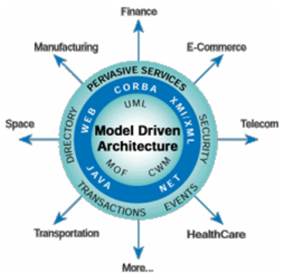
# Model Driven Web Services Development using
## UML, XML, WSDL & BPEL4WS

SI-SE 2004

University of Zurich

March 18-19, 2004



Sridhar Iyengar
Distinguished Engineer, IBM Rational Software
Member, OMG Architecture Board
*siyengar@us.ibm.com*

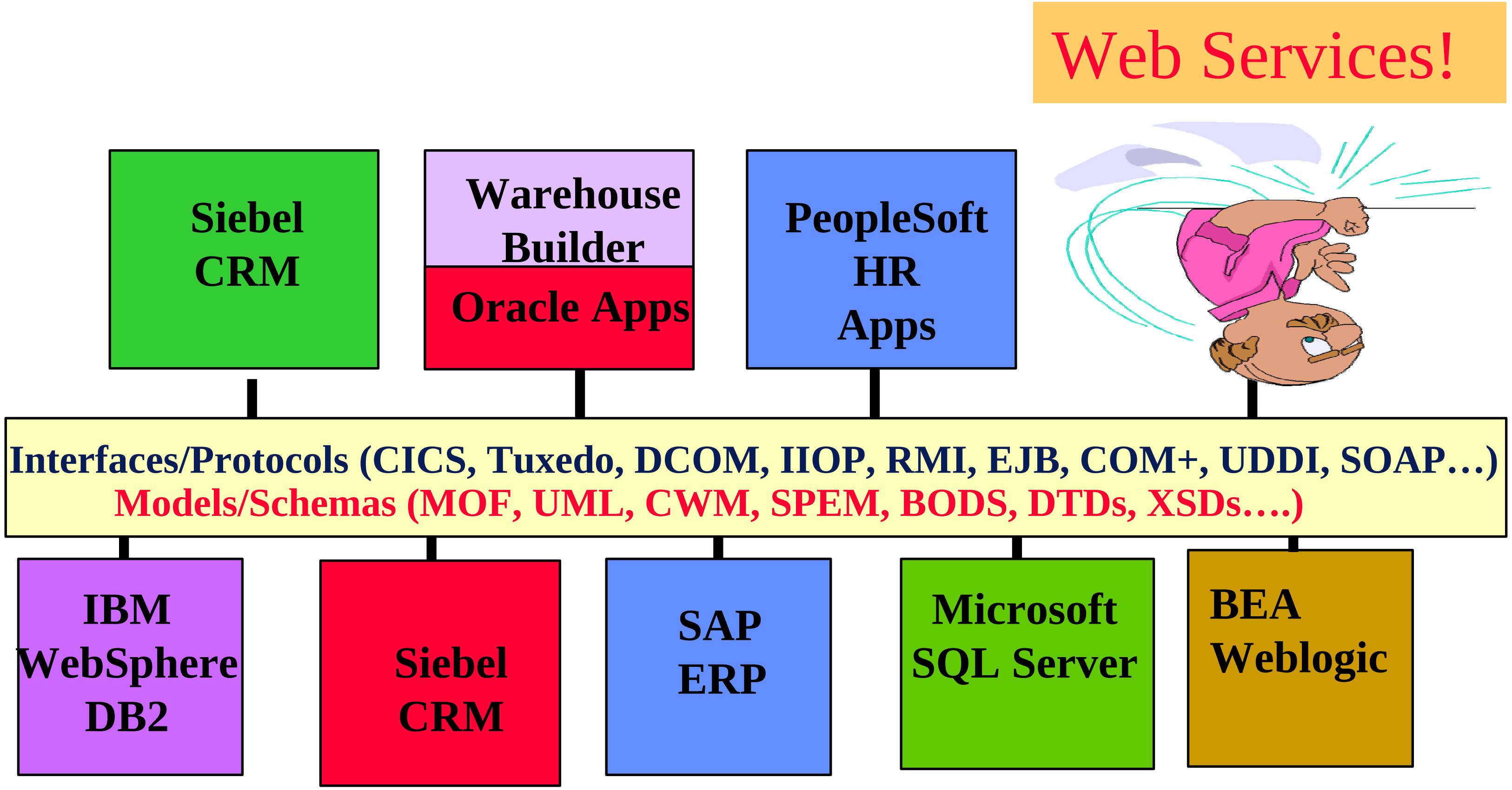
# Presentation Overview

- Introduction - The integration problem
- OMG MDA - Model Driven Architecture Overview
  - What is MDA and why is it important
  - Emerging MDA standards for Model Driven Business Integration
- Web Services Architecture Overview
  - What are Web Services and why are they important
  - Emerging Web Services Standards and Architecture
- MDA and Web Services – How do you use them together
  - Modeling Web Services
  - Advanced Modeling (Web Services Orchestration)
  - Model Driven Business Integration – A peek ahead
- Wrap-up

# Acknowledgements

- OMG - For use and reference to many OMG specs and trademarks including MDA, UML, XMI, MOF and CWM
- W3C - For use and reference to many W3C specs including XML
- OASIS – For use and reference to UDDI & BPEL4WS which is now being progressed thru OASIS
- IBM - For its continued pioneering of open standards in middleware, data management, web services and metadata
- SUN - For use and reference to many Java standards including J2EE, Java and JMI
- Other organizations and companies I will refer to in the talk

# ‘Muddleware’ Architect’s Dilemma



Web Services!

Siebel CRM

Warehouse Builder

Oracle Apps

PeopleSoft HR Apps

Interfaces/Protocols (CICS, Tuxedo, DCOM, IIOP, RMI, EJB, COM+, UDDI, SOAP…)

Models/Schemas (MOF, UML, CWM, SPEM, BODS, DTDs, XSDs….)

IBM WebSphere DB2

Siebel CRM

SAP ERP

Microsoft SQL Server

BEA Weblogic

**Distributed, Heterogeneous, Client/Server !**

***Multiple Clients, Servers, Tools, Apps, O/S, File systems, Databases, Repositories, Data Models, Object Models...***

# The Middleware Salad Bar

| Capability | J2EE | COM+ | CORBA/OMA | Web Services | OMG MDA | .Net |
|---|---|---|---|---|---|---|
| Network Layer | TCP/IP | TCP/IP | TCP/IP | TCP/IP | TCP/IP | TCP/IP |
| Web Protocol | HTTP | HTTP | HTTP | HTTP | HTTP | HTTP |
| Interface Definition | Java | Microsoft IDL | CORBA IDL | WSDL | IDL/WSDL… | WSDL/C# |
| Meta Language | XML | XML | MOF/XML | XML | MOF/XML | XML |
| RPC Mechanism | RMI/IIOP | DCOM | IIOP | SOAP; XMLP | SOAP; IIOP | SOAP |
| Registry/Repository | JNDI; LDAP | LDAP; ADSI | CORBA IR | UDDI | MOF;UDDI | UDDI |
| Process Flow | JPC | Proprietary | Proprietary | BPEL4WS.. | UML | BPEL4WS |
| Modeling Language | UML | UML | UML | UML, XSD? | UML, MOF | UML |

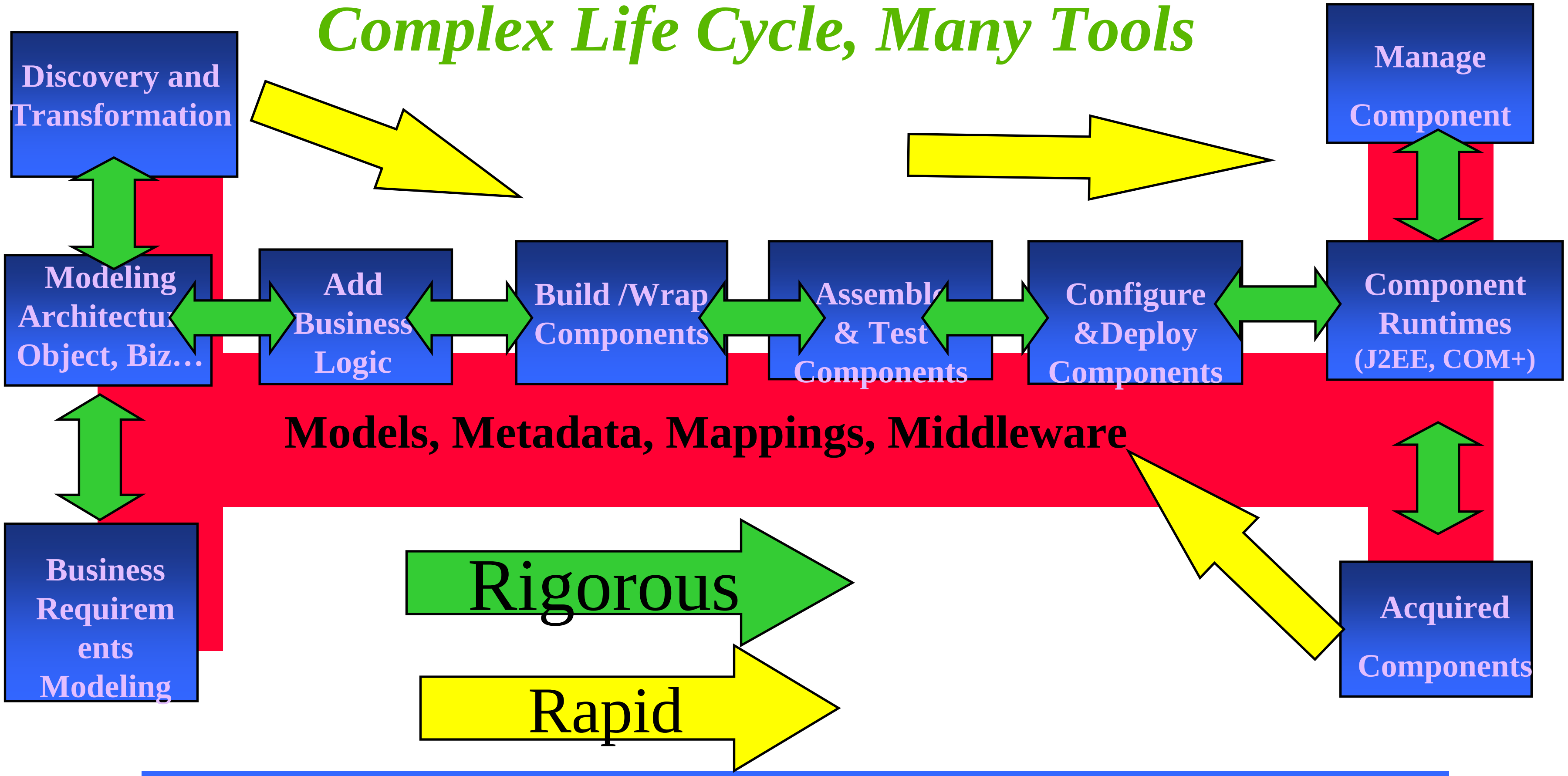
Tiny fragment of the landscape

# Solving the Integration Problem What is needed?

- Short Answer is XML and HTTP ala SOAP!
- Slightly longer answer is SOAP, WSDL, UDDI
- Even longer answer is SOAP, WSDL, UDDI, BPEL4WS, …
- The real answer is : All the buzzwords above are fine, but
  - We need to think and build software like engineers
  - Where is the process, methodology and architecture, disussion of the full lifecycle? – A focus for OMG MDA and how it bridges modeling, methods and middleware
- The OMG Architecture Board, OMG Web Services SIG, W3C Web Services Architecture WG have started to look at this issue
- First let us recap 'Web Services'!
  - And peek at some recent advances (ex: BPEL4WS)

# *The Enterprise Application Life Cycle*

## *Architecture Centric, Business driven, Complex Life Cycle, Many Tools*



Discovery and Transformation

Modeling Architectu Object, Biz…

Business Requirem ents Modeling

Add Business Logic

Build /Wrap Components

Assembl & Test Components

Configure &Deploy Components

Component Runtimes (J2EE, COM+)

Manage Component

Acquired Components

Models, Metadata, Mappings, Middleware

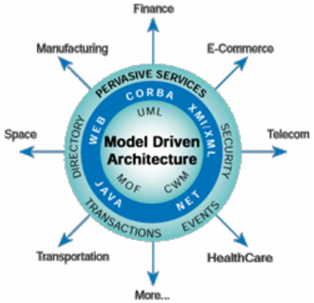
Rigorous

Rapid

**And do this with quality in a distributed environment**

# *Presentation Overview*

- Introduction - The integration problem
- OMG MDA - Model Driven Architecture Overview
  - What is MDA and why is it important
  - Emerging MDA standards for Model Driven Business Integration
- Web Services Architecture Overview
  - What are Web Services and why are they important
  - Emerging Web Services Standards and Architecture
- MDA and Web Services – How do you use them together
  - Modeling Web Services
  - Advanced Modeling (Web Services Orchestration)
  - Model Driven Business Integration – A peek ahead
- Wrap-up



# *An Overview of OMG Model Driven Architecture*

An Architectural Style that recommends the use Industry Standard Models, Metadata, Mappings (Patterns & Transformations) for integrating software.

MDA allows developers and users to productively *design, build, integrate and manage applications throughout the lifecycle* while separating technology & business concerns while 'bridging' the 'gap'

Sridhar's Usage

# OMG History

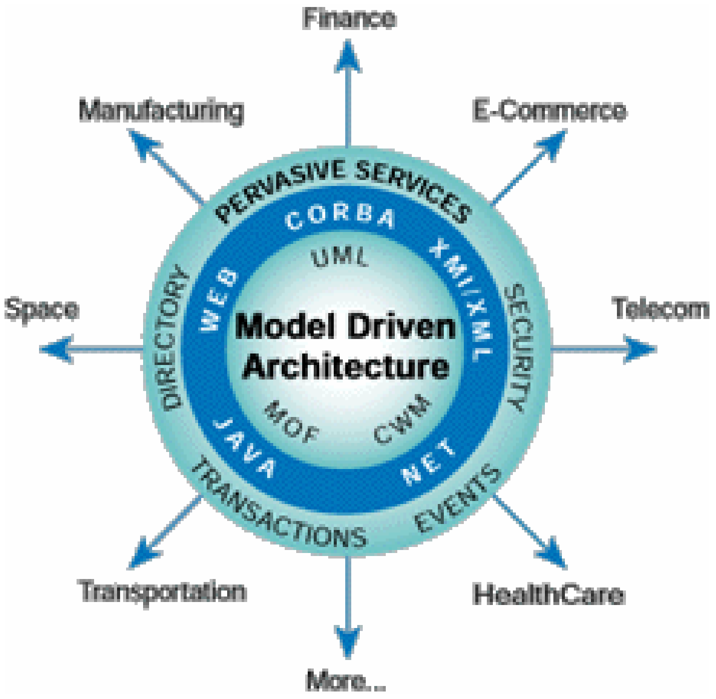
- 500+ Vendors and End User members (www.omg.org)
- 1989 to 1997 : OMA and CORBA  gain prominence
  - 1989 OMA Vision & Architecture
  - 1991 CORBA 1.0
  - 1995 CORBA 2.0 IIOP - CORBA Interoperability
- 1995 – 2001 : The foundation for MDA is established
  - 1995 UML and MOF work begins, Java arrives on the scene
  - 1997 UML and MOF adopted, Domain specs begin to be adopted
  - 1998 XML arrives on the scene, Java and XML gain momentum
  - 1999 XMI  (integration of MOF, UML and XML) adopted
  - 2000 CWM, XMI for XML Schema work begins
  - 2001 UML for EDOC, EAI, UML 2.0 work begins
- 2001 OMG unveils Model Driven Architecture - MDA
  - 2002 XMI 2 adopted, UML2 and MOF 2 proposals arrive
  - 2003 UML2 and MOF2 standards to be adopted
  - 2004 Business Rules and Business Modeling standards expected

