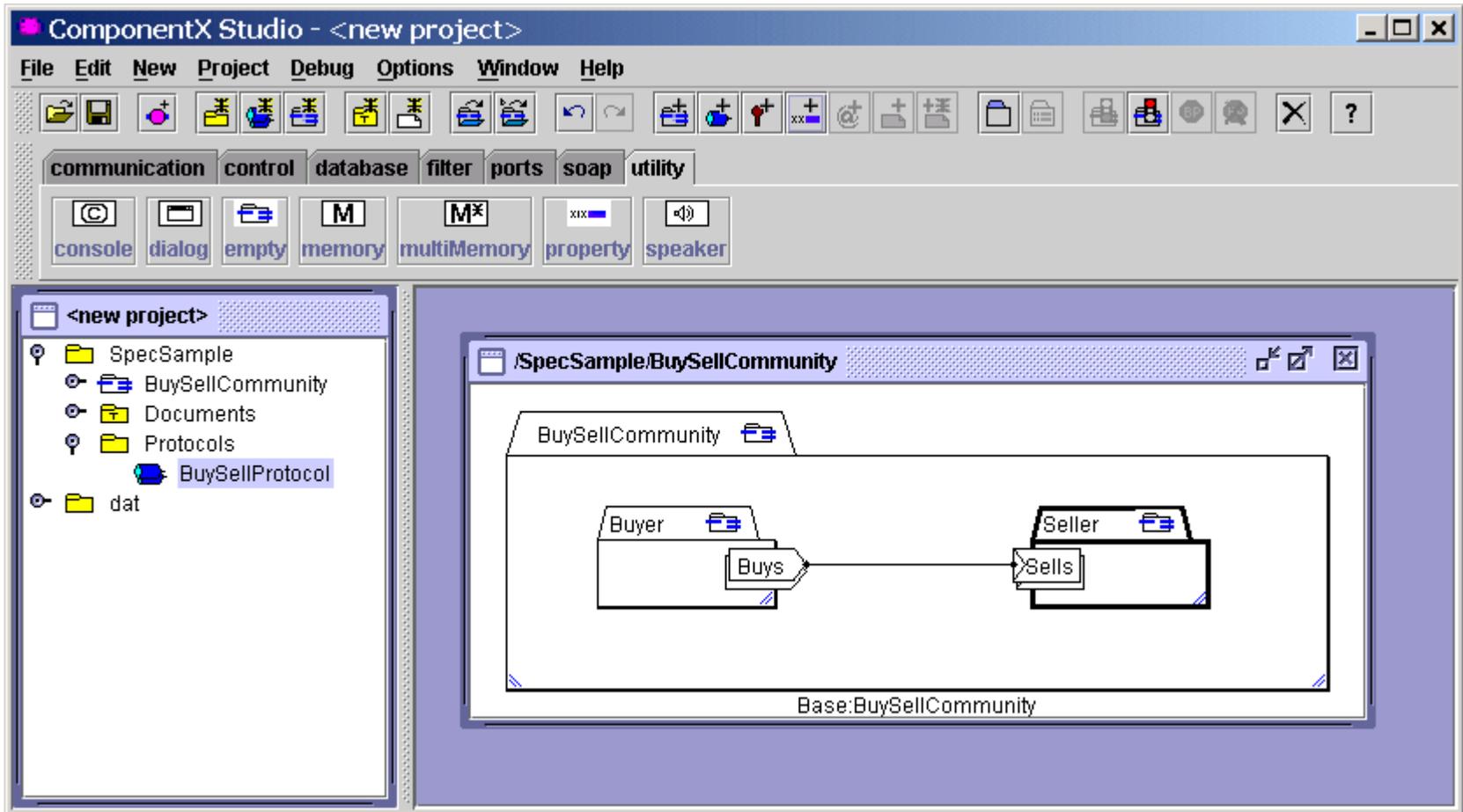
# Parts of a CCA Specification



- Structure of process components and protocols
  - Process components, ports, protocols and documents
    - Class Diagram or CCA Notation
- Composition of process components
  - How components are used to specify components
    - Collaboration diagram or CCA Notation
- Choreography
  - Ordering of flows and protocols in and between process components
    - Activity Diagram



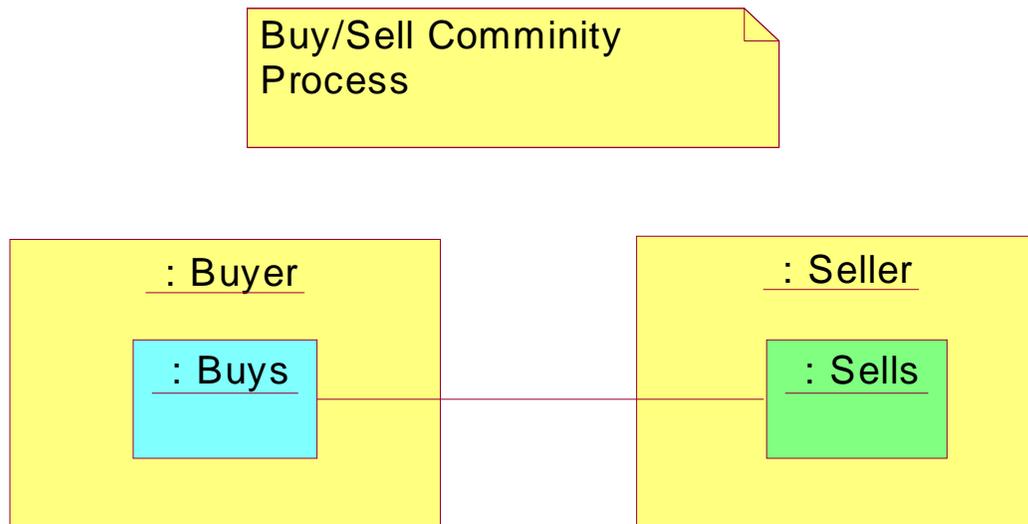
# Community Process (CCA)