# *OMG Analysis & Design Platform Task Force Premise*

- The use of Modeling and Metadata enabled architectures allows us to manage the complexity of software development, application integration and data warehouse management
- Modeling and Metadata standards are necessary for interchange of software artifacts and interoperability between tools, applications, middleware and data stores across platforms
- Largely influential in OMG evolution from OMA to MDA

Major efforts underway :
UML2, MOF2, Action Semantics

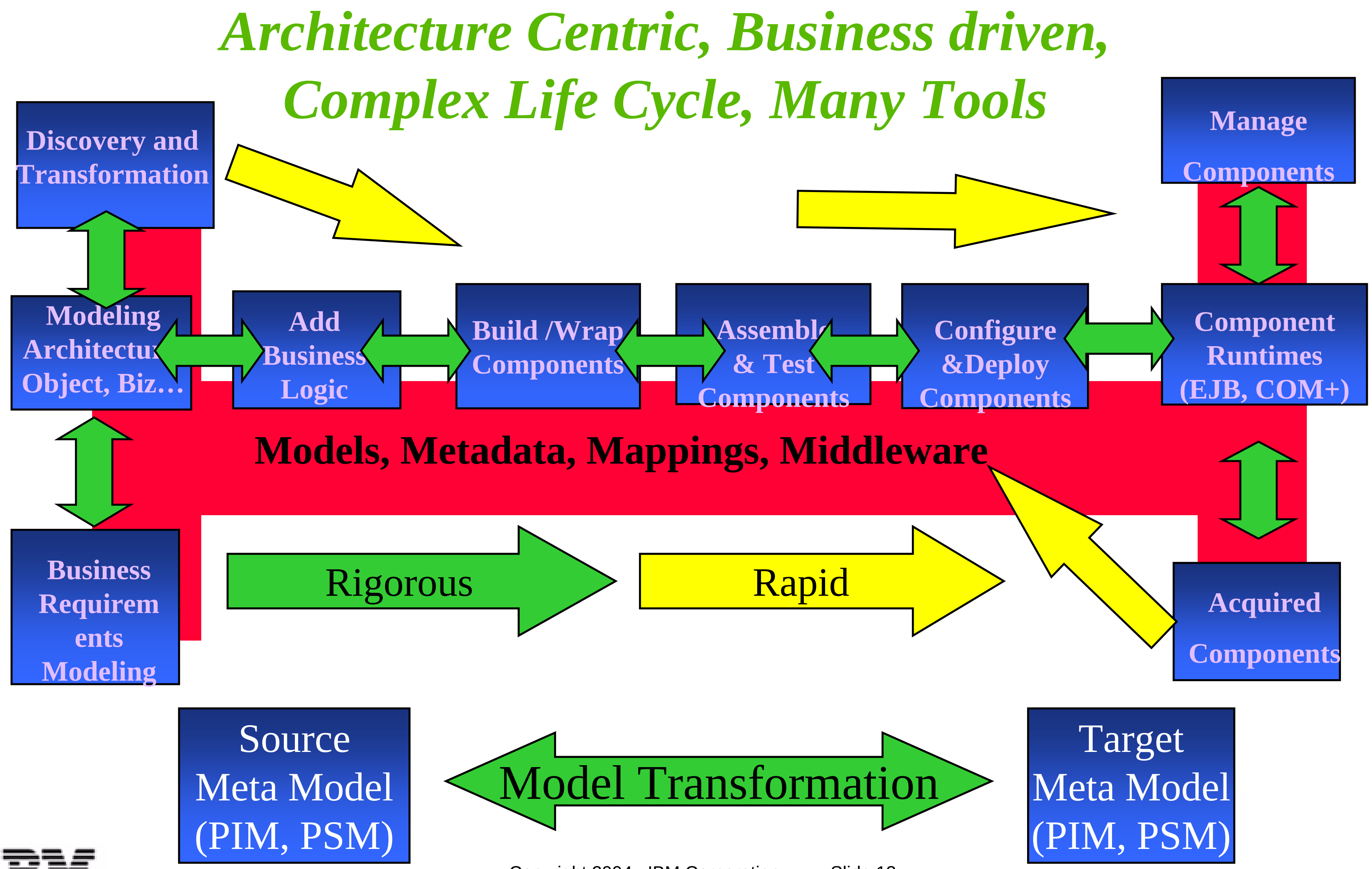
# Solution for Managing Complexity : MDA: Model Driven Architecture



- An integration of best practices in Modeling , Middleware, Metadata, Internet and Software Architecture
- Model Driven (UML, MOF, CWM…)
  - Platform Independent Models (PIM)
  - Platform Specific Models (PSM)
  - Mappings : PIM <==> PSM,
  - Applies across the software life cycle
- Key Benefits
  - Improved Productivity for Architects, Designers, Developers and Administrators
  - Lower cost of Application Development and Management
  - Enhanced Portability and Interoperability
  - Business Models and Technologies evolve at own pace on platform(s) of choice

* PIM usually applies to a class of platforms/protocols (ex: Messaging)

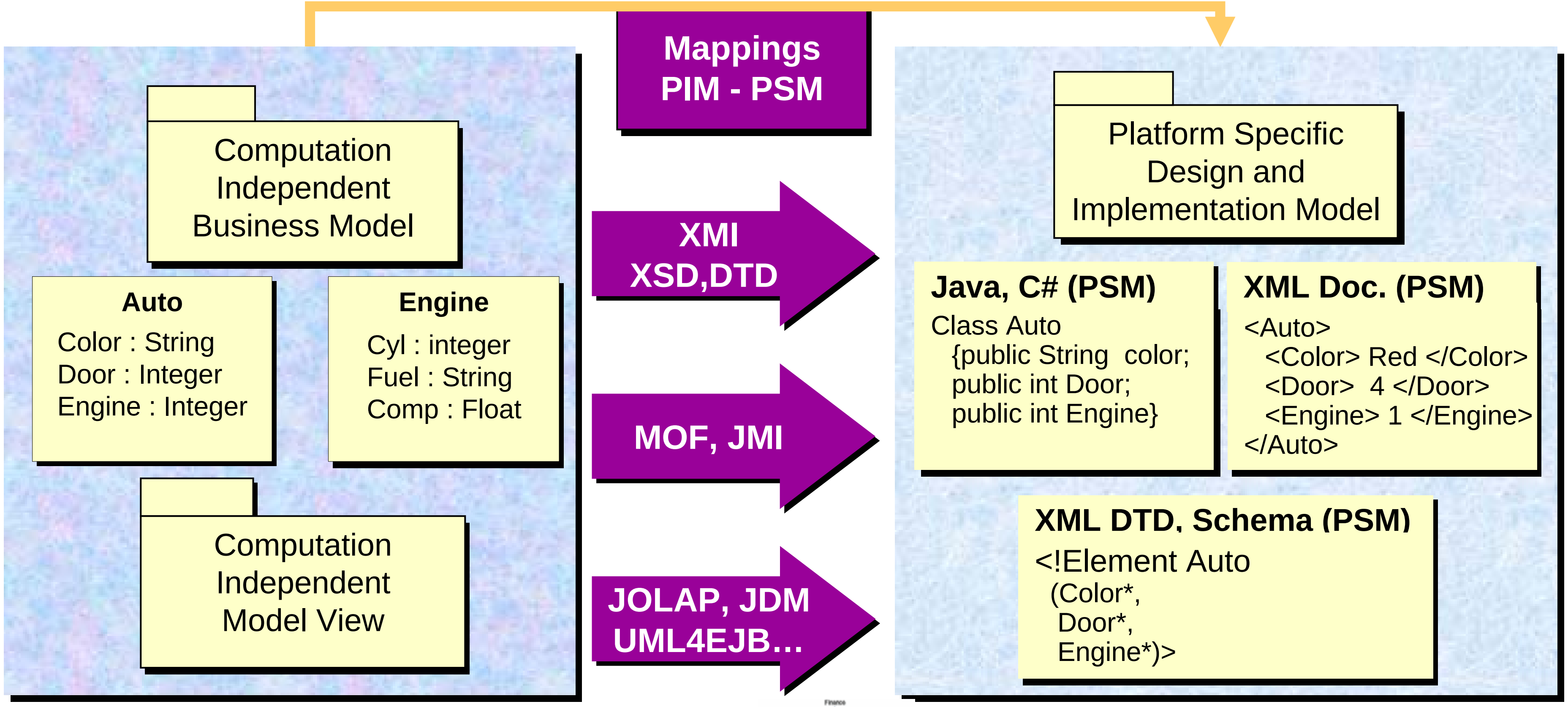
# The Enterprise Application Life Cycle


Architecture Centric, Business driven,
Complex Life Cycle, Many Tools
Discovery and Transformation
Manage Components
Modeling Architectur Object, Biz…
Add Business Logic
Build /Wrap Components
Assembl & Test Components
Configure &Deploy Components
Component Runtimes (EJB, COM+)
Models, Metadata, Mappings, Middleware
Business Requirem ents Modeling
Rigorous
Rapid
Acquired Components
Source Meta Model (PIM, PSM)
Model Transformation
Target Meta Model (PIM, PSM)

# MDA as an 'Architectural Style' for Model Driven Integration

- Understand the problem domain (technology or business)
- Model the problem domain
  - Use UML for the visual modeling, analysis & design of meta model
  - Use a UML compliant tool/repository that supports XMI import/export
  - Opportunity : Make UML more data/information/web services modeler friendly, provide E/R visualization
- Formally represent the metamodel semantics using MOF
  - Simple class modeling is all you need to know
  - OCL (Object Constraint Language) can capture additional semantics
  - Reverse engineer existing DTD, XSD, XMI, Java to MOF (jump start)
- Use Standard transformation (mappings) patterns for
  - Metadata Interchange (XMI – MOF to XML, DTD, XSD)
  - Metadata Interfaces (JMI – MOF to Java, MOF to IDL etc.)
- Use open source meta modeling frameworks for metadata management
  - Eclipse EMF : www.eclipse.org
  - Netbeans MDR : www.netbeans.org
- Summary : Understand, Model, Map and Manage metadata to integrate

# Platform Independence using OMG Model Driven Architecture (MDA)



Mappings
PIM - PSM

Computation
Independent
Business Model

**Auto**
Color : String
Door : Integer
Engine : Integer

**Engine**
Cyl : integer
Fuel : String
Comp : Float

Computation
Independent
Model View

**XMI**
**XSD,DTD**

**MOF, JMI**

**JOLAP, JDM**
**UML4EJB…**

Platform Specific
Design and
Implementation Model

**Java, C# (PSM)**

```
Class Auto
  {public String  color;
  public int Door;
  public int Engine}
```

**XML Doc. (PSM)**

```
<Auto>
  <Color> Red </Color>
  <Door>  4 </Door>
  <Engine> 1 </Engine>
</Auto>
```

**XML DTD, Schema (PSM)**

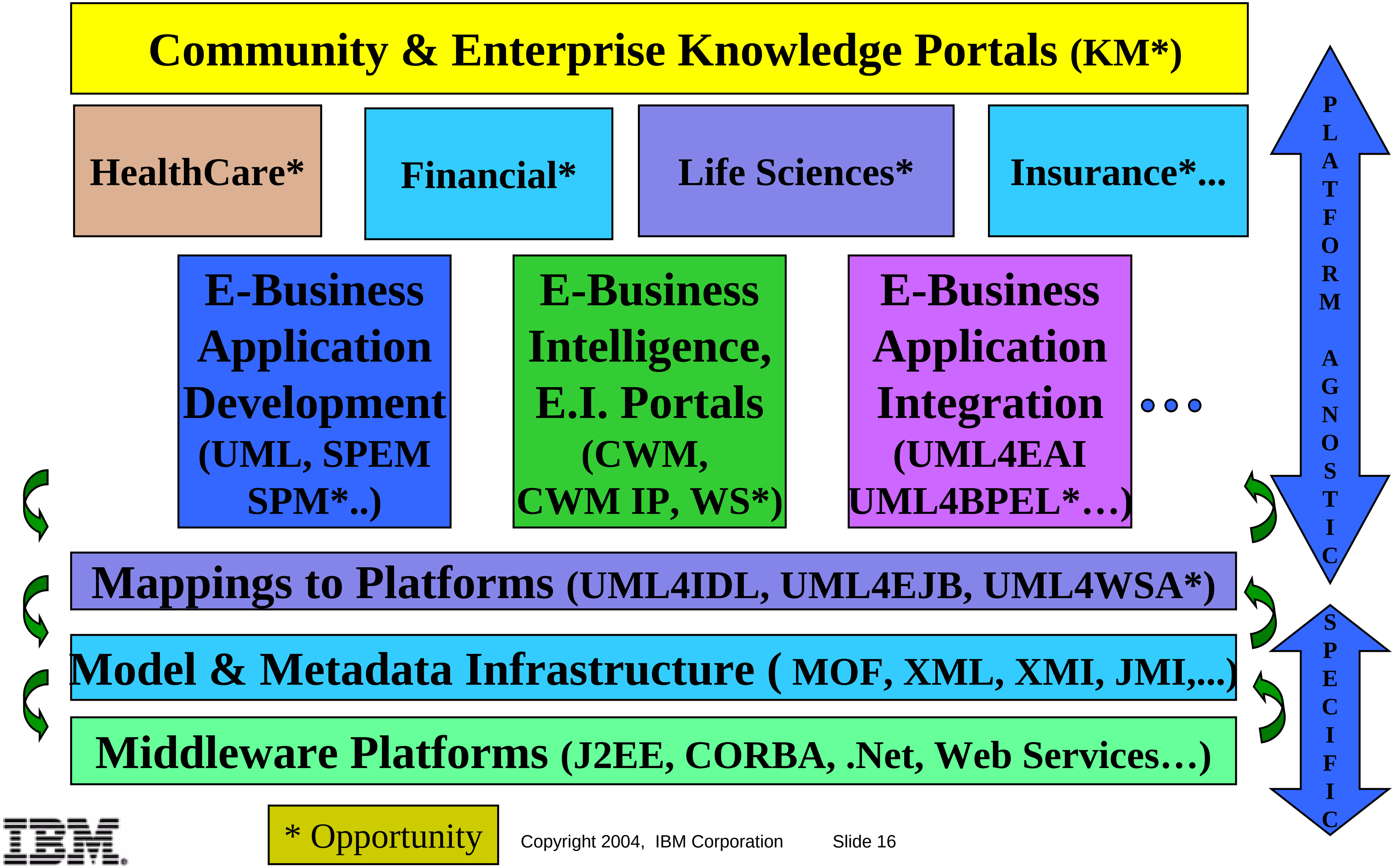
```
<!Element Auto
  (Color*,
   Door*,
   Engine*)>
```

**Platform Independent Models (PIM)**

**Platform Specific Models  (PSM)**

Mappings to UDDI and WSDL being defined

# OMG MDA - Specific Models/Work in Progress



Community & Enterprise Knowledge Portals (KM*)

HealthCare*

Financial*

Life Sciences*

Insurance*...

E-Business Application Development (UML, SPEM SPM*..)

E-Business Intelligence, E.I. Portals (CWM, CWM IP, WS*)

E-Business Application Integration (UML4EAI UML4BPEL*…)

• • •

Mappings to Platforms (UML4IDL, UML4EJB, UML4WSA*)

Model & Metadata Infrastructure ( MOF, XML, XMI, JMI,...)

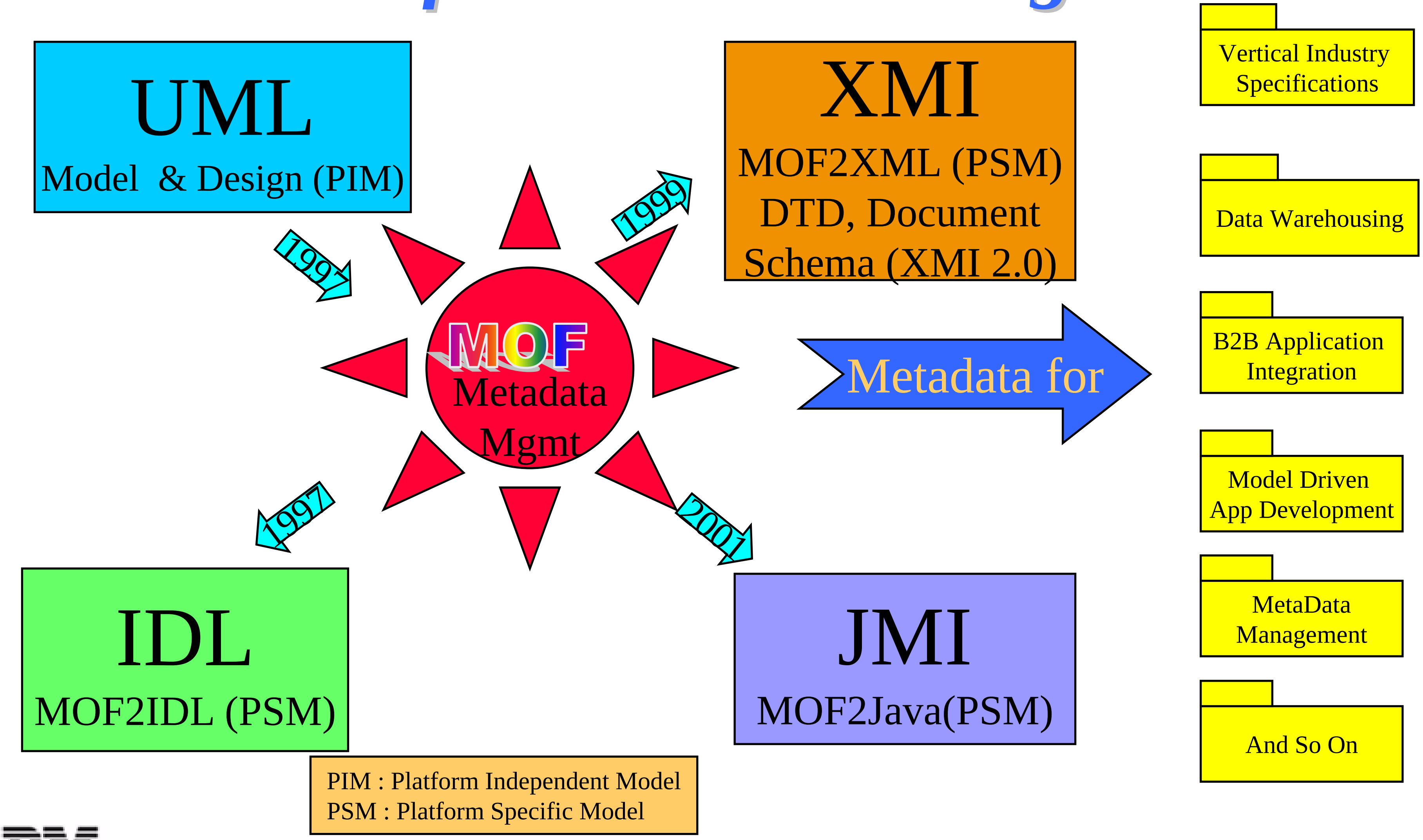
Middleware Platforms (J2EE, CORBA, .Net, Web Services…)

* Opportunity

# *Early examples of MDA*

- OMG has been using MDA principles for about 5 years in several specifications starting with MOF & UML in 1997
- Metadata and Object Interchange/Integration
  - UML and MOF for modeling, MOF and XMI for metadata integration
  - MOF is a subset of UML (Class diagrams) used to model metadata
    - MOF 2.0 and UML 2.0 reuse a common UML2 Infrastructure
    - Essential MOF (EMOF) compliance pointt
- Application Development
  - UML for modeling, XMI and IDL for tool integration and interchange
  - UML profile EJB, CORBA : UML for modeling, J2EE for Integration
- Data Warehousing
  - UML and CWM for Modeling metadata, XMI, JMI, JOLAP, JDM for Warehouse integration
- Application Integration
  - UML profile for Enterprise Application Integration (UML4EAI)

# OMG Model Driven Architecture
# MDA for Metadata Integration

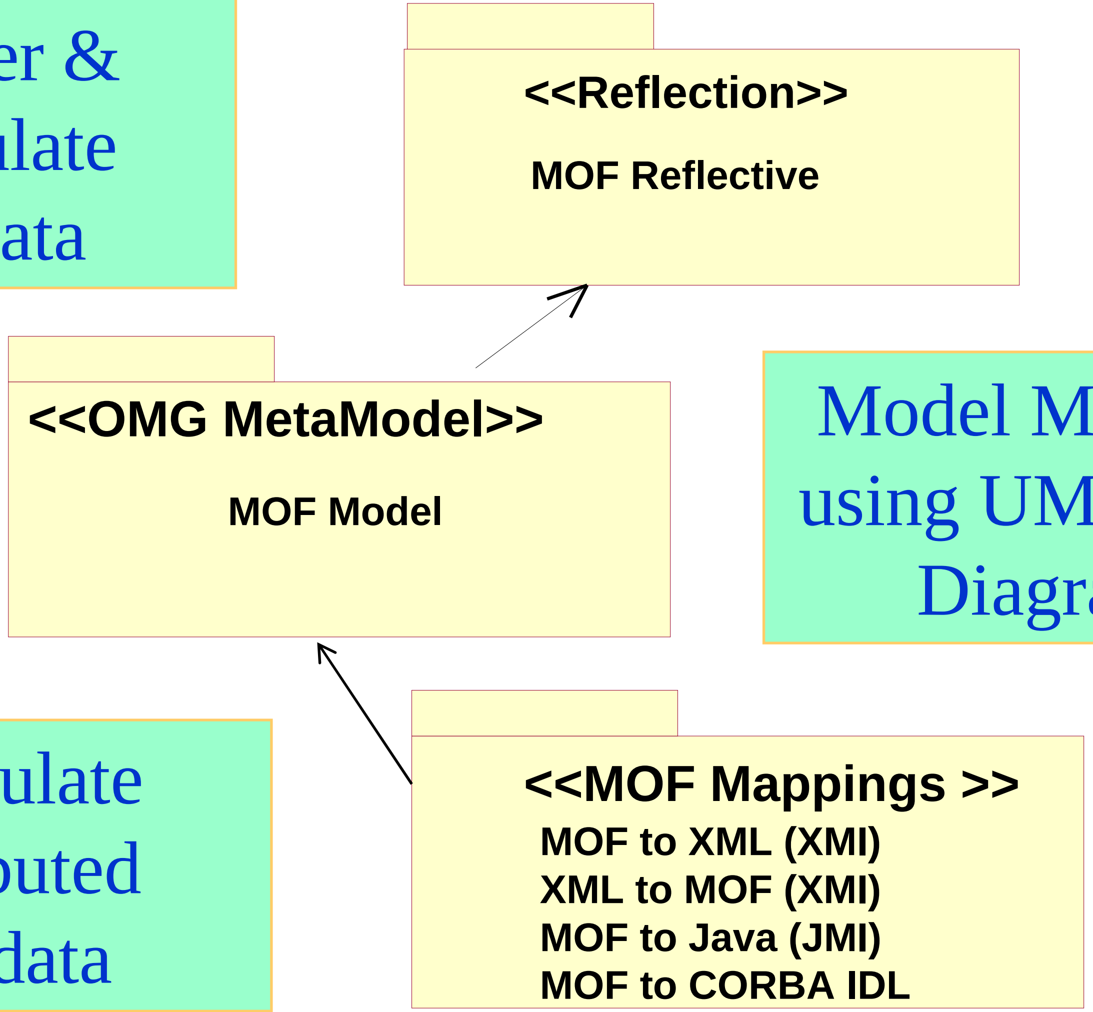


PIM : Platform Independent Model
PSM : Platform Specific Model

# Meta Object Facility Overview

## Current: MOF 1.4 , New MOF 2.0

Discover & Manipulate Metadata



<<Reflection>>

MOF Reflective

<<OMG MetaModel>>

MOF Model

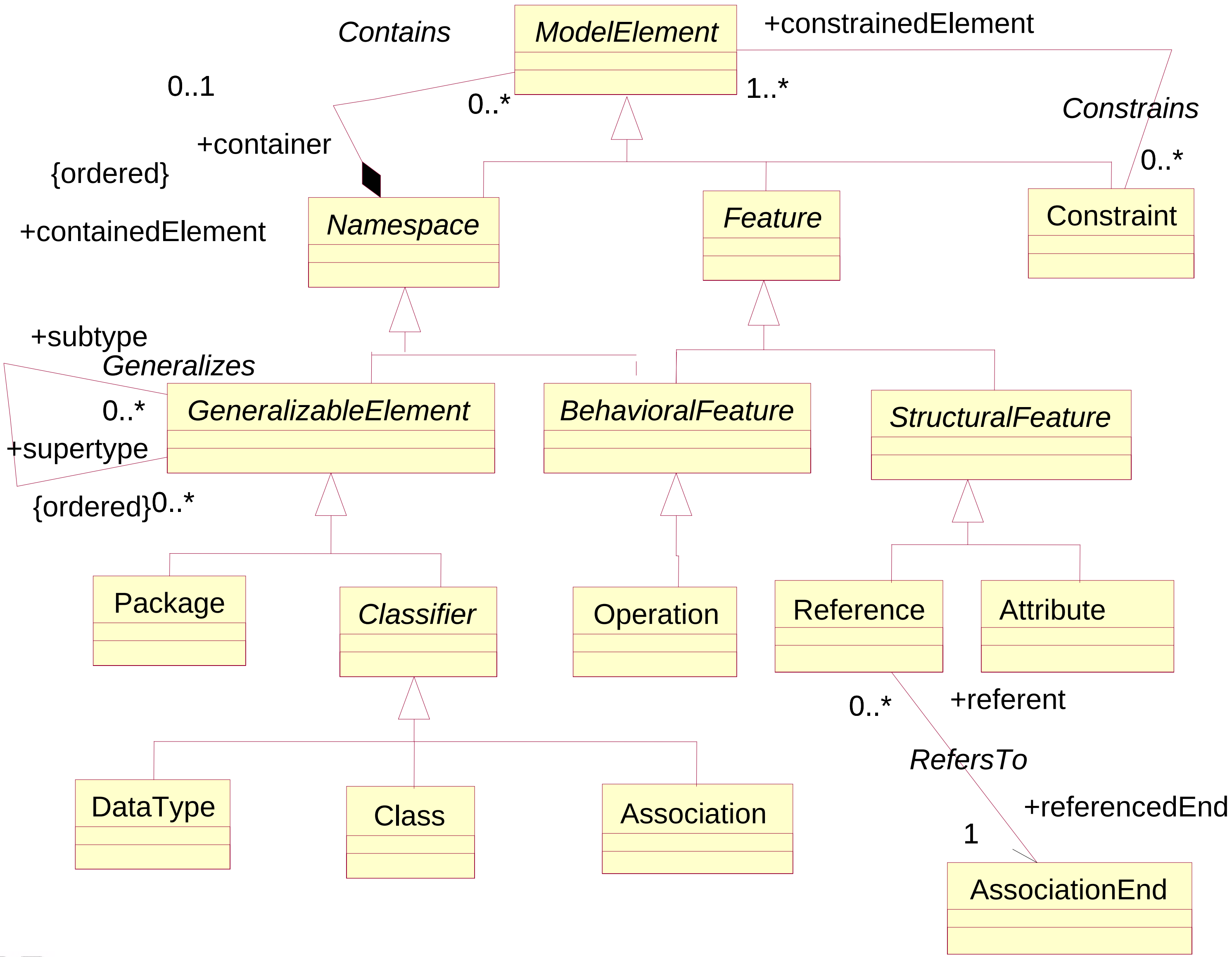
<<MOF Mappings >>

MOF to XML (XMI)
XML to MOF (XMI)
MOF to Java (JMI)
MOF to CORBA IDL

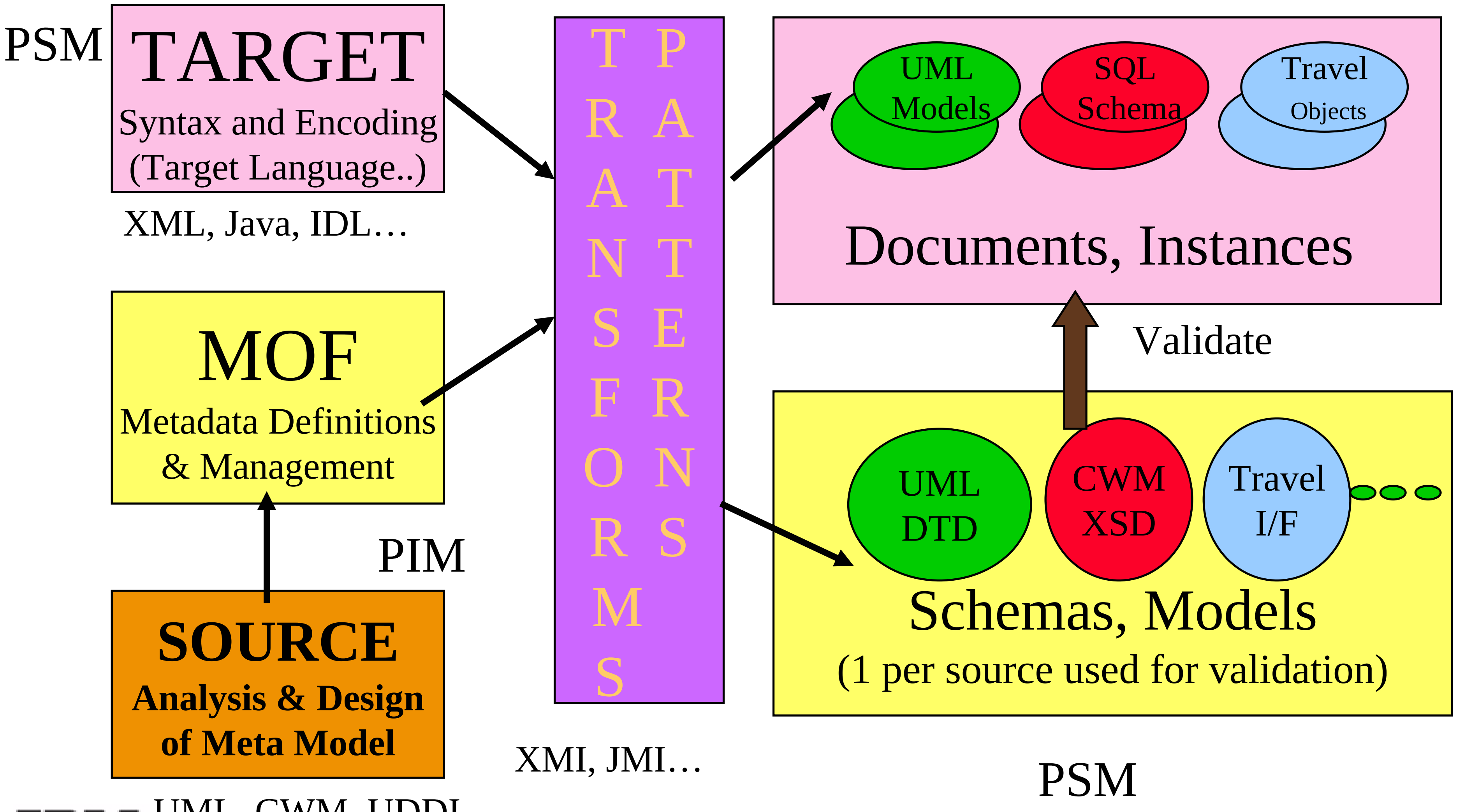
Model Metadata using UML Class Diagrams

Manipulate Distributed Metadata

# OMG MOF 1.4 Model Subset



# Applying the MDA Architectural Style for Metadata Management



PSM

TARGET

Syntax and Encoding (Target Language..)