## CCA Notation

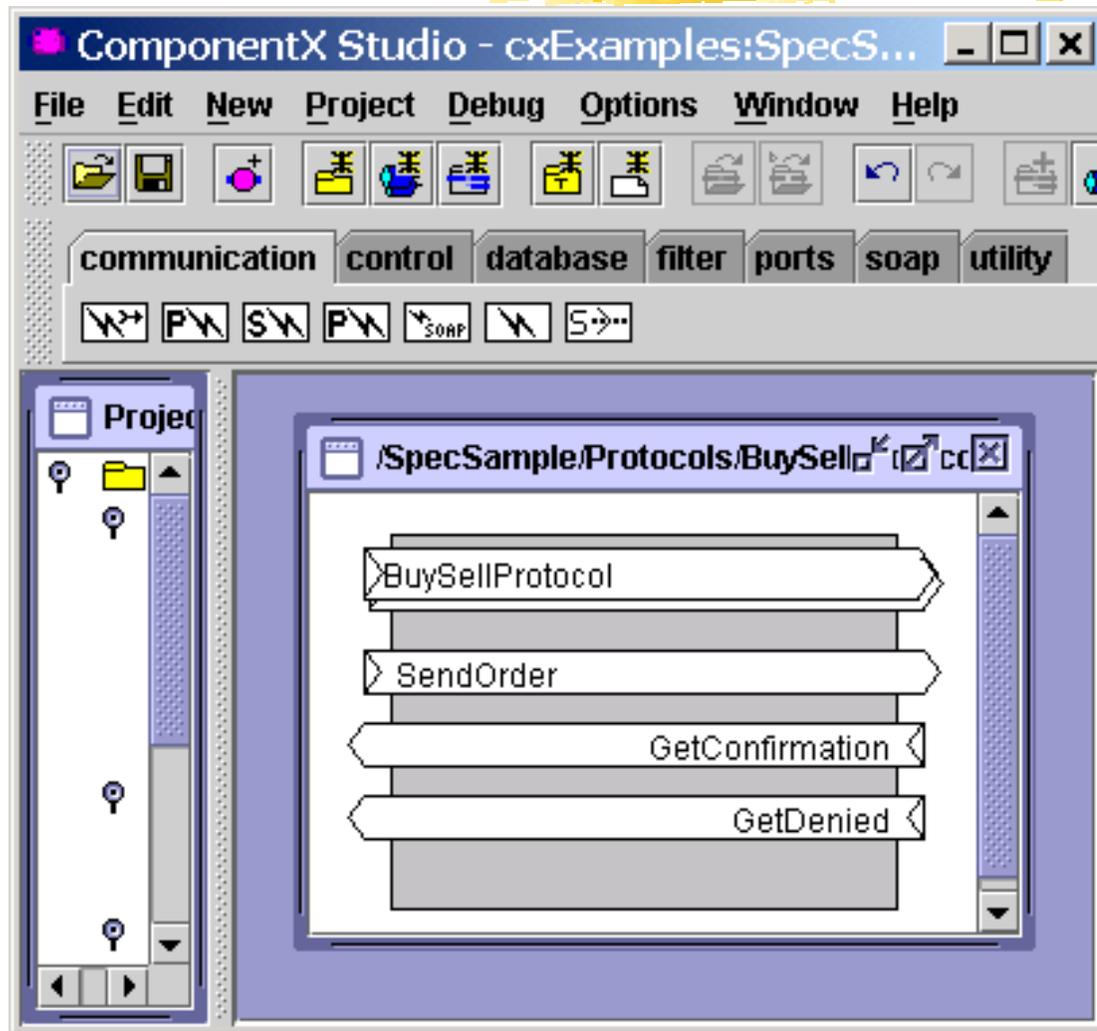
# Community Process

*Generic UML*



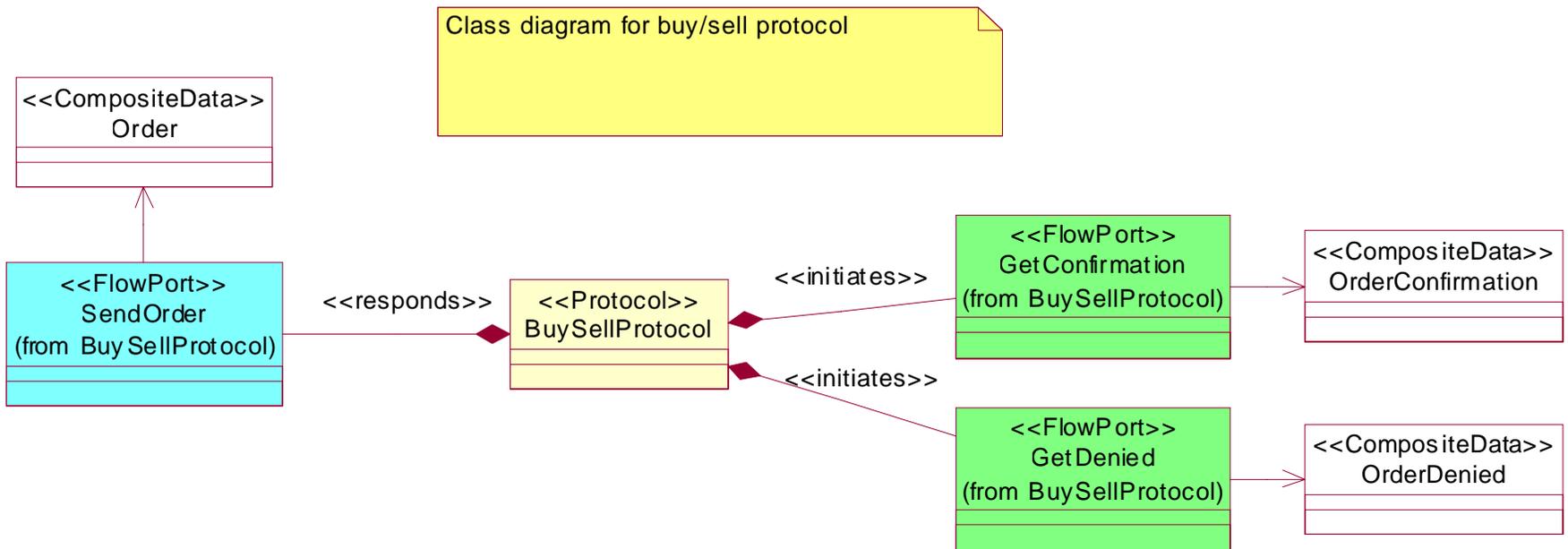
UML Collaboration Diagram

# Protocol (CCA)



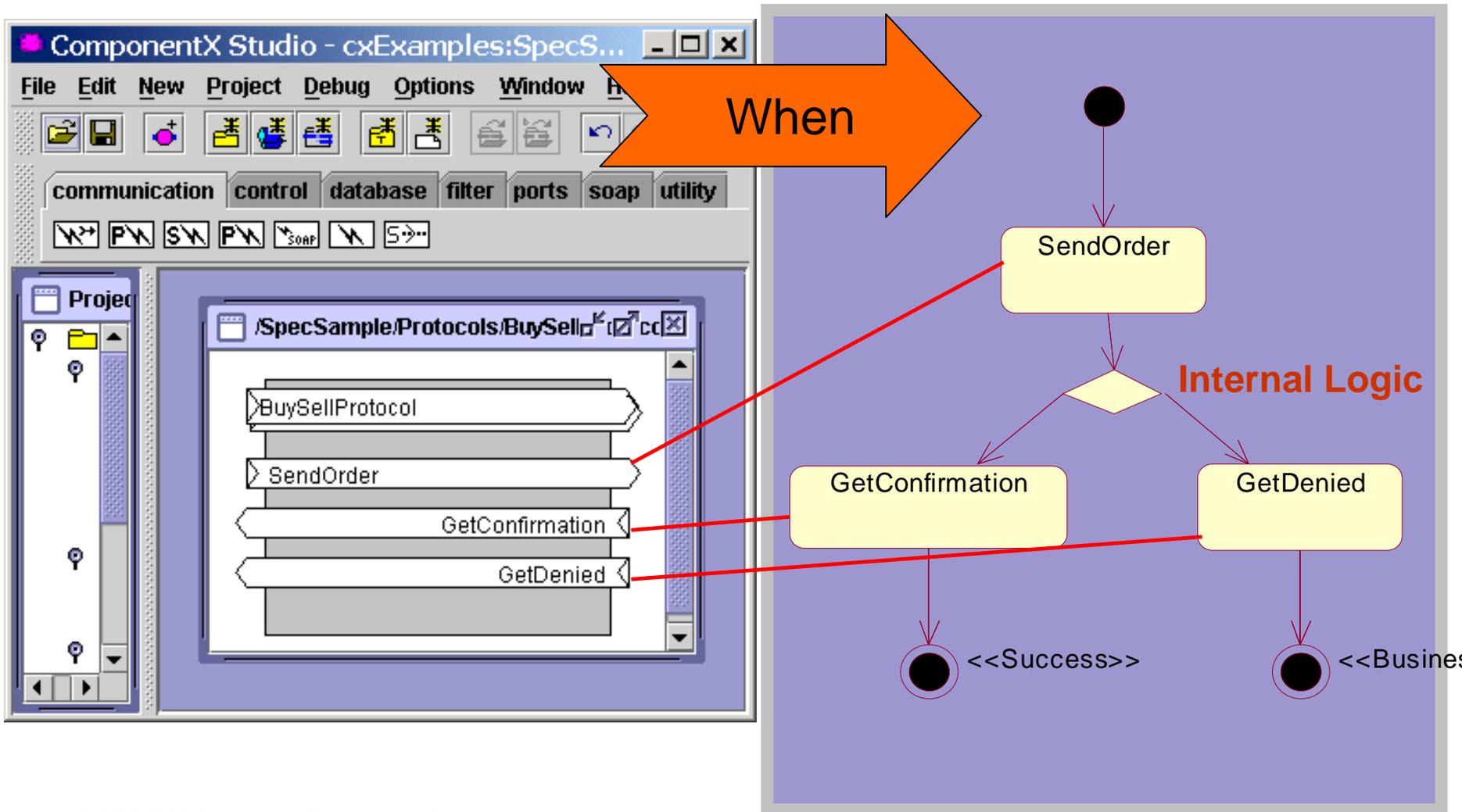
# Protocol

## Generic UML

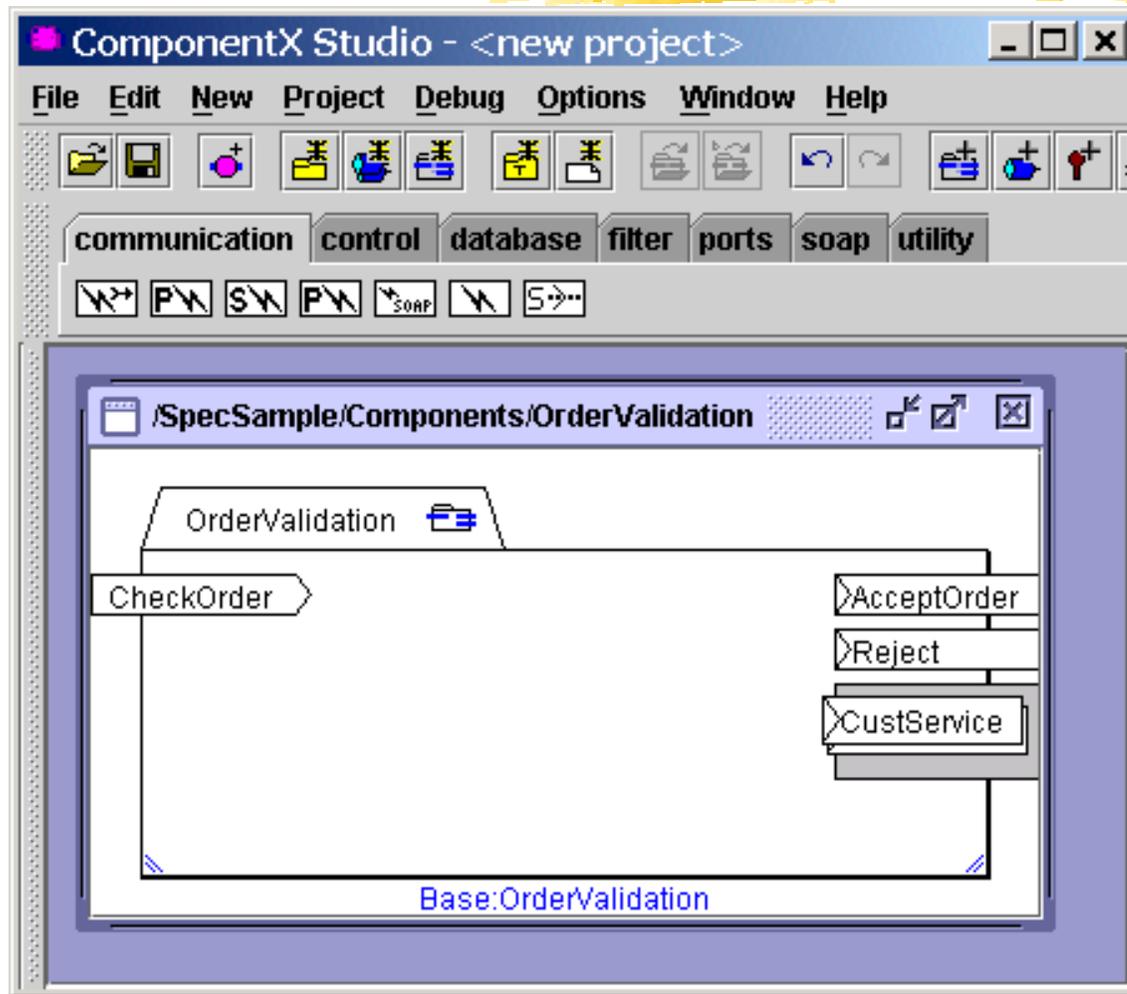


## UML Class Diagram

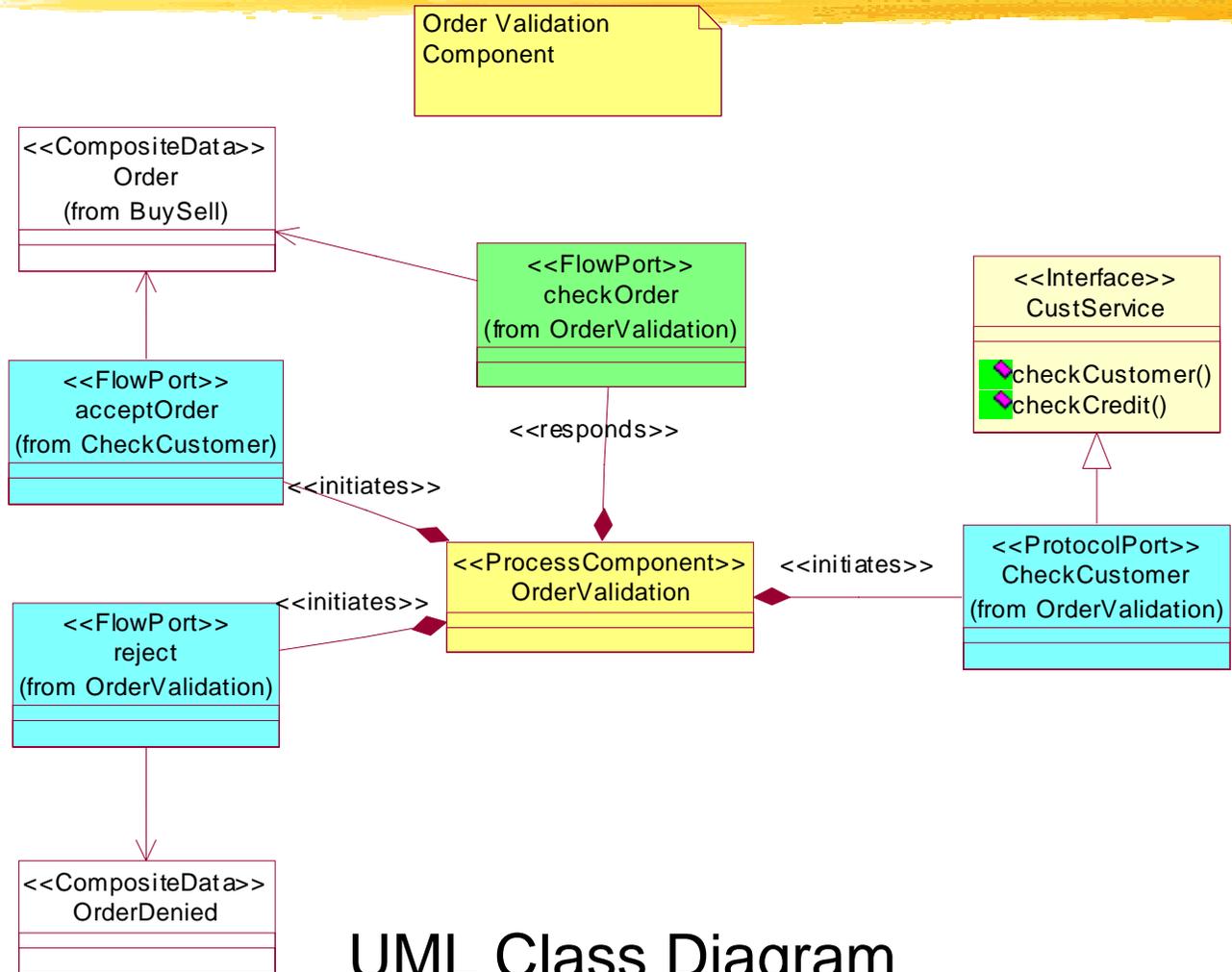
# Protocol Choreography



# Validation Component (CCA)



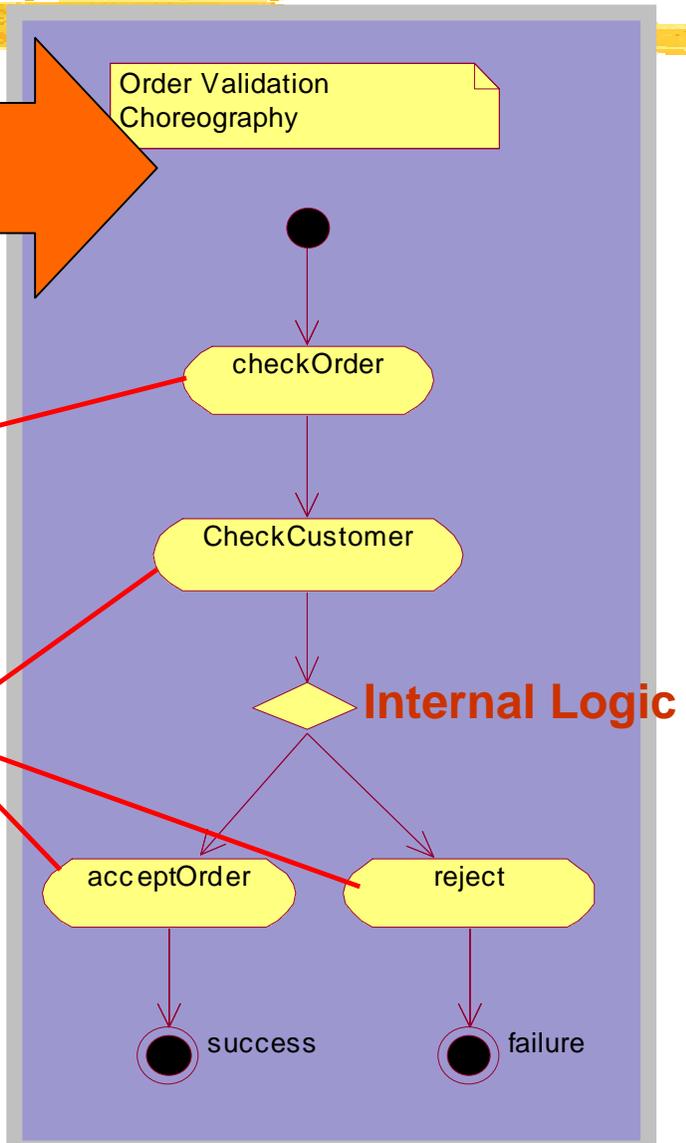
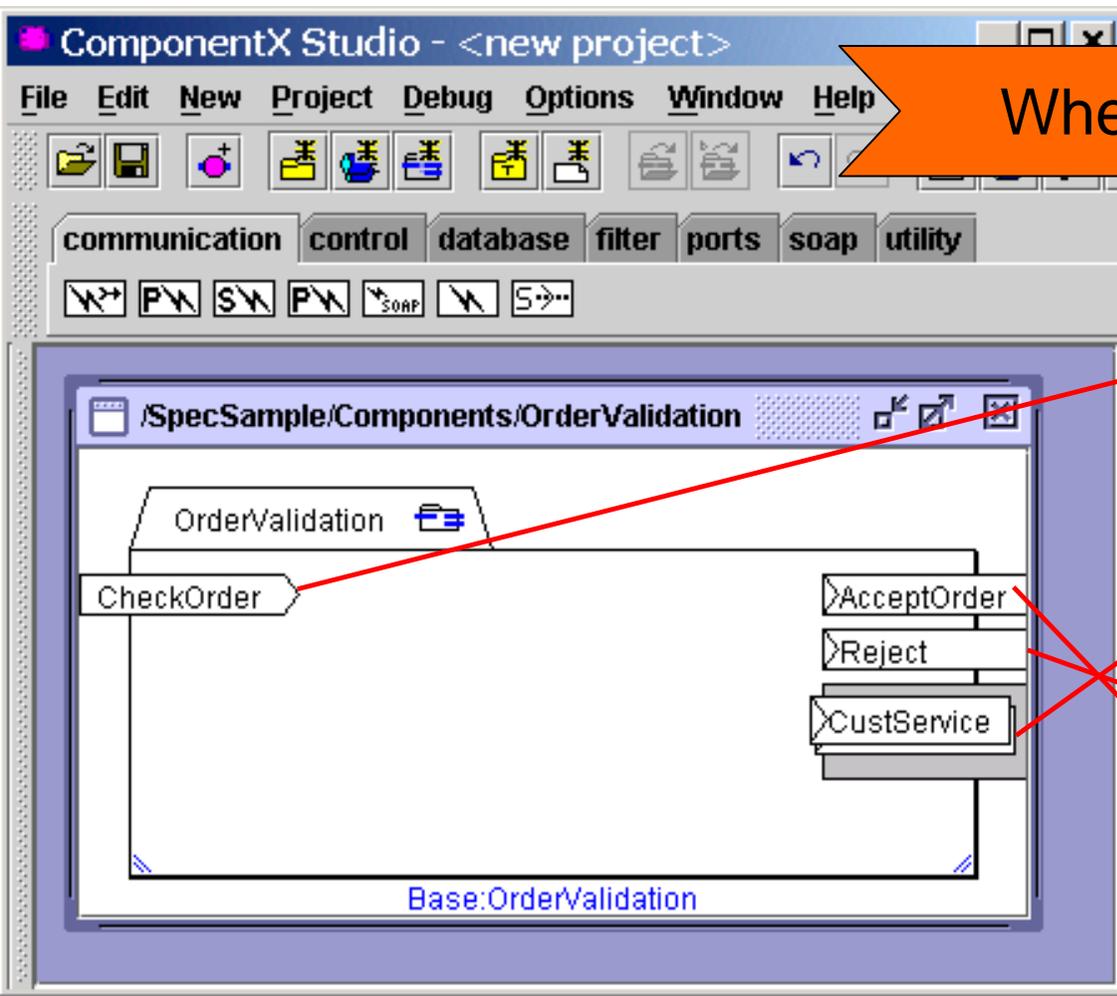
# Validation Component