XML, Java, IDL…

MOF

Metadata Definitions & Management

PIM

SOURCE

**Analysis & Design of Meta Model**

TRANSFORMS

PATTERNS

XMI, JMI…

UML Models

SQL Schema

Travel Objects

Documents, Instances

Validate

UML DTD

CWM XSD

Travel I/F

Schemas, Models

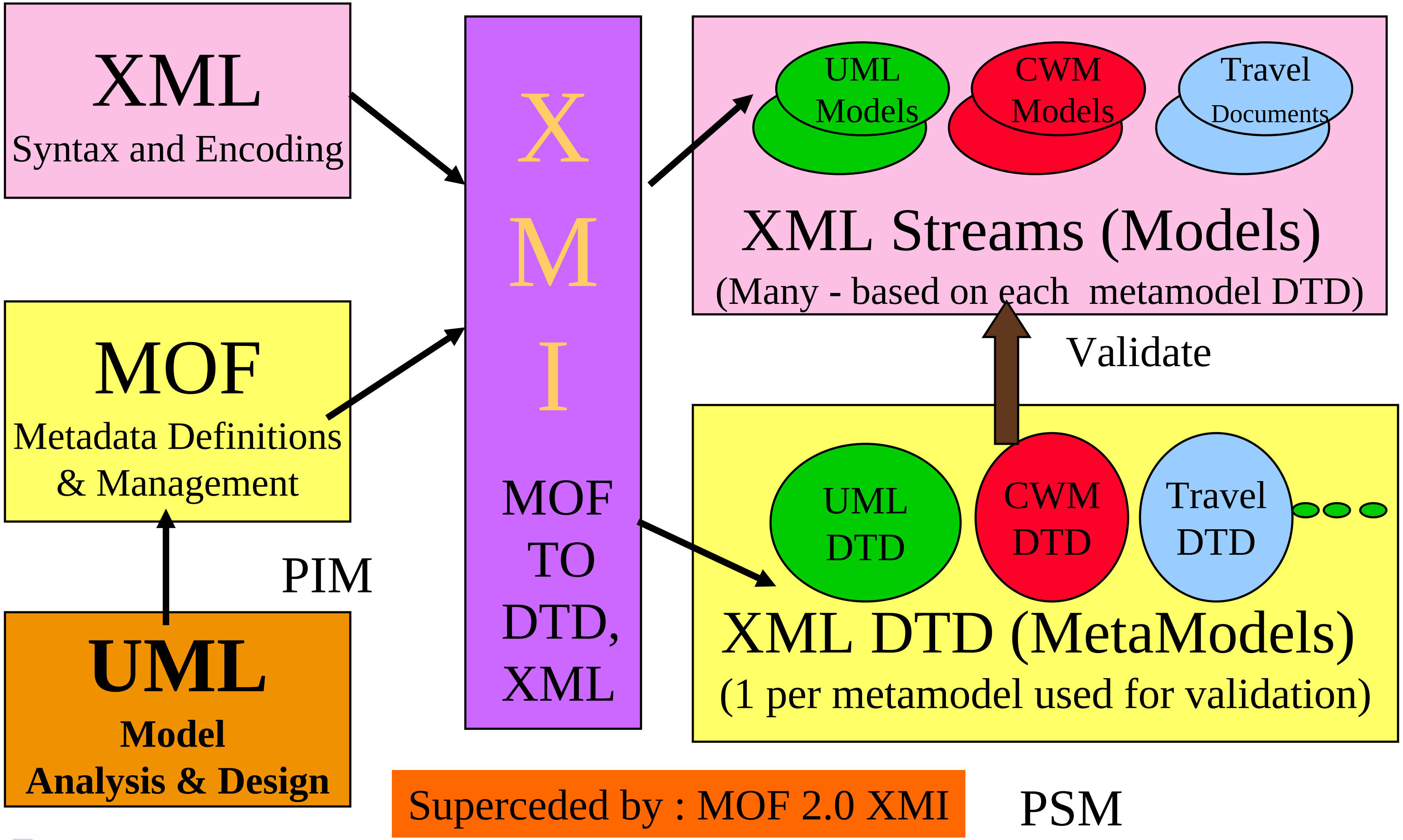
(1 per source used for validation)

PSM

UML, CWM, UDDI…

# Transform Models to DTD

## XMI 1.1 or XMI 1.2 (1999, 2001)



XML

Syntax and Encoding

MOF

Metadata Definitions & Management

UML

Model

Analysis & Design

PIM

XMI

MOF TO DTD, XML

XML Streams (Models)

(Many - based on each metamodel DTD)

Validate

XML DTD (MetaModels)

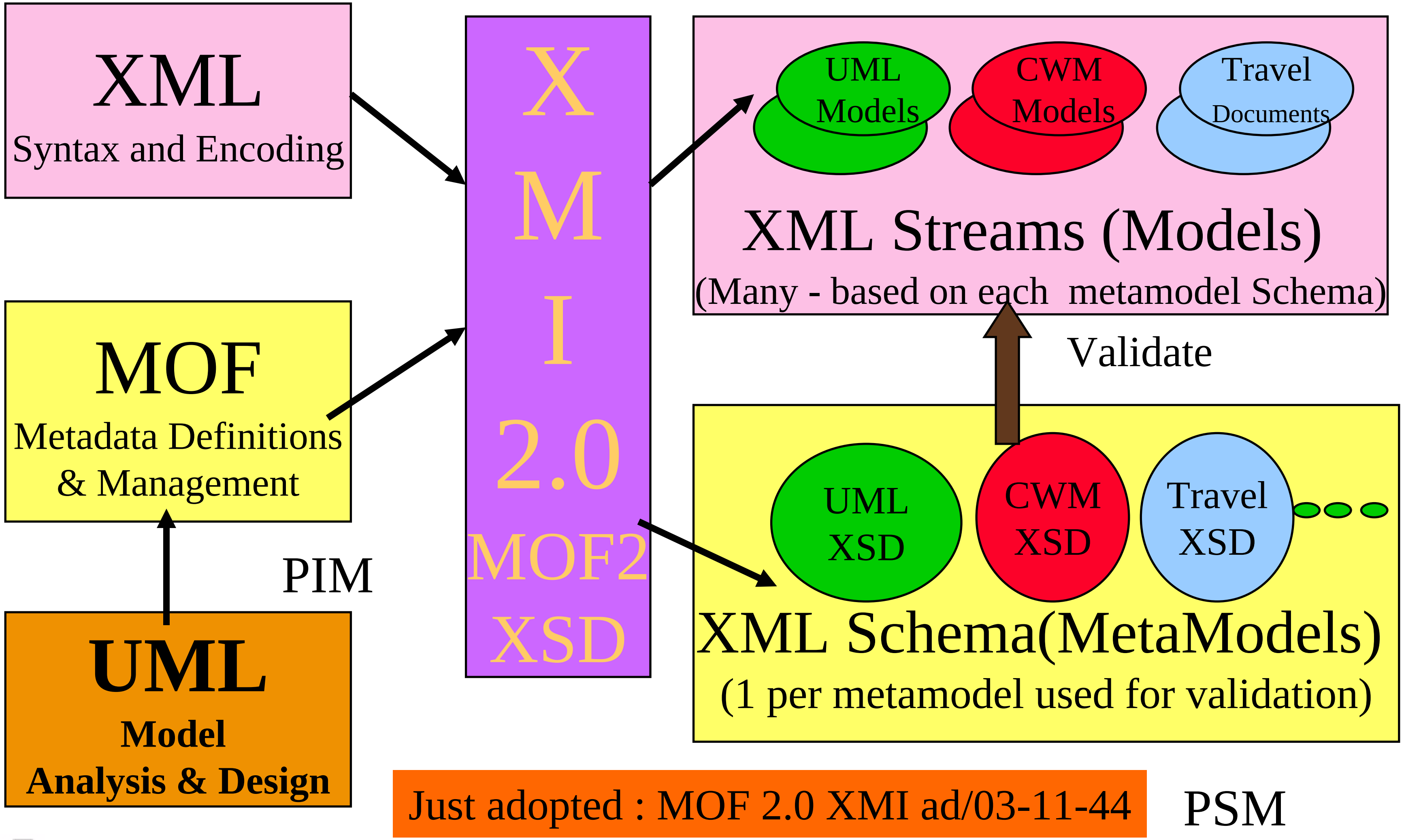
(1 per metamodel used for validation)

Superceded by : MOF 2.0 XMI

PSM

# Transform UML Models to Schemas

## XMI 2.0 OMG Document : ad/01-06-12,13 (2002)



XML
Syntax and Encoding

MOF
Metadata Definitions & Management

UML
Model
Analysis & Design

PIM

XMI 2.0 MOF2 XSD

XML Streams (Models)
(Many - based on each metamodel Schema)

Validate

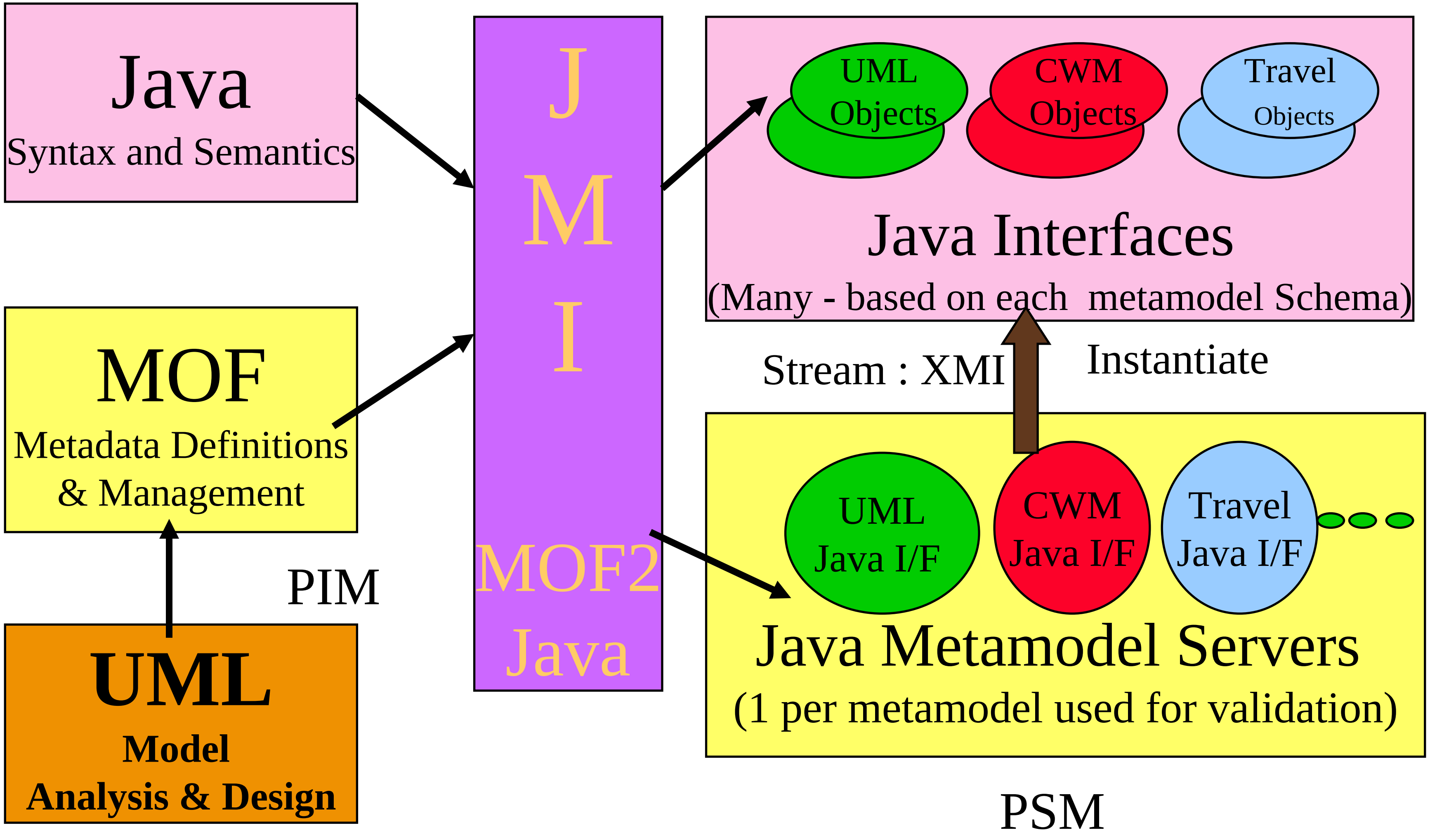
XML Schema(MetaModels)
(1 per metamodel used for validation)

Just adopted : MOF 2.0 XMI ad/03-11-44

PSM

# Transform Models to Java

## JSR-40 : Java Metadata Interface (JMI) 2002



Java

Syntax and Semantics

MOF

Metadata Definitions & Management

UML

**Model**

**Analysis & Design**

PIM

J M I

MOF2 Java

Java Interfaces

(Many - based on each metamodel Schema)

Stream : XMI

Instantiate

Java Metamodel Servers

(1 per metamodel used for validation)

PSM

# The 4Ms of MDA

# Models, Metadata, Mappings, Middleware

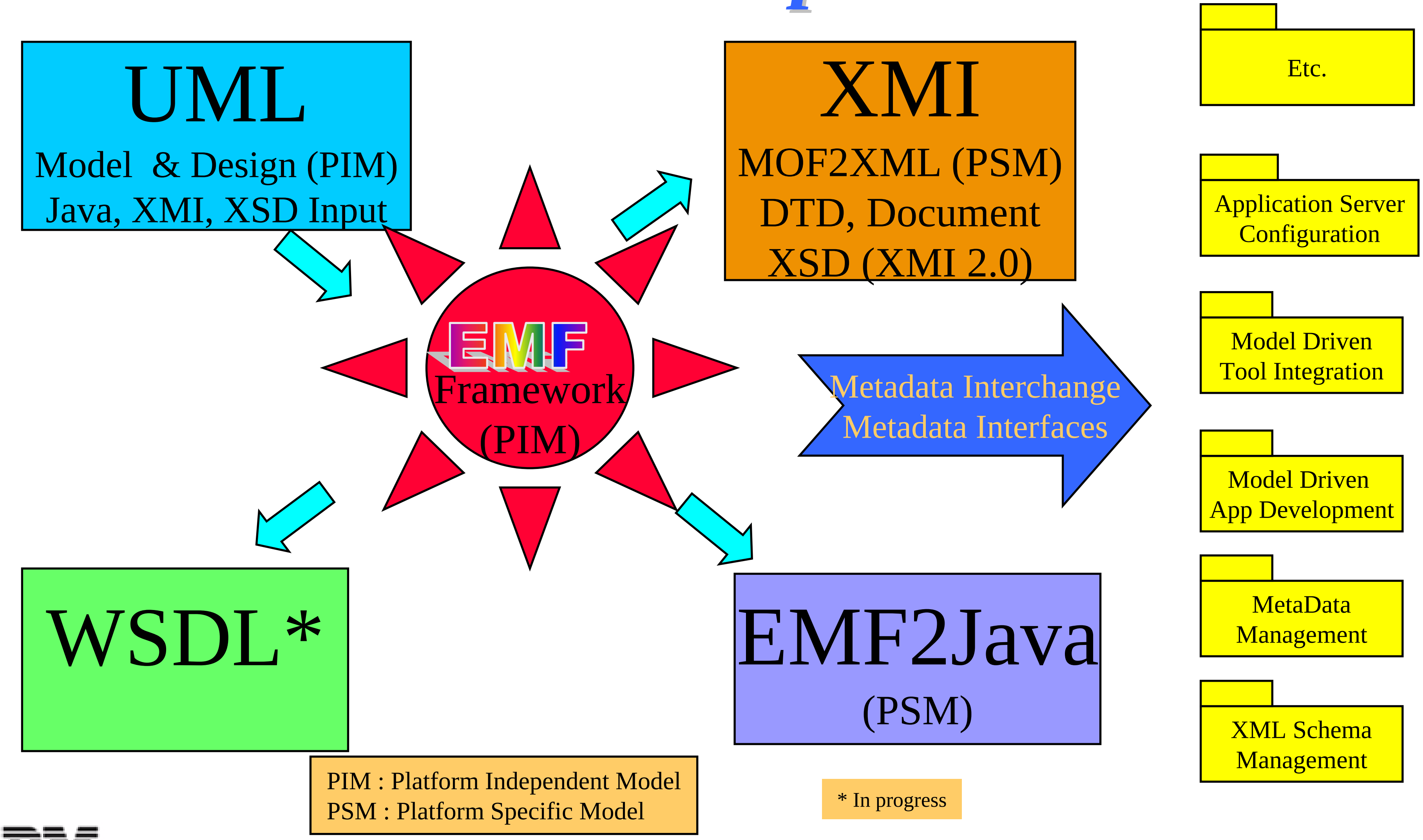
| Problem Domain | Source Meta Model (PIM) | Mapping Specification | Target Middleware, Language (PSM) | Interchange Format/API | Comments | Status |
|---|---|---|---|---|---|---|
| Application Development<br>OO Analysis & Design | UML | UML Profile for CORBA | CORBA | XMI | OMG Spec | Adopted 1997 |
| Software Process Engineering | UML | UML Profile for SPEM, SPEM Metamodel | Agnostic | XMI | | Adopted 2001 |
| Metadata Management | MOF | MOF to IDL | CORBA | IDL interfaces | PIM for metadata | Adopted 1997 |
| | MOF | MOF to XML DTD (XMI)<br>MOF to XML Docs (XMI) | XML DTD,<br>XML Documents | XMI | Common Interchange Format | Adopted 1999 |
| | MOF | MOF to XML Schema (XMI)<br>XML to MOF (Reverse) | XML Schema,<br>XML Documents | XMI, Native XSD | Common Interchange Format | Adopted 2002 |
| | MOF | MOF to Java (JMI) | Java | JMI, XMI | Common Programming Model for MOF | Adopted 2002 |
| Enterprise Components/ Middleware | CORBA (CCM spec) | CORBA Component Model | CORBA 3.0 | IDL, XMI | First metamodel of the CORBA platform itself | Adopted 2001 |
| Enterprise Components / Middleware | UML | UML Profile for EJB | J2EE | XMI, JMI | JSR-26 Java Community process | Public Final Draft 2002 |
| Enterprise Components/ Middleware | UML | UML Profile for EDOC; CCA, EJB and Java Metamodels | J2EE, Java and EJB | XMI, IDL (JMI) | Reuse UML Profile for EJB | Adopted 2002 |
| Data Warehousing<br>Business Intelligence | CWM | MOF to IDL, XMI, JMI* | Various Database Middleware | XMI, JMI* | JMI via Java for OLAP and Java for Data Mining | Adopted 2001 |
| | CWM | Java for OLAP | OLAP - Multi Dimensional Database | JOLAP, XMI | Uses JMI | Public Draft 2002 |
| | CWM | Java for Data Mining | Data Mining, Data base middleware | JDM, XMI | Uses JMI | Public Draft 2002 |
| | CWM | CWM for Web Services | Database Middleware | SOAP, UDDI and WSDL | Uses MOF to WSDL and UDDI mapping | Initial Submission |
| Web Services | MOF | MOF to WSDL | WSDL/SOAP | WSDL | Used by CWM for Web Services | Initial Submission |
| | MOF | MOF to UDDI | UDDI/SOAP | UDDI | Used by CWM for Web Services | Initial Submission |