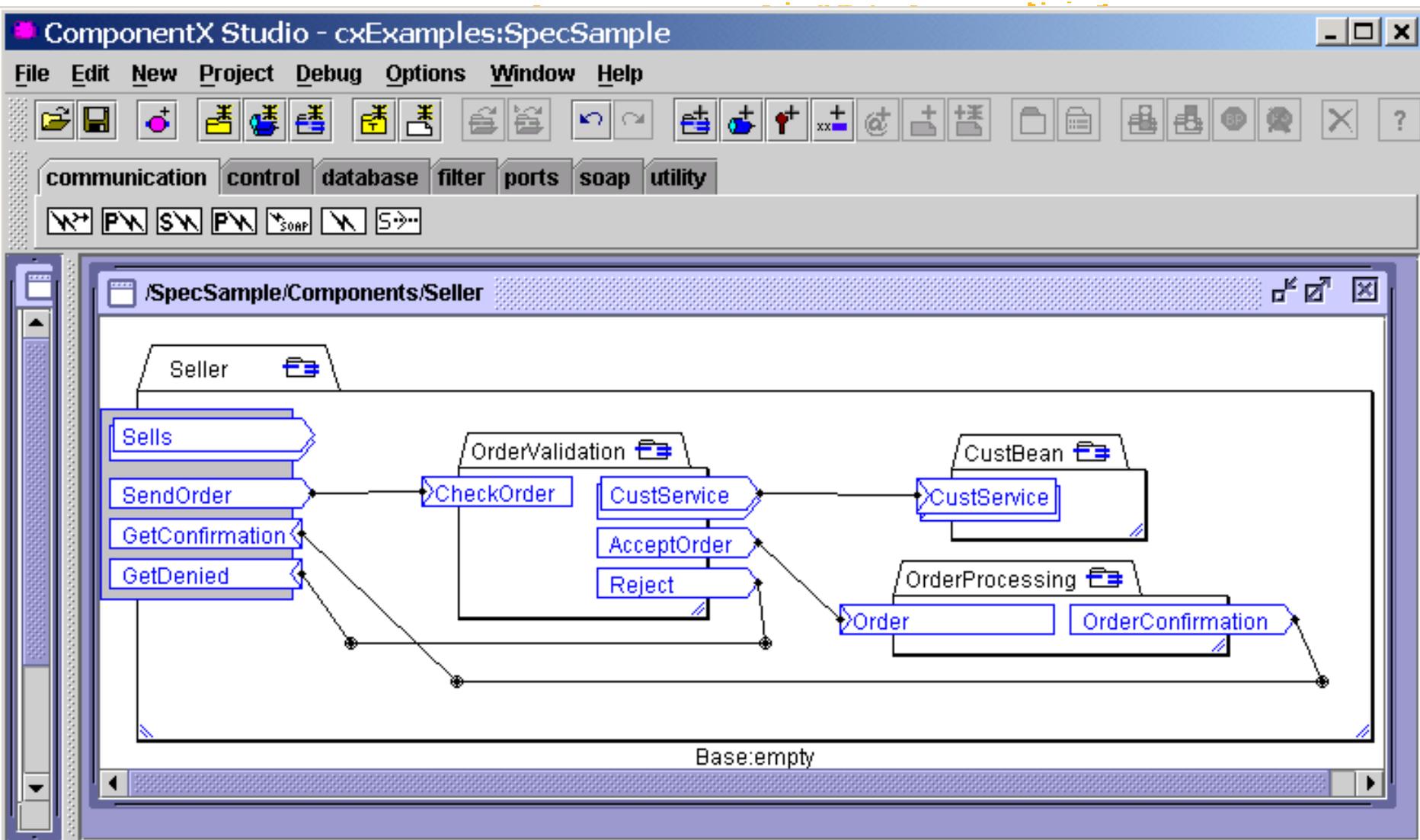
*Generic UML*

UML Class Diagram

# Choreography



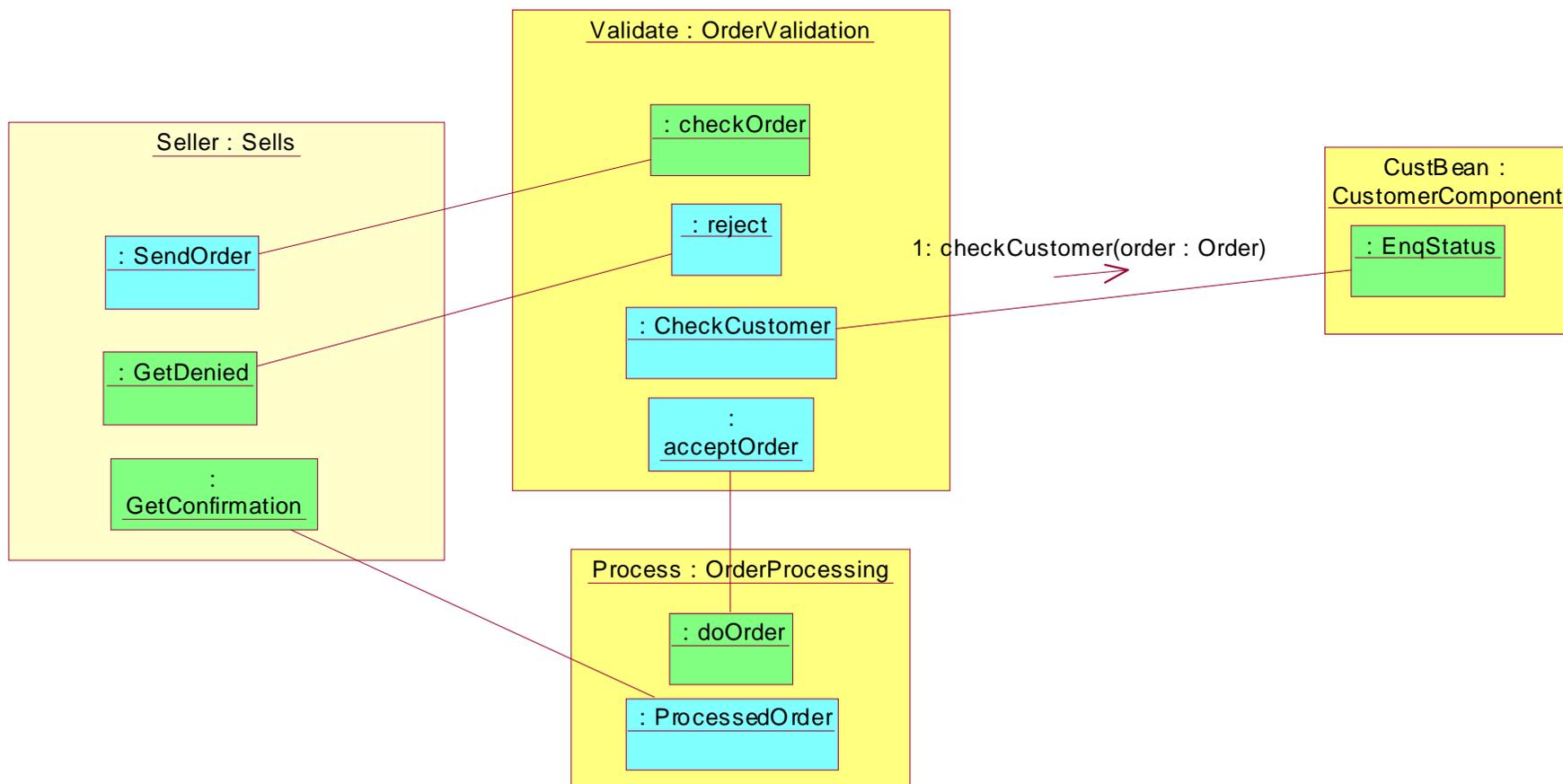
# Composition (CCA)



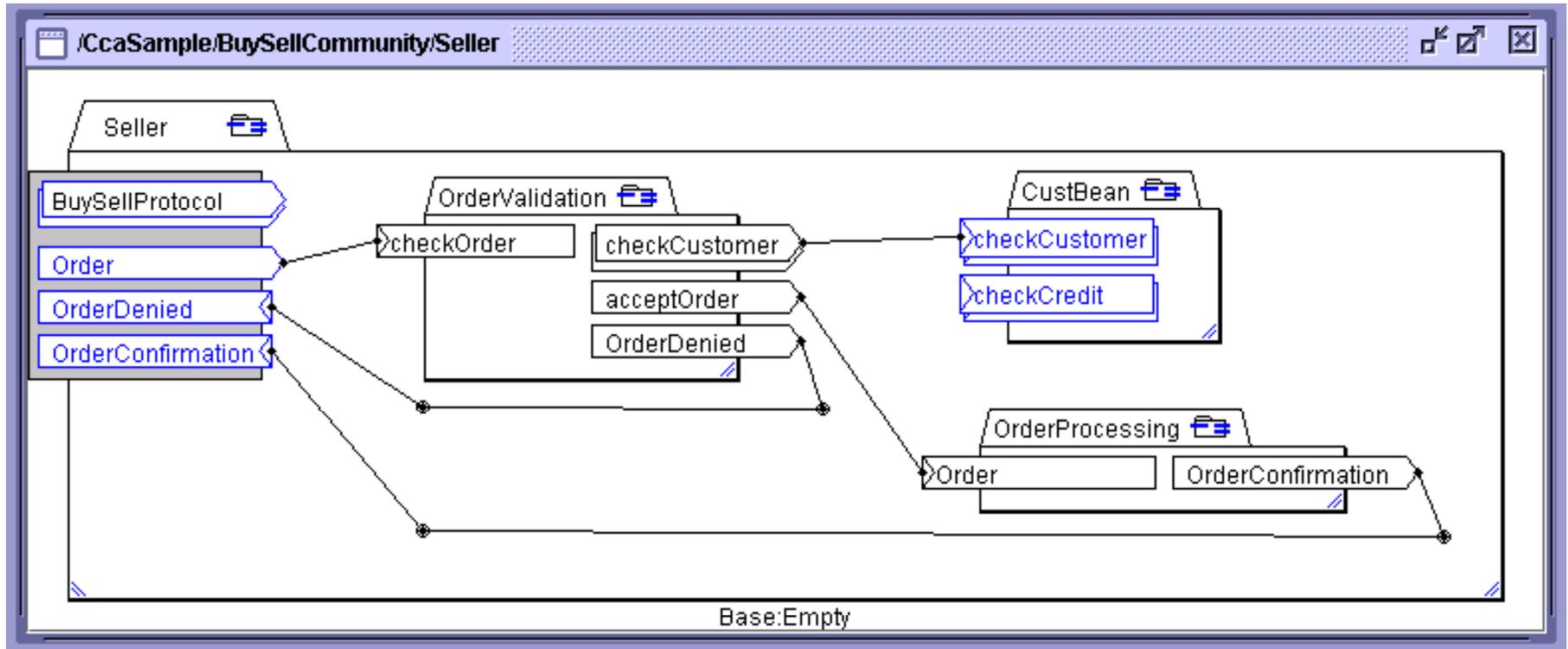
# Composition (UML Collaboration)

Seller Composition

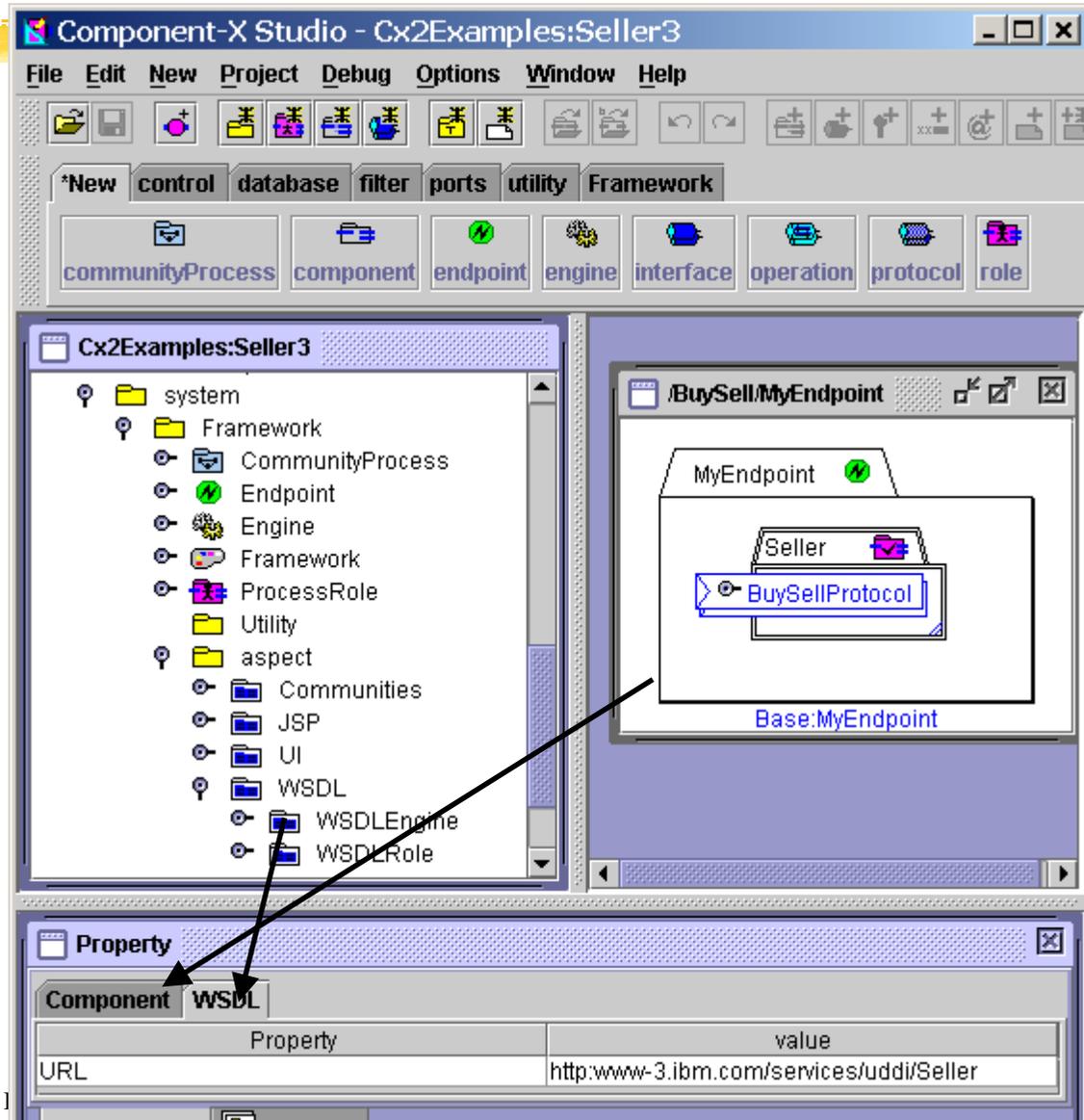
*Generic UML*



# Composition



# Aspects



*Generic UML*

Tagged Values

Property	
Component	WSDL
Property	value
URL	<a href="http://www-3.ibm.com/services/uddi/Seller">http://www-3.ibm.com/services/uddi/Seller</a>

# WSEC

A horizontal brushstroke in a vibrant yellow color, with a textured, painterly appearance, extending across the width of the slide below the 'WSEC' text.

Web Services for Enterprise  
Collaboration

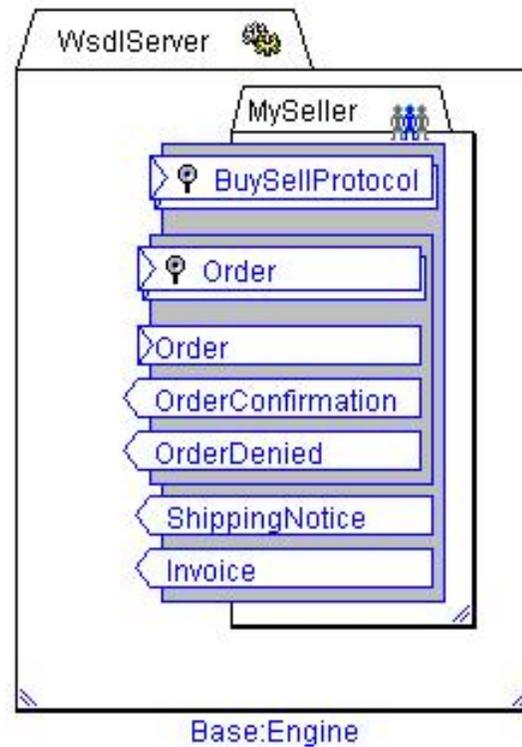
Initial Proposal

# Distributed Components

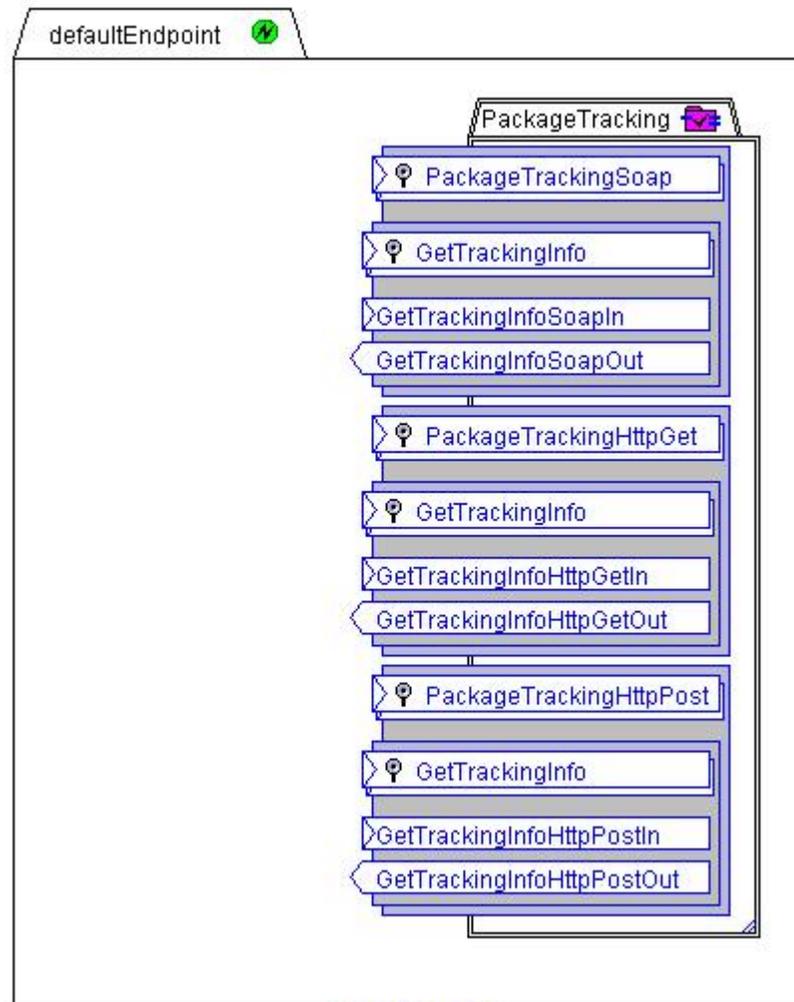


- Define “role” as the abstract contract
- Define “Engine” as exposing a set of DCs
- Define “Endpoint” as consuming a set of DCs
- Define “Proxy” as the use of an external role