UML2.0, MOF 2.0, MOF 2.0 XMI just adopted

# *How are IBM and IBM partners using MDA for Tool Integration*

- Use Eclipse tools integration framework
  - www.eclipse.org (Java IDE, tools integration framework)
- Use Eclipse EMF for modeling & metadata management
  - www.eclipse.org/emf, www.eclipse.org/xsd
  - Simplification of OMG MOF 1.4 with support for XMI 2.0
  - Use UML models, XMI, XSD or Annotated Java as metamodel input
  - Code generation for metadata interfaces and implementation
  - XML interchange (XMI, XSD serialization)
- Used in IBM WebSphere Studio & in WebSphere Application Server (ex: configuration metadata), Rational, Borland..
- Expect the usage to increase across software development lifecycle
  - UML modeling, testing, software quality improvement…
- Research area : ‘Model Driven Business Integration’

# OMG Model Driven Architecture
# EMF in WebSphere Tools



UML

Model & Design (PIM)
Java, XMI, XSD Input

XMI

MOF2XML (PSM)
DTD, Document
XSD (XMI 2.0)

EMF
Framework
(PIM)

Metadata Interchange
Metadata Interfaces

WSDL*

EMF2Java
(PSM)

Etc.

Application Server Configuration

Model Driven Tool Integration

Model Driven App Development

MetaData Management

XML Schema Management

PIM : Platform Independent Model
PSM : Platform Specific Model

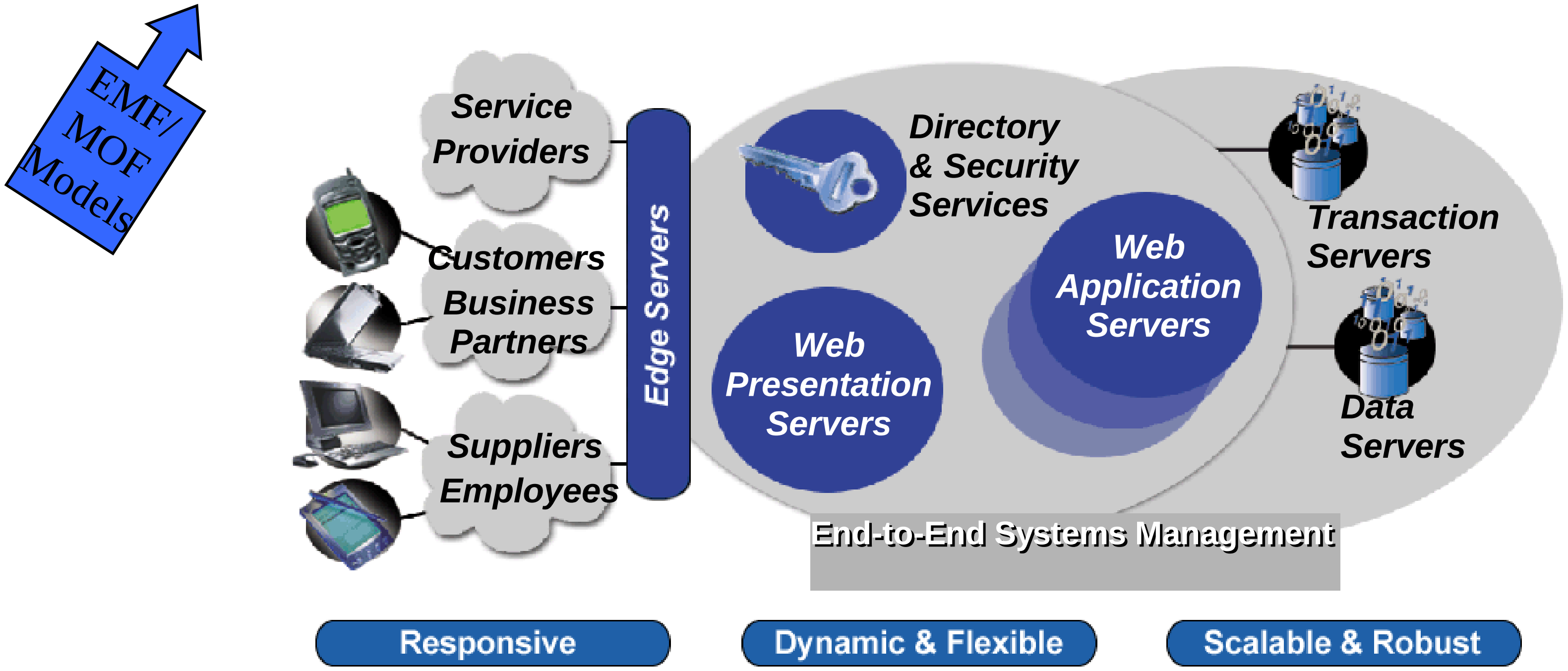
* In progress

# Sample EMF Metamodels Used in WebSphere

## Integration of key software domains

XMI | XML | HTML | JSP | EJB | Java | C/C++ | COBOL | SQL | BPM*

MOF | XSD | BPEL* | WSDL | UDDI | FCM | Mapping | WCCM | UML*



EMF/ MOF Models

Service Providers

Customers

Business Partners

Suppliers

Employees

Edge Servers

Directory & Security Services

Web Application Servers

Web Presentation Servers

Transaction Servers

Data Servers

End-to-End Systems Management

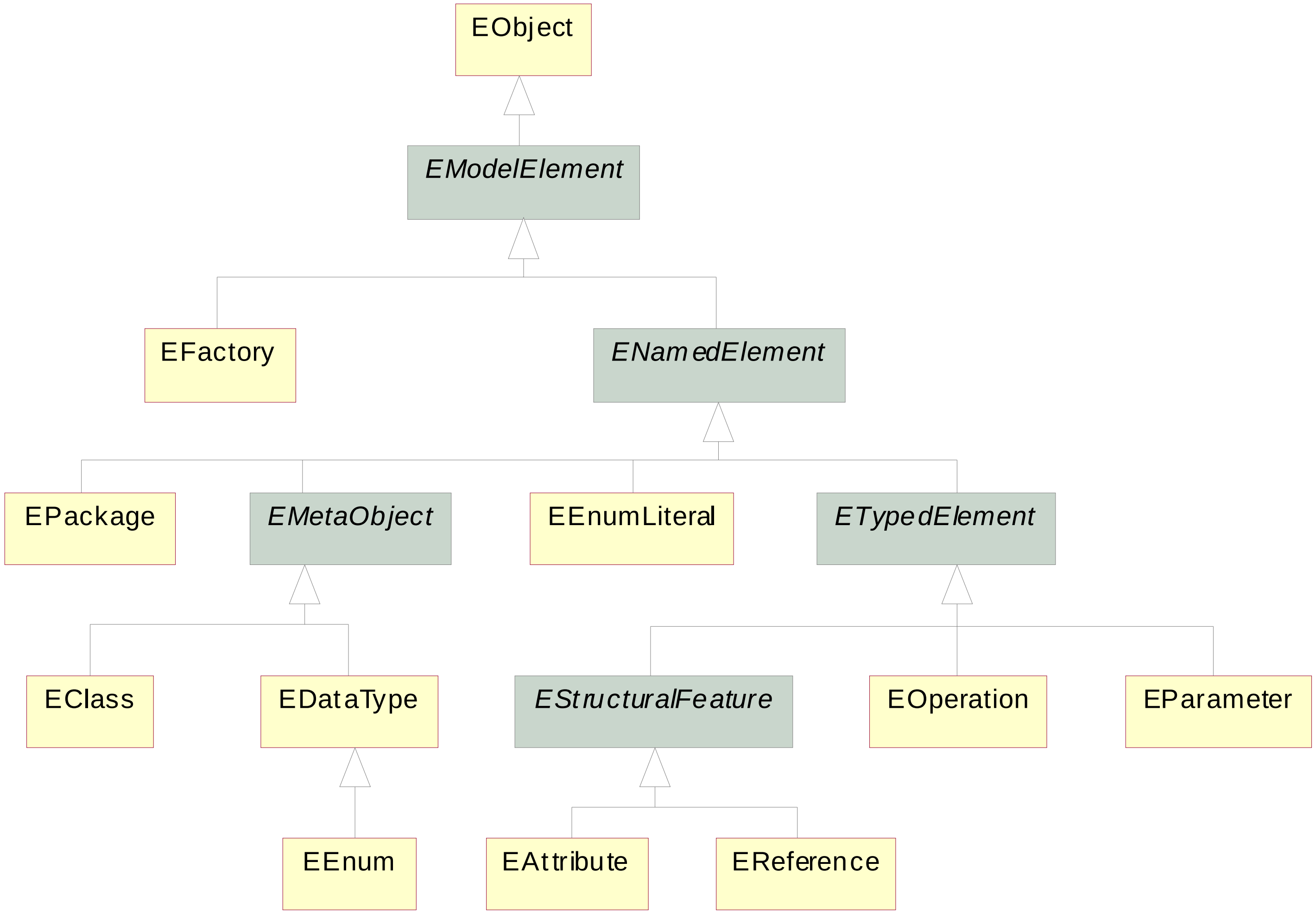
Responsive

Dynamic & Flexible

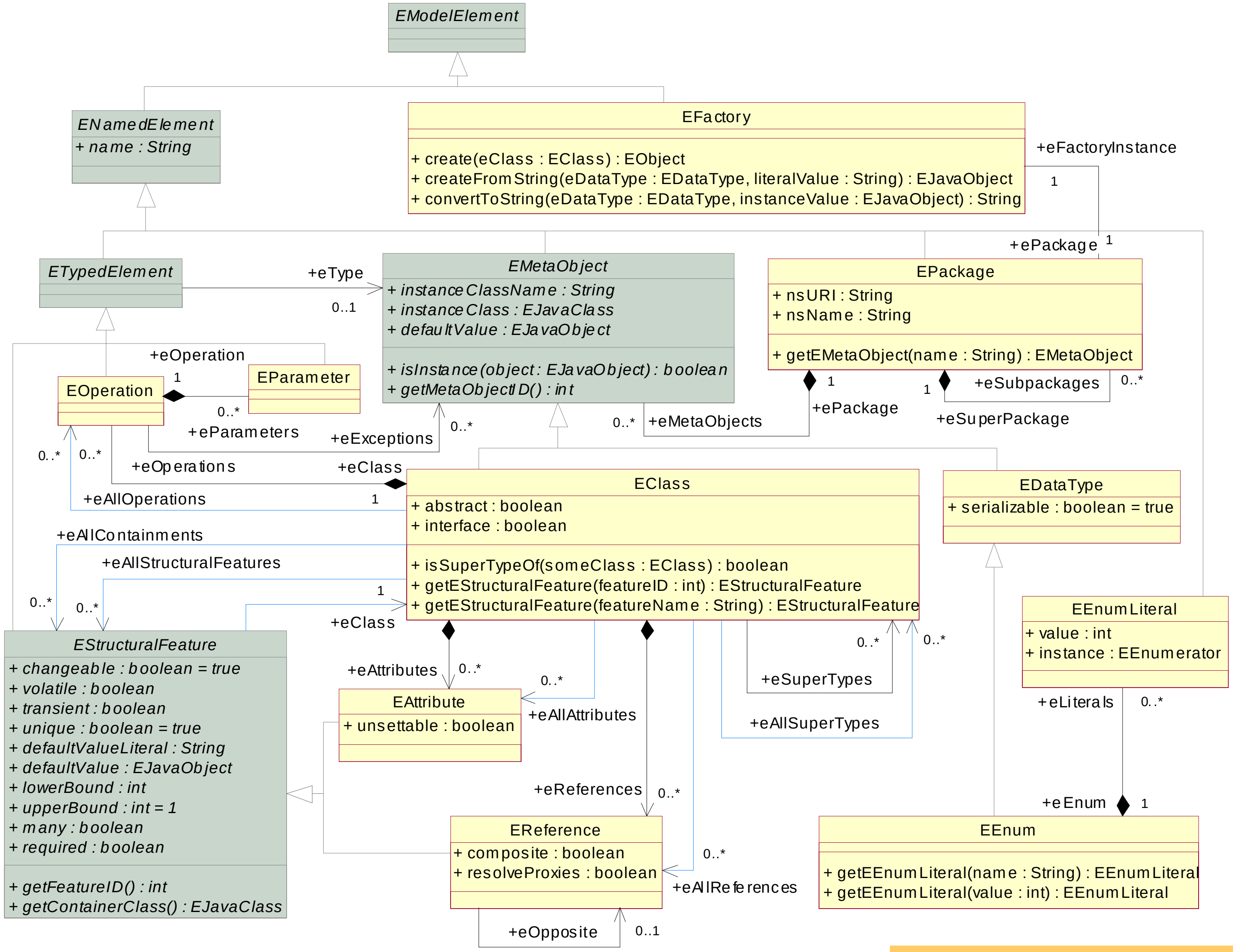
Scalable & Robust

* In R&D

# EMF Ecore Model (www.eclipse.org/emf)



# EMF::Ecore Details



*EModelElement*

*ENamedElement*
*+ name : String*

EFactory
+ create(eClass : EClass) : EObject
+ createFromString(eDataType : EDataType, literalValue : String) : EJavaObject
+ convertToString(eDataType : EDataType, instanceValue : EJavaObject) : String

*ETypedElement*

*EMetaObject*
*+ instanceClassName : String*
*+ instanceClass : EJavaClass*
*+ defaultValue : EJavaObject*
*+ isInstance(object : EJavaObject) : boolean*
*+ getMetaObjectID() : int*

EPackage
+ nsURI : String
+ nsName : String
+ getEMetaObject(name : String) : EMetaObject

EOperation

EParameter

EClass
+ abstract : boolean
+ interface : boolean
+ isSuperTypeOf(someClass : EClass) : boolean
+ getEStructuralFeature(featureID : int) : EStructuralFeature
+ getEStructuralFeature(featureName : String) : EStructuralFeature

EDataType
+ serializable : boolean = true

EEnumLiteral
+ value : int
+ instance : EEnumerator

*EStructuralFeature*
*+ changeable : boolean = true*
*+ volatile : boolean*
*+ transient : boolean*
*+ unique : boolean = true*
*+ defaultValueLiteral : String*
*+ defaultValue : EJavaObject*
*+ lowerBound : int*
*+ upperBound : int = 1*
*+ many : boolean*
*+ required : boolean*
*+ getFeatureID() : int*
*+ getContainerClass() : EJavaClass*

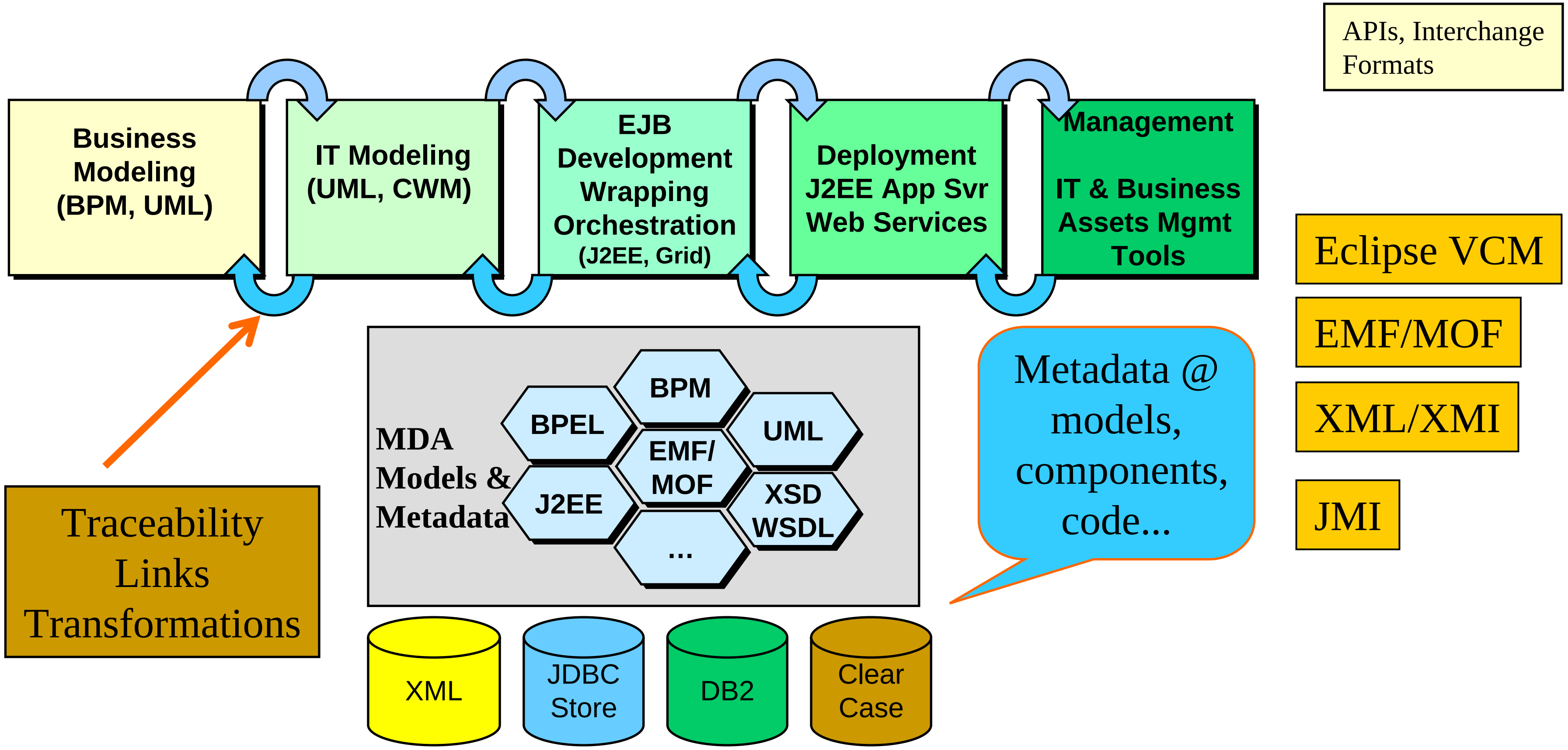
EAttribute
+ unsettable : boolean

EReference
+ composite : boolean
+ resolveProxies : boolean

EEnum
+ getEEnumLiteral(name : String) : EEnumLiteral
+ getEEnumLiteral(value : int) : EEnumLiteral

Influenced MOF2::EMOF

# Model Driven Tools Integration
# An end to end view - A peek ahead?


APIs, Interchange Formats
Business Modeling (BPM, UML)
IT Modeling (UML, CWM)
EJB Development Wrapping Orchestration (J2EE, Grid)
Deployment J2EE App Svr Web Services
Management
IT & Business Assets Mgmt Tools
Eclipse VCM
EMF/MOF
XML/XMI
JMI
Traceability Links Transformations
MDA Models & Metadata
BPM
BPEL
UML
EMF/ MOF
J2EE
XSD WSDL
...
Metadata @ models, components, code...
XML
JDBC Store
DB2
Clear Case

| | |
|---|---|
| Business Modeling | : MDA Computation Independent Model (BPM) |
| IT Modeling | : MDA Platform Independent Model (UML, CWM) |
| | MDA Platform Specific Model (J2EE… |

Model Transformations across layers

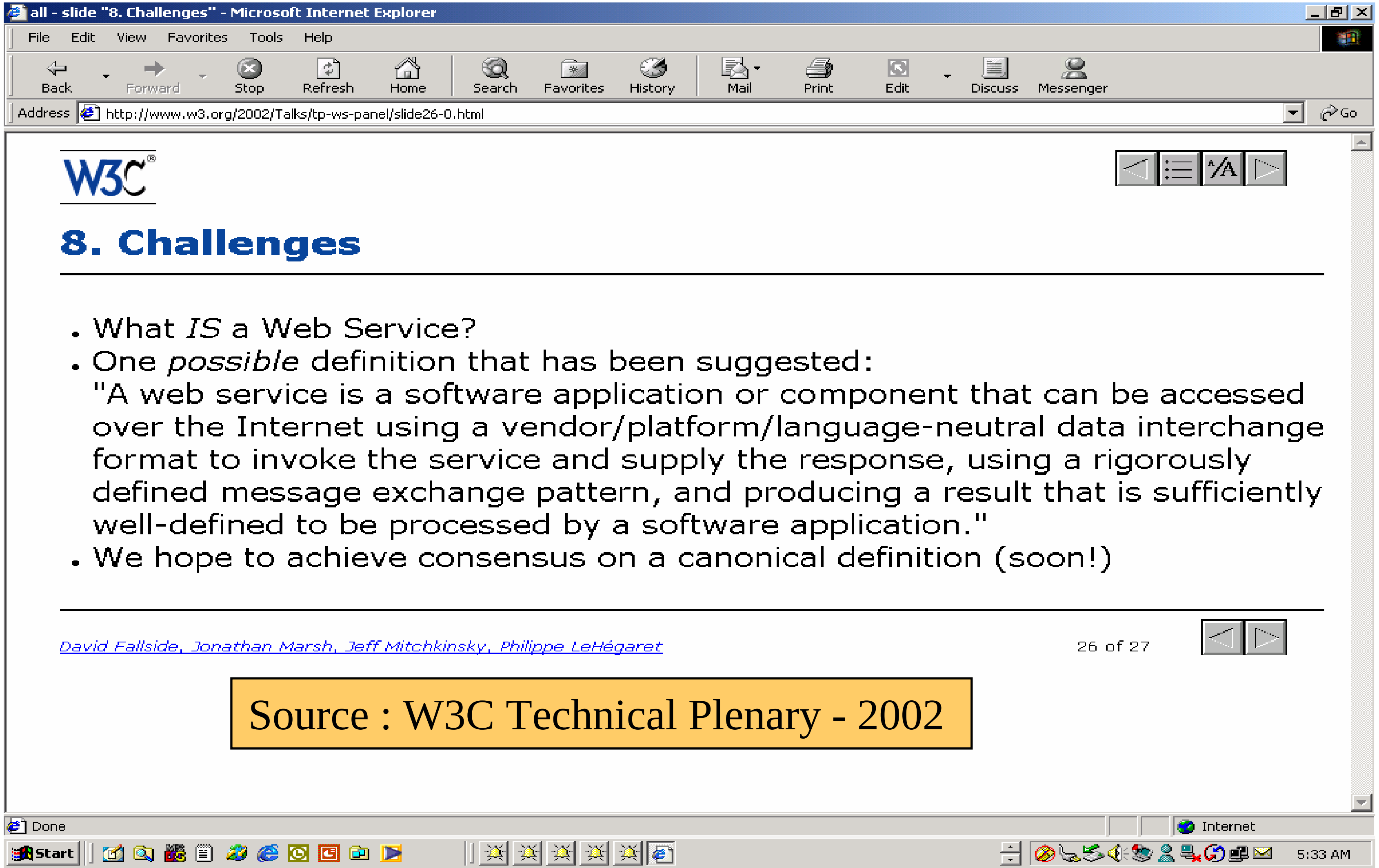
# Presentation Overview

- Introduction - The integration problem
- OMG MDA - Model Driven Architecture Overview
  - What is MDA and why is it important
  - Emerging MDA standards for Model Driven Business Integration
- Web Services Architecture Overview
  - What are Web Services and why are they important
  - Emerging Web Services Standards and Architecture
- MDA and Web Services – How do you use them together
  - Modeling Web Services
  - Advanced Modeling (Web Services Orchestration)
  - Model Driven Business Integration – A peek ahead
- Wrap-up

# What are Web Services?

- ‘Modular and reusable’ applications that can be
  - Invoked over the web (using a protocol like SOAP) for exchanging XML documents
  - Described using a service description language (like WSDL)
  - Published into a registry (like UDDI) so the service can be searched, bound and invoked
  - Orchestrated using a business process execution language (like BPEL4WS)
- Web Services do not imply component or object based development – but obviously they can be used together.
  - In fact a 20 year old CICS transaction can be wrapped as a web service
- Web Services need to be composable
  - Enable better granularity of services and solutions being delivered
  - Enable service orchestration/flow
  - Use BPEL4WS with or without WS-Transactions, WS-Security…
- Can be simple to very complex; free to expensive

# What is a Web Service?
# ‘The challenge of defining it’



all - slide "8. Challenges" - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss Messenger
Address http://www.w3.org/2002/Talks/tp-ws-panel/slide26-0.html
Go
W3C®

## 8. Challenges

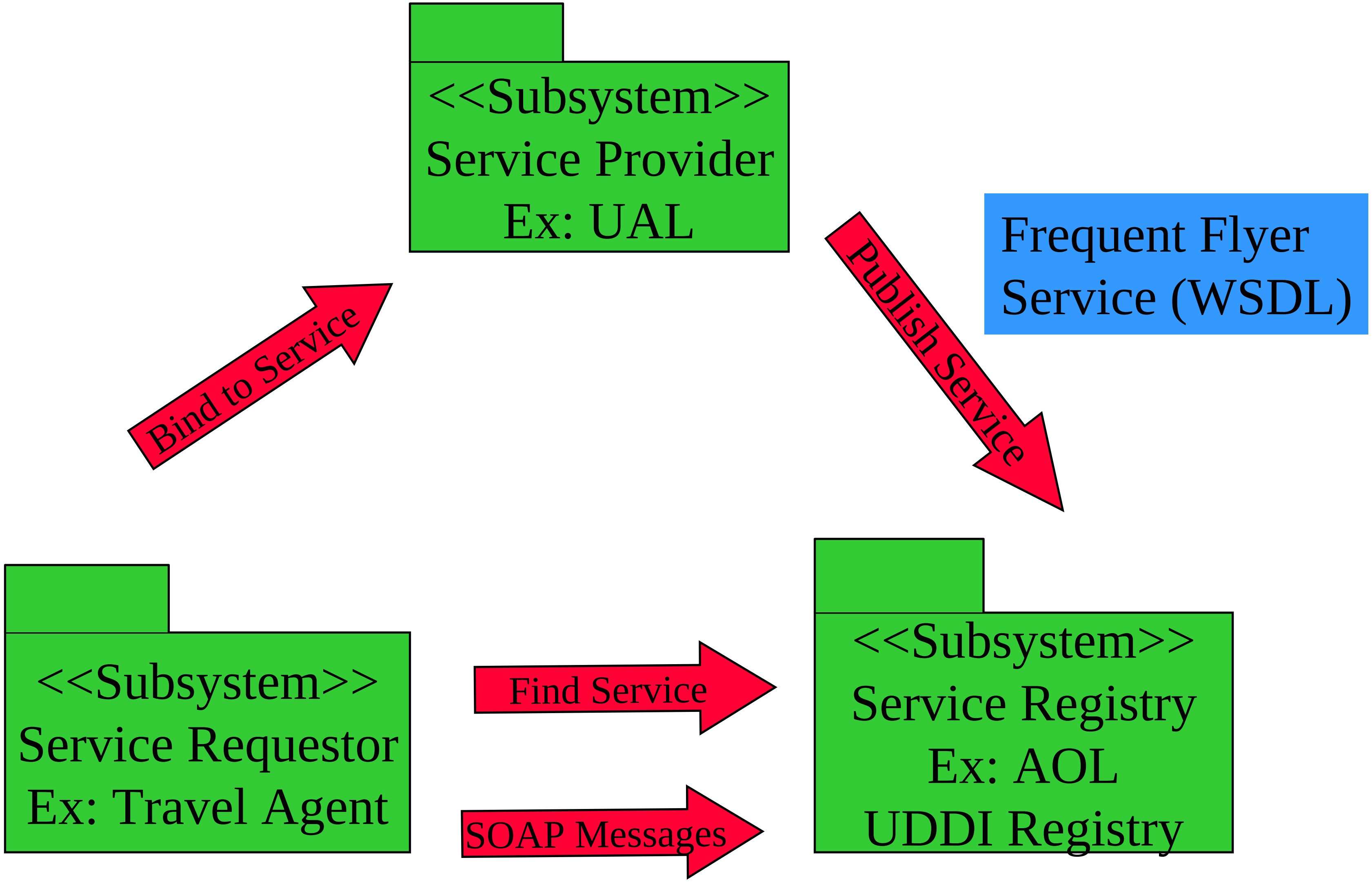
- What *IS* a Web Service?
- One *possible* definition that has been suggested:
  "A web service is a software application or component that can be accessed over the Internet using a vendor/platform/language-neutral data interchange format to invoke the service and supply the response, using a rigorously defined message exchange pattern, and producing a result that is sufficiently well-defined to be processed by a software application."
- We hope to achieve consensus on a canonical definition (soon!)

David Fallside, Jonathan Marsh, Jeff Mitchkinsky, Philippe LeHégaret

26 of 27

Source : W3C Technical Plenary - 2002

# Web Services : The basic idea



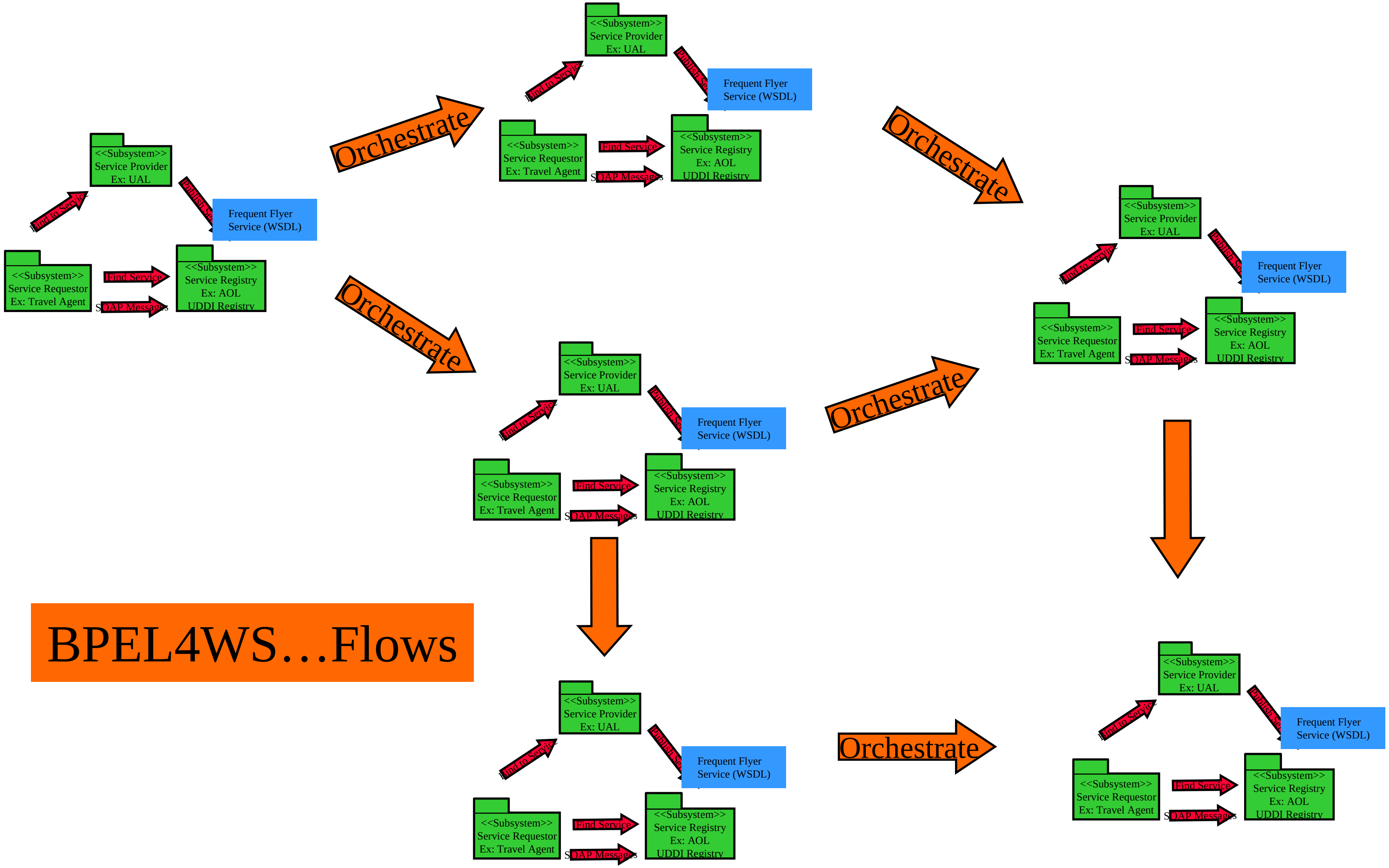
<<Subsystem>>
Service Provider
Ex: UAL

Frequent Flyer
Service (WSDL)

<<Subsystem>>
Service Requestor
Ex: Travel Agent

<<Subsystem>>
Service Registry
Ex: AOL
UDDI Registry

# Web Services - Orchestration



BPEL4WS…Flows

# *Web Services – Some Key Events (When!)*

- The early pioneers
  - XML RPC at WebMethods
  - Dave Winer (CEO Userland Software) defines SOAP
  - Microsoft jumps on the SOAP bandwagon
  - IBM endorses SOAP and then the flood follows
- The UDDI (Universal Description, Discovery and Integration) efforts by Microsoft, IBM and Ariba gave another strong push
- Soon WSDL (Web Services Description Language) joined the party
- SOAP (XML/HTTP), UDDI and WSDL form the core formats and protocols of any Web Services Architecture today
- All major vendors, middleware platforms and architectures have embraced/endorsed these technologies
  - Platforms : WebSphere, WebLogic, .Net Servers
- Now we enter the second phase : Orchestration/Flows, Security, Transaction standardization

# *Web Services*
# *The Hype*

- Middleware ‘nirvana’ that ‘solves’ the integration problem
- Everyone is using it
- Works well
- Makes CORBA, J2EE, COM+ obsolete
- Brand new innovation
- Every developer better learn it ‘or else’…
- There is a huge market
- Standards are simple and usable and complete

# Web Services
# The Reality

- A very important step in the right direction towards lower cost ubiquitous middleware
  - Key technologies – XML, HTTP, SOAP and WSDL are freely available on most platforms
  - Key enabler for ‘Service Oriented Architecture’
- Most vendors and some customers are incorporating web services into their architectural frameworks and plans
  - Few production quality implementations, but growing fast
- Standards and tools are beginning to mature
  - More work – especially Transactions, Security well underway
- Being integrated with J2EE, .Net and CORBA frameworks
- Products : WebSphere, WebLogic, Microsoft Biztalk server…
- Market potential : Promising but will need wide spread interoperable implementations (WS-I role)

# Web Services
# Why are they important?

- Promise of delivering ‘information services’ to anyone, any program, anywhere at anytime
  - Some services free, others not
- All the major vendors have (at least for now!) called a truce and embraced web services
- Better chance for improved interoperability across technology silos
- Development tools & Middleware from Microsoft, IBM, BEA… have rapidly embraced web services
- Open standards for ‘Service Oriented Architectures’
  - More pragmatic infrastructure for business model and process integration standards
- W3C, OASIS and WS-I leading the definition of additional standards

# *Web Services : The Stack - 1*

- Network
  - TCP/IP, HTTP, FTP, SMTP..
- XML Messaging
  - Messages conform to well defined XML Schemas
  - SOAP is the preferred messaging infrastructure
    - Eventually W3C XMLP will supplant SOAP
- Service Description
  - These are essentially interface definitions ( we called these IDLs before!)
  - WSDL is the interface definition language for web services
  - Implementation (binding) information is also captured by WSDL

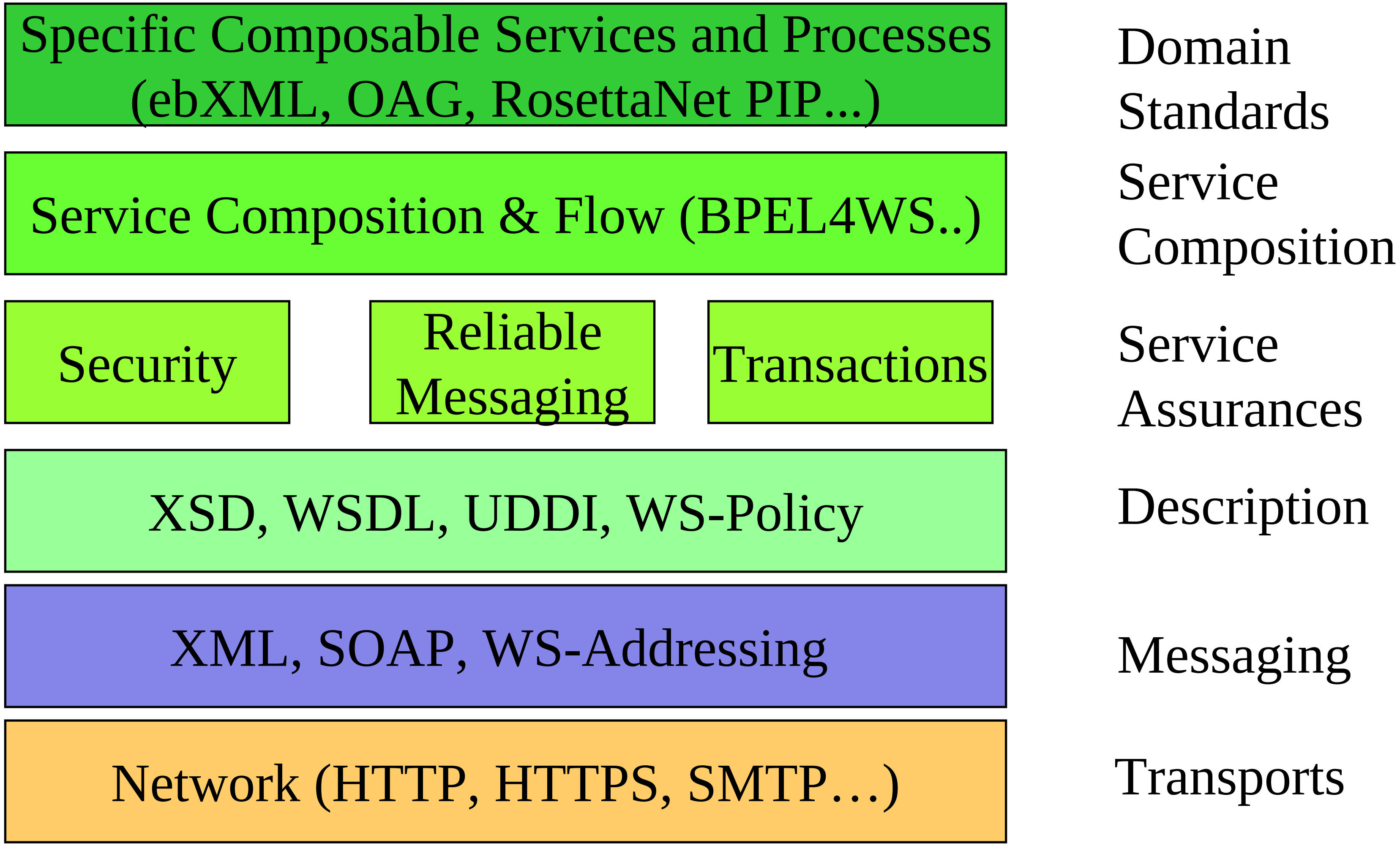
# *Web Services : The Stack - 2*

- Service Publication and Discovery
  - Publish the service definition and implementation metadata
  - Capture additional business oriented service description information (D&B, Phone, URL…)
  - This metadata is published into a UDDI Registry
    - Registry is searchable
  - Service discovery happens dynamically at run time
    - Similar to CORBA Interface Repository and DII; Java Reflection + JNDI …
  - Services are bound statically at design/development time
- Service sequencing and flows build on XSD and WSDL
  - BPEL4WS – Unification of IBM WSFL and Microsoft XLANG
  - Endorsed by BEA, SAP and many more vendors

# *Web Services : The Stack - 3*

- Service Flows
  - Individual services can be steps in a flow (BPEL4WS)
  - Flows can be graphically modeled
    - Various notations including UML Activity Diagrams have been used
    - No dominant industry standard for visual notations
    - OMG Business Process Definition Metamodel RFP to address this issue
  - Service composition needed for multi party service integration
- This area does not have as much consensus as lower level protocols
- Standardization track still being resolved

# *Web Services : The ‘Standards Stack’ Evolves in W3C, OASIS and WS-I*



Specific Composable Services and Processes (ebXML, OAG, RosettaNet PIP...) — Domain Standards

Service Composition & Flow (BPEL4WS..) — Service Composition

Security | Reliable Messaging | Transactions — Service Assurances

XSD, WSDL, UDDI, WS-Policy — Description

XML, SOAP, WS-Addressing — Messaging

Network (HTTP, HTTPS, SMTP…) — Transports

Not intended to be viewed as a ‘software stack’

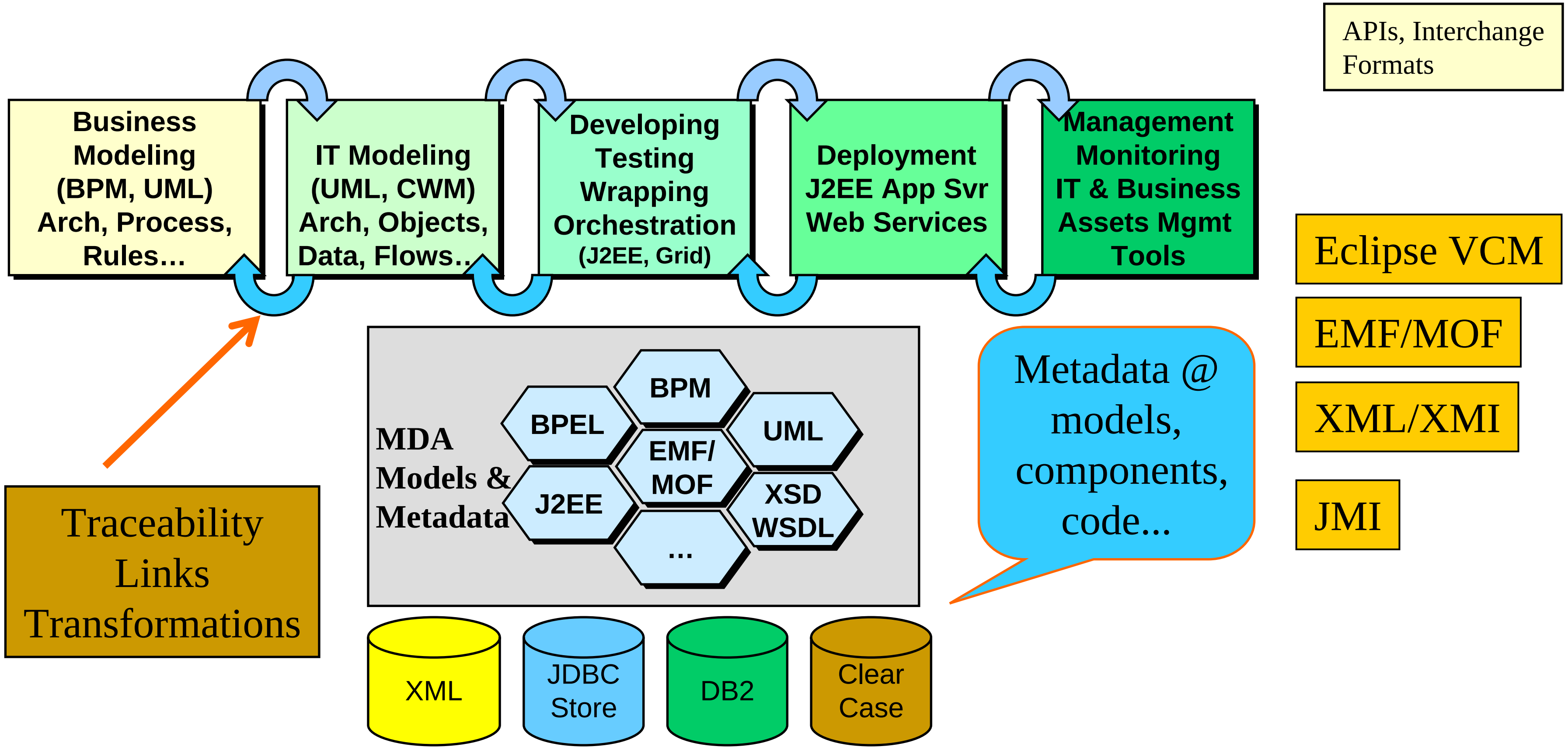
# Summary Web Services Standards

- W3C
  - XMLP (SOAP+), WSDL,
  - W3C Architecture Working Group
- UDDI Consortium → UDDI 3.0 being progressed in OASIS
- OASIS
  - UDDI, BPEL4WS, Many WS*
- SUN Java Community Process
  - Integrate J2EE and Web Services Technologies
  - JAXP, JAX RPC, JAXB, JAXR, JAXM (J2EE 1.4 target – JSR 109)
    - http://jcp.org/jsr/detail/109.jsp
- OMG
  - Modeling and metadata management for Web Services
  - UML2 Activity Graphs, UML2 Infrastructure, MOF2
  - Business Process Definition Metamodel RFP (UML2 -- BPEL4WS)
- Web Services Interoperability Organization (WS-IO)
- Web Services Security (WS-Security, WS-Trust)
  - IBM, Microsoft, Verisign joint proposal submitted to OASIS
  - www.ibm.com/developerworks/library/ws-secure

# Presentation Overview

- Introduction - The integration problem
- OMG MDA - Model Driven Architecture Overview
  - What is MDA and why is it important
  - Emerging MDA standards for Model Driven Business Integration
- Web Services Architecture Overview
  - What are Web Services and why are they important
  - Emerging Web Services Standards and Architecture
- MDA and Web Services – How do you use them together
  - Modeling Web Services
  - Advanced Modeling (Web Services Orchestration)
  - Model Driven Business Integration – A peek ahead
- Wrap-up

# Model Driven Tools Integration
# An end to end view - A peek ahead?



APIs, Interchange Formats
Business Modeling (BPM, UML) Arch, Process, Rules…
IT Modeling (UML, CWM) Arch, Objects, Data, Flows…
Developing Testing Wrapping Orchestration (J2EE, Grid)
Deployment J2EE App Svr Web Services
Management Monitoring IT & Business Assets Mgmt Tools
Eclipse VCM
EMF/MOF
XML/XMI
JMI
MDA Models & Metadata
BPM
BPEL
UML
EMF/ MOF
J2EE
XSD WSDL
…
Metadata @ models, components, code...
Traceability Links Transformations
XML
JDBC Store
DB2
Clear Case

Business Modeling : MDA Computation Independent Model (BPM)

IT Modeling : MDA Platform Independent Model (UML, CWM)

MDA Platform Specific Model (J2EE…

Model Transformations across layers

# Modeling & Design for Web Services

- UML can be used to model Web Services
  - Web Service Operations (WSDL Operations) are UML Operations
  - Groups of Web Service structures Requests (WSDL PortTypes) as UML Interfaces or Classes
  - XML Schemas using UML Class Diagrams
  - Web Services data structures using UML Class Diagrams
  - Web Services flow (E.g.:BPEL4WS) using UML Activity Diagrams
- MOF and XMI can be used to define standard transformations from models to XML schemas, DTDs, Documents
  - www.eclipse.org/emf, xsd, uml
- But standard metamodels, transformations, UML profiles etc, would be useful
  - http://www.omg.org (MOF, XMI, UML and CWM specs)
  - XMI - Mappings from MOF/UML to XML DTD, XML Schema
  - Modeling XML applications with UML by David Carlson
    - http://www.xmlmodeling.com
  - UML to BPEL4WS mappings

# *Service Oriented Architecture & Component Based Development*

- Secure, Reliable, Transacted Web Services
  - http://www-106.ibm.com/developerworks/webservices/library/ws-securtrans/
- Using Service Oriented Architecture & Component Based Development to build Web Services Applications
  - http://www-106.ibm.com/developerworks/rational/library/510.html
- Modeling XML Applications with UML
  - http://www.xmlmodeling.com/DesktopDefault.aspx

# Design Patterns/Concerns for Web Services Applications

- Interface based Design (eg: separate interface & implementation concerns)
- Discoverable Services (eg: reflection pattern)
- Coarse-grained interactions (eg: value object pattern)
- Loosely Coupled (eg: message based connection)
- Single instance (eg: Stateless sessions in a running instance VS instantiating individual components)
- Asynchronous (eg: nature of the web, latency… often drives this)

# SOA Design Terminology

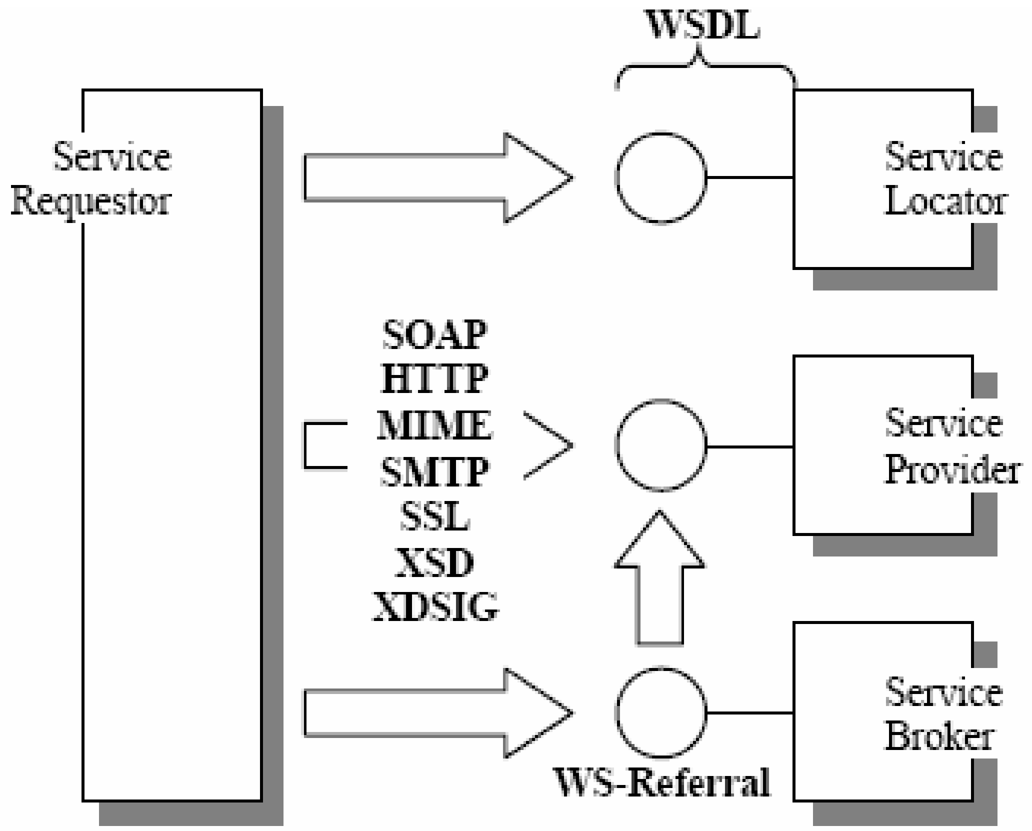
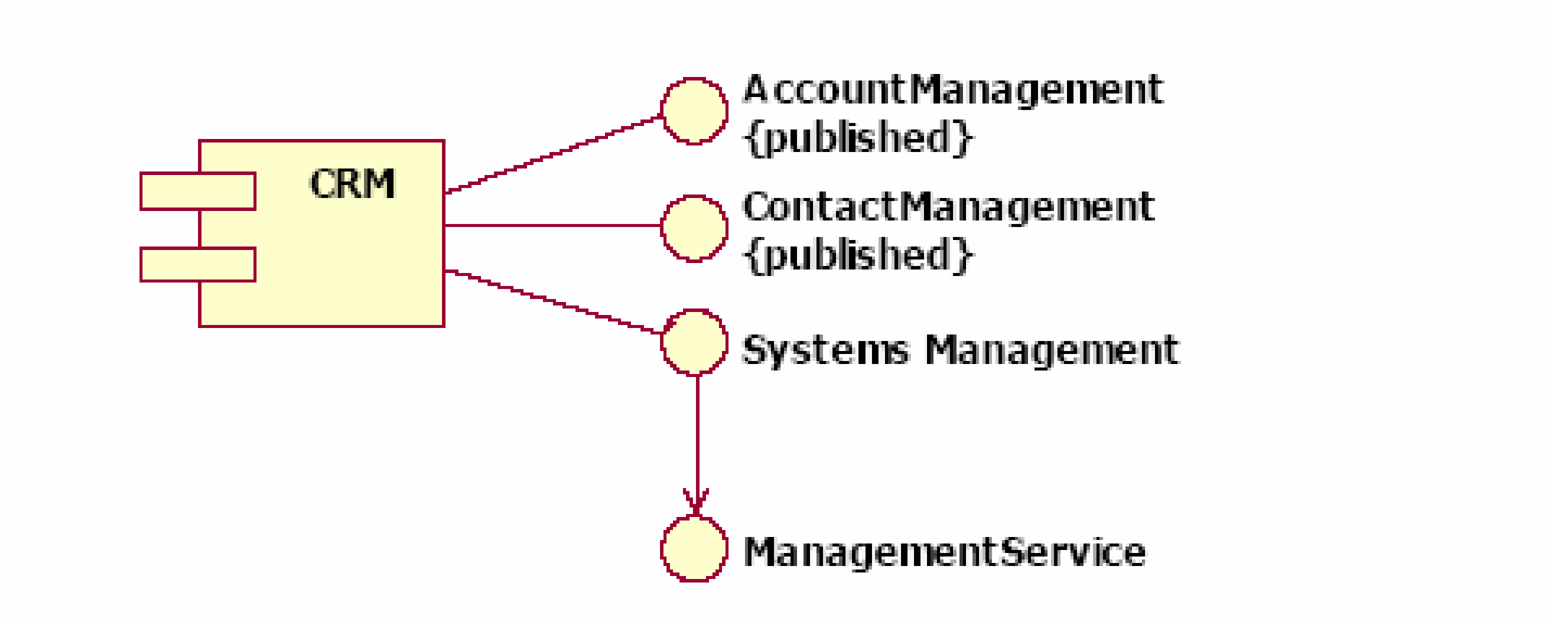


Figure 9 – XML Web service standards

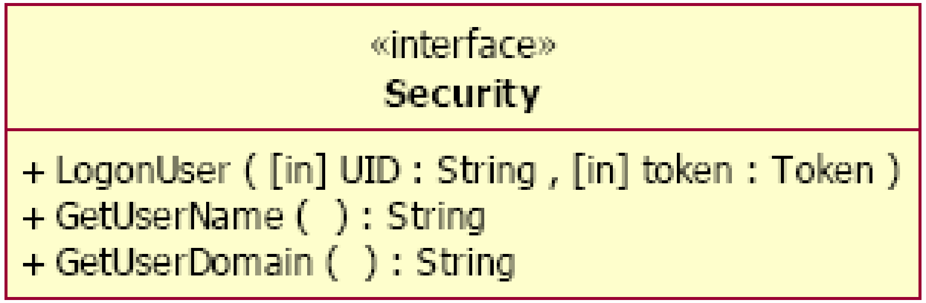
Courtesy : Brown et al

# *Services modeled as UML Components*



Courtesy : Brown et al

# *Modeling a Service Interface (UML Interface)*



«interface»
**Security**

+ LogonUser ( [in] UID : String , [in] token : Token )
+ GetUserName (  ) : String
+ GetUserDomain (  ) : String

**Figure 3 – Interface in UML**

# Modeling Service Behavior Semantics (Ex: UML State Machine)

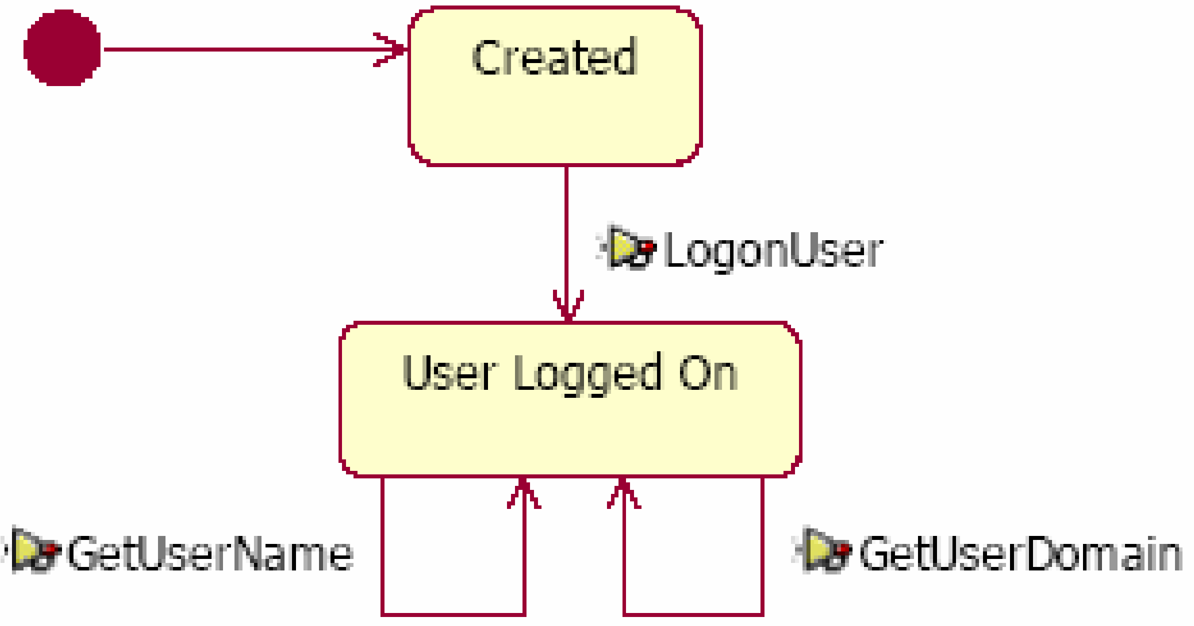


**Figure 4 – Interface behavior**

# Application Layers CBD vs SOA

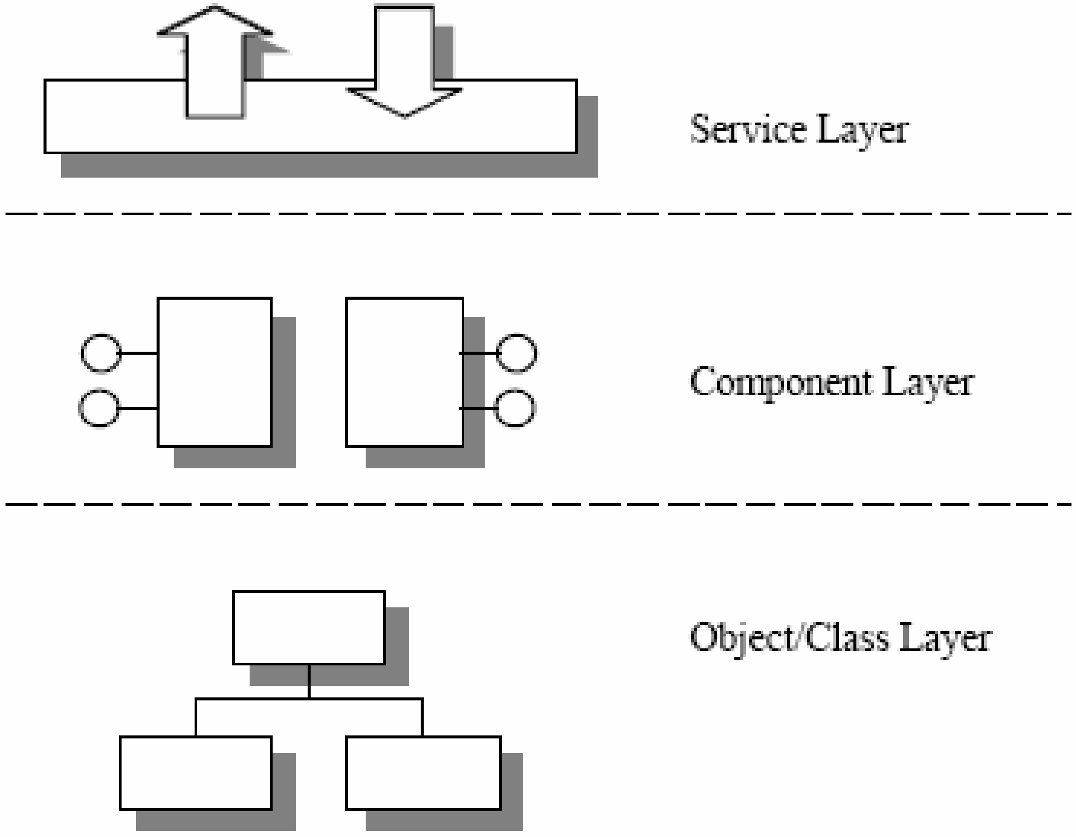