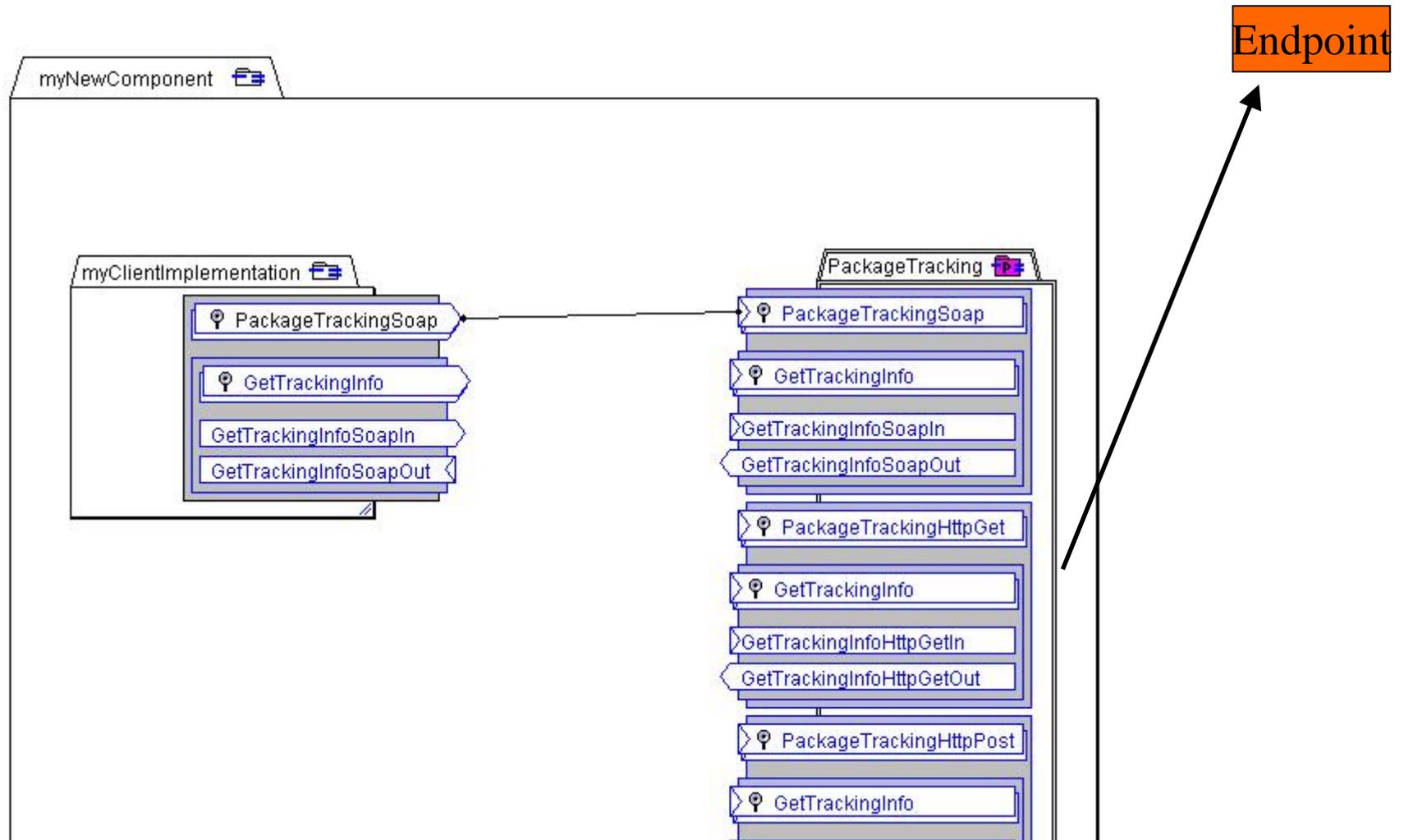
# Engine exposing a DC



# Defining an external component resource



# Using a proxy



# Mapping of a WSDL Engine

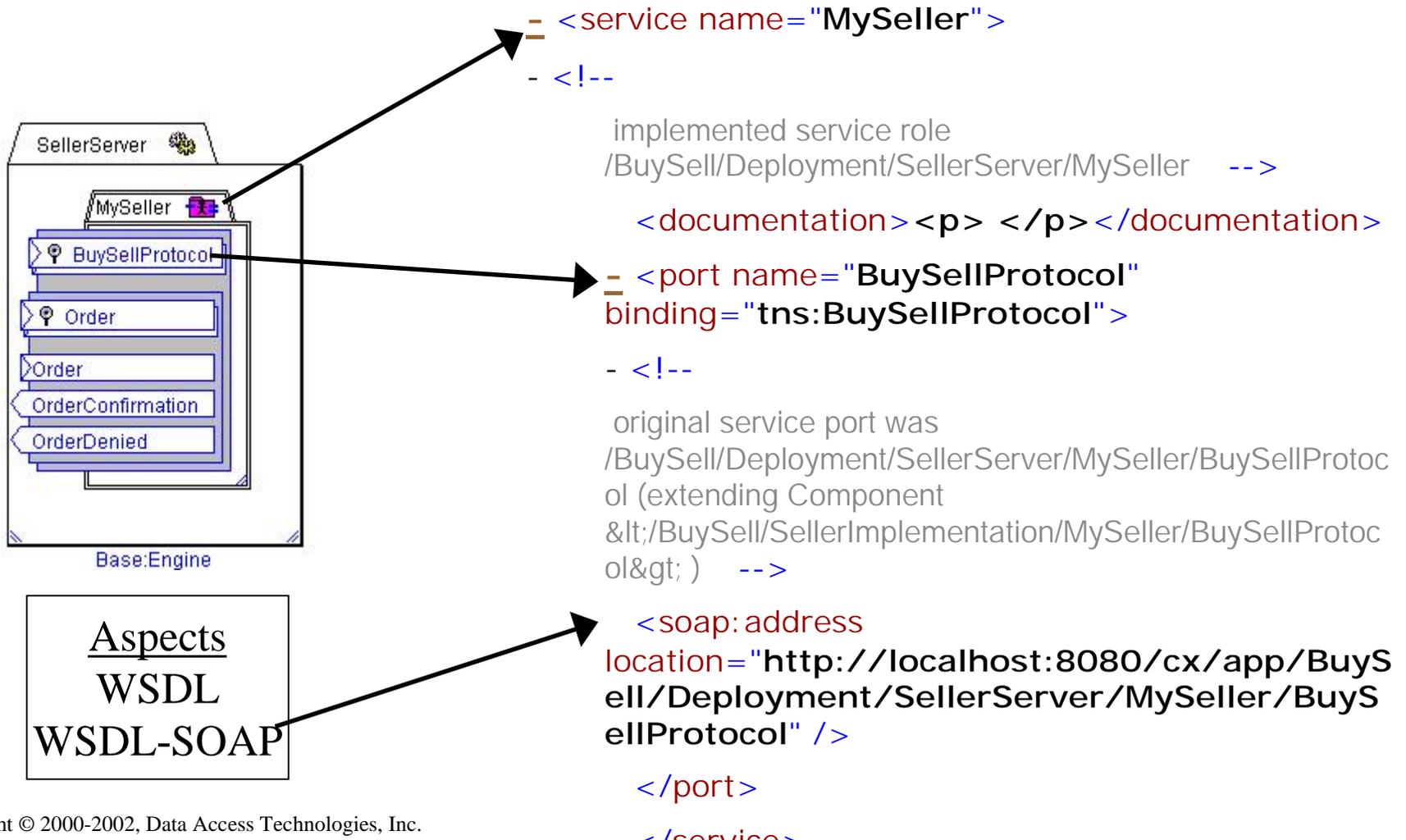


Aspects  
WSDL  
WSDL-SOAP

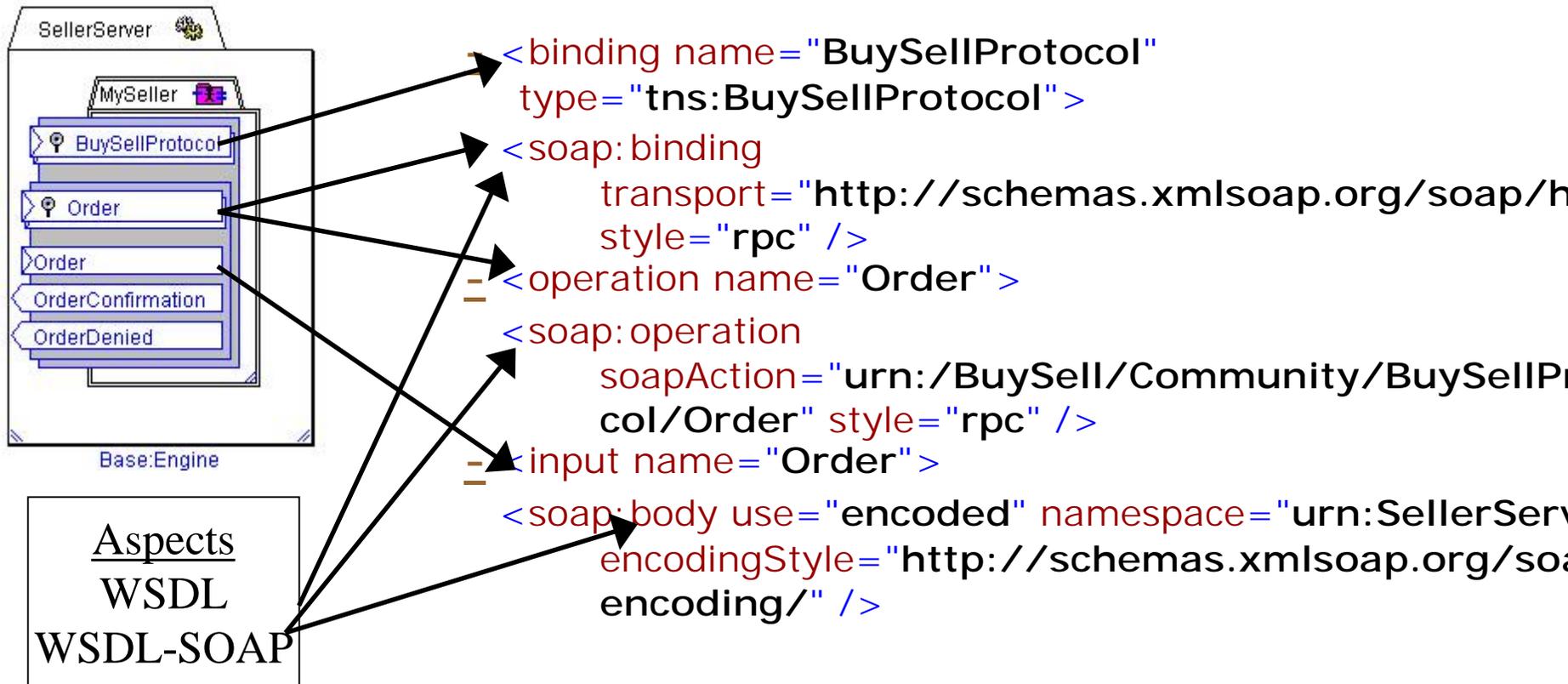
```
- <definitions xmlns="http://schemas.xmlsoap.org/wsdl"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http"
  ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xs2000="http://www.w3.org/1999/XMLSchema"
  xmlns:xs2001="http://www.w3.org/2001/XMLSchema"
  targetNamespace="urn:SellerServer" xmlns:tns="urn:SellerServer"
  xmlns:CoreTypes="urn:CoreTypes" xmlns:Ordering="urn:Ordering"
- <!--
```

definitions obtained from component /BuySell/Deployment/SellerServer

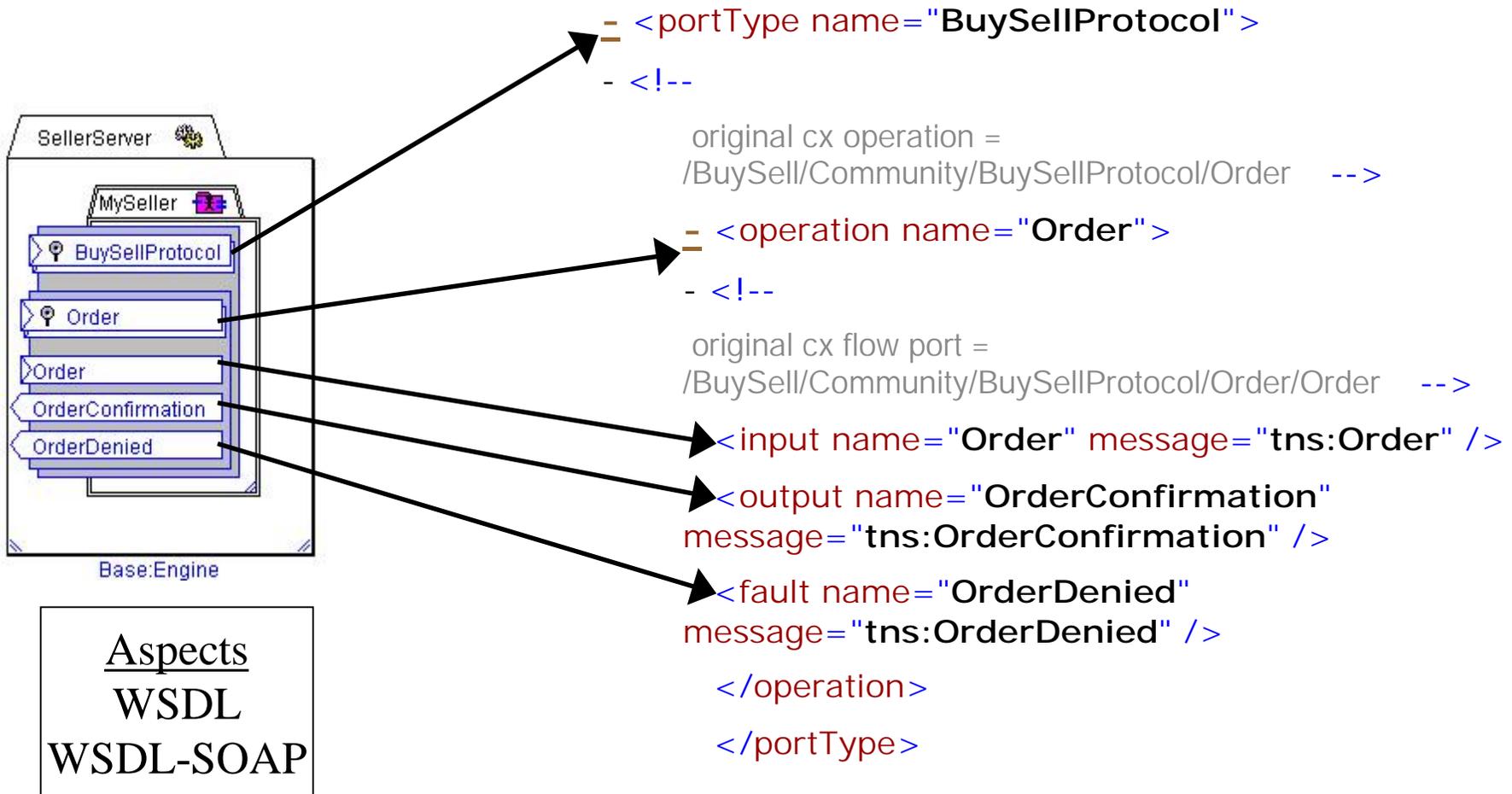
# Mapping of a DC



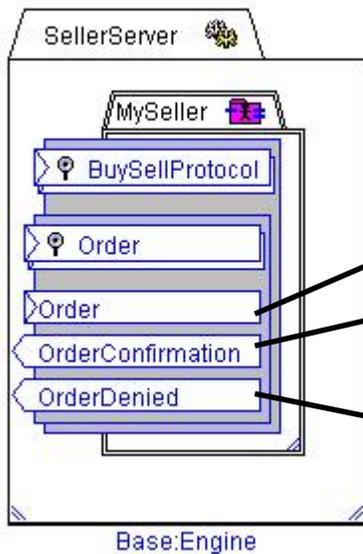
# Mapping of a protocol binding



# Mapping of a protocol



# Mapping of message types

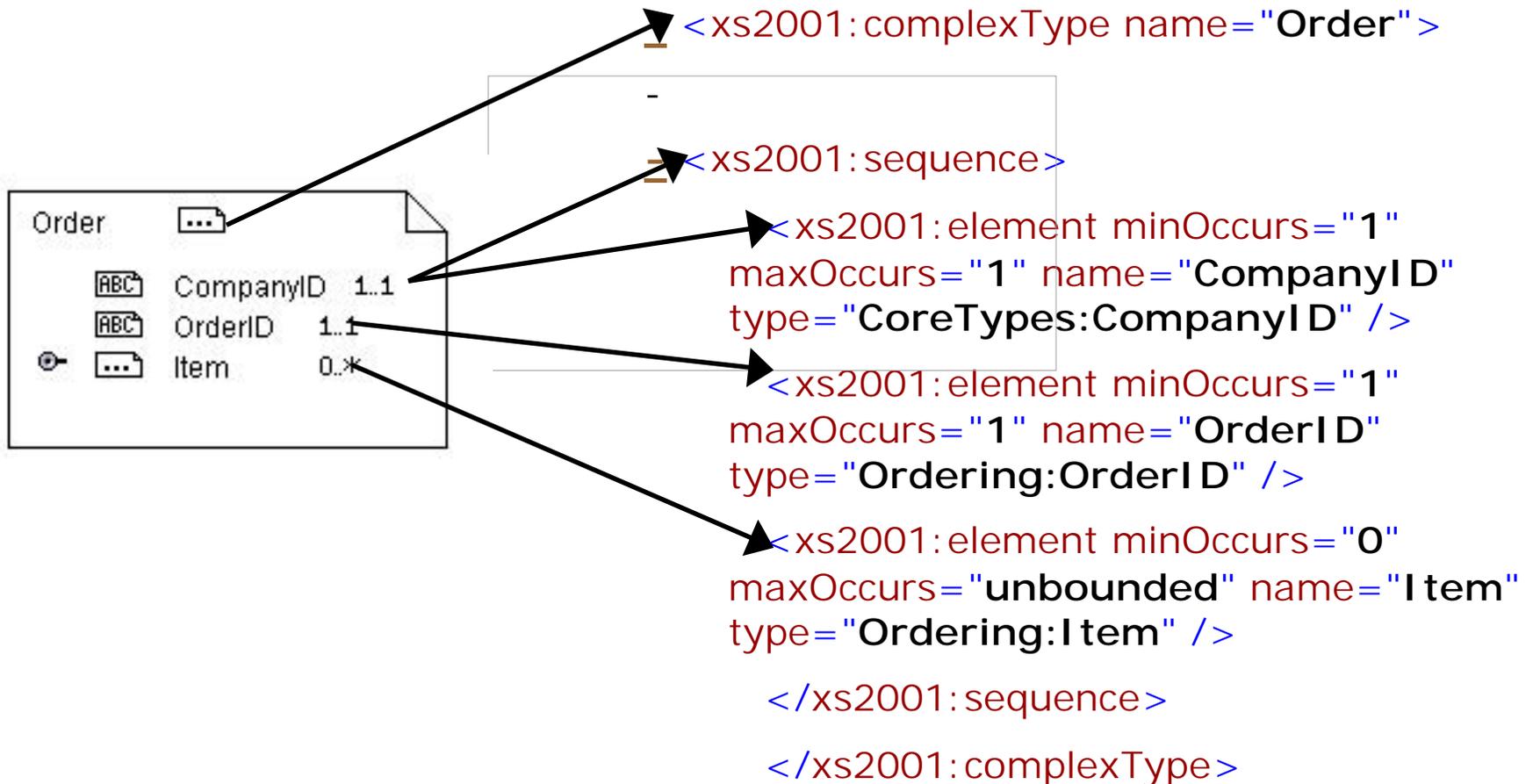


Aspects  
WSDL  
WSDL-SOAP

=

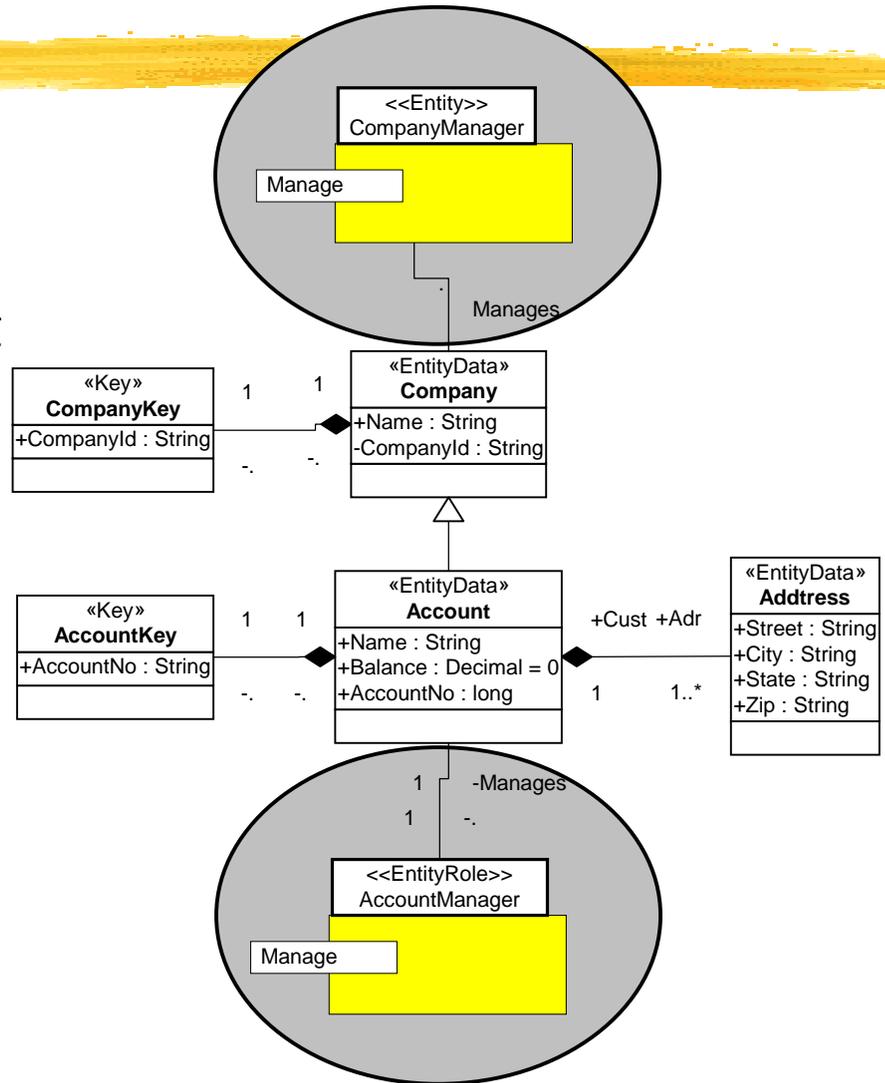
```
<message name="Order">  
  <part name="Order" type="Ordering:Order" />  
</message>  
<message name="OrderConfirmation">  
  <part name="OrderConfirmation"  
    type="Ordering:OrderConfirmation" />  
</message>  
<message name="OrderDenied">  
  <part name="OrderDenied"  
    type="Ordering:OrderDenied" />  
</message>
```

# Mapping of data types

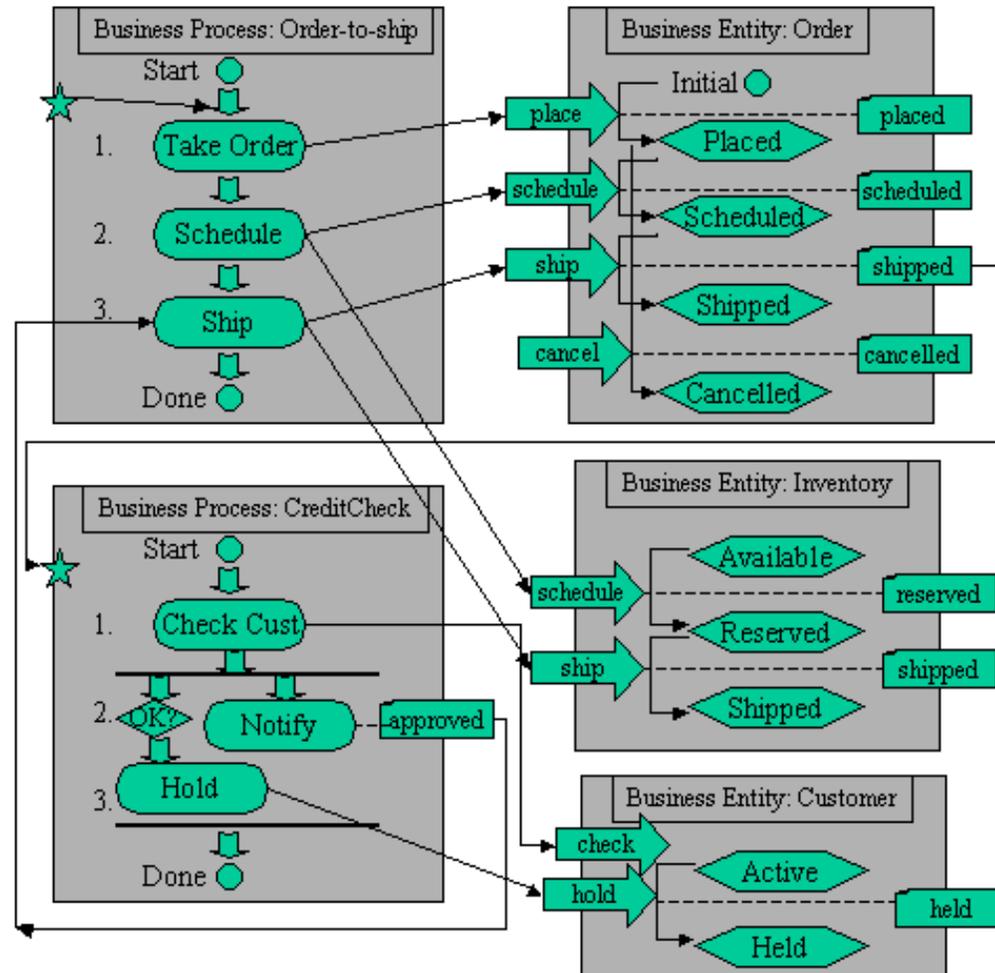


# Adding Entities

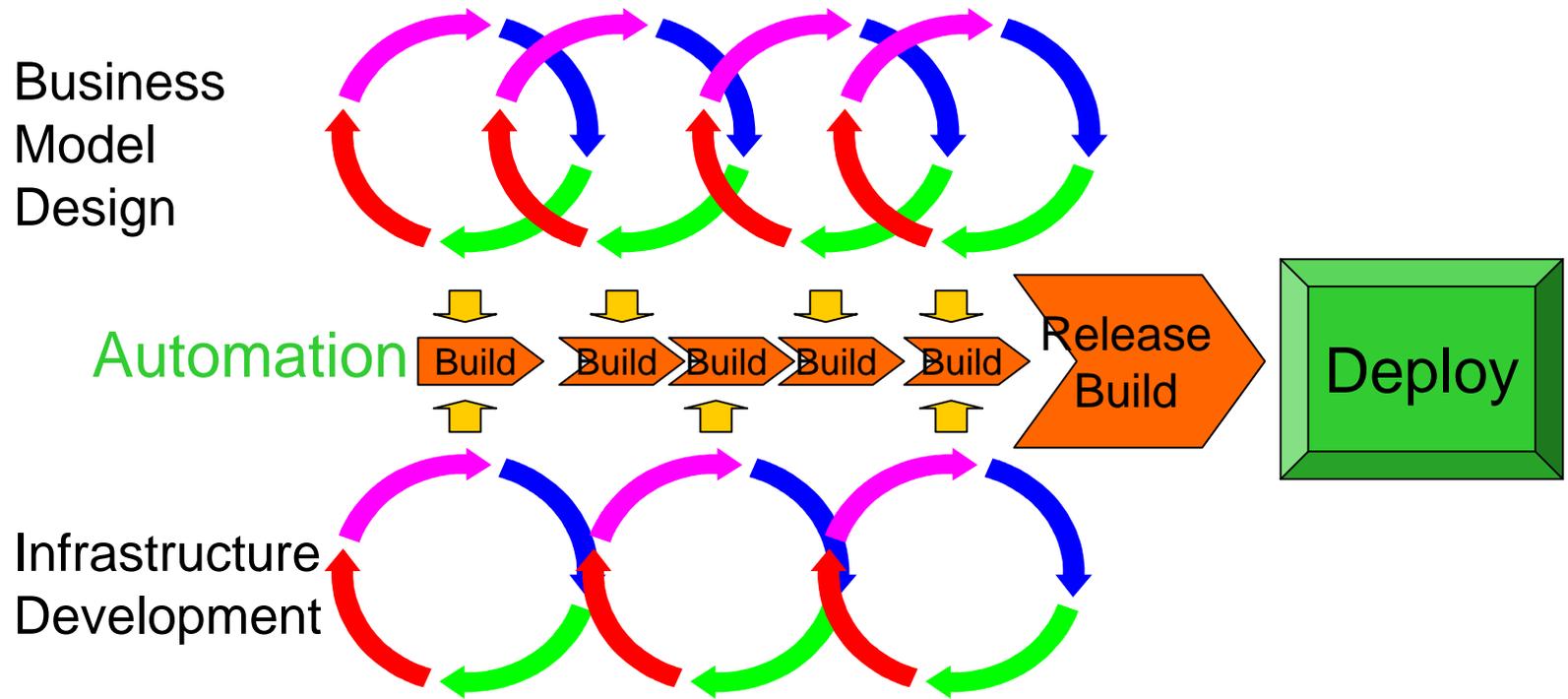
- Entities are added to manage entity data
- Entity Roles are managers that provides a view of the same identity in another context
- The Entities have ports for managing and accessing the entities
- Non-entities which are owned by (aggregate into) an entity are managed by the entity



# Event Tied to Information



# Iterative Development



# MDA Solution Factory



- Put together the
  - Best practices
  - Expertise
  - Enterprise Architecture
  - Infrastructure
  - Automated tooling
- To produce and integrate robust business collaborations quickly & reliably

# Net effect



- Using these open standards and automated techniques we can;
  - Achieve the strategic advantage of an open and flexible enterprise
  - Produce and/or integrate these systems FASTER and CHEAPER than could be done with legacy techniques
  - Provide a lasting software asset that will outlive the technology of the day

# Data Access Technologies



## ■ Products & Technologies

### ■ Component-X (Product)

- | Implements the EDOC-CCA and ebXML "BPSS" using XML and Java
- | Provides drag and drop specification & assembly of enterprise components for collaborative web services

### ■ Model Driven Architecture (Technology)

- | Based on OMG-MOF & UML
- | Provides forward and reverse engineering between models and technology artifacts
- | Use to automate and integrate multiple technologies

# Information & Contact



OMG MDA & ECA

[www.omg.org/mda](http://www.omg.org/mda)

Cory Casanave

[cory-c@enterprise-component.com](mailto:cory-c@enterprise-component.com)

(305) 234-7077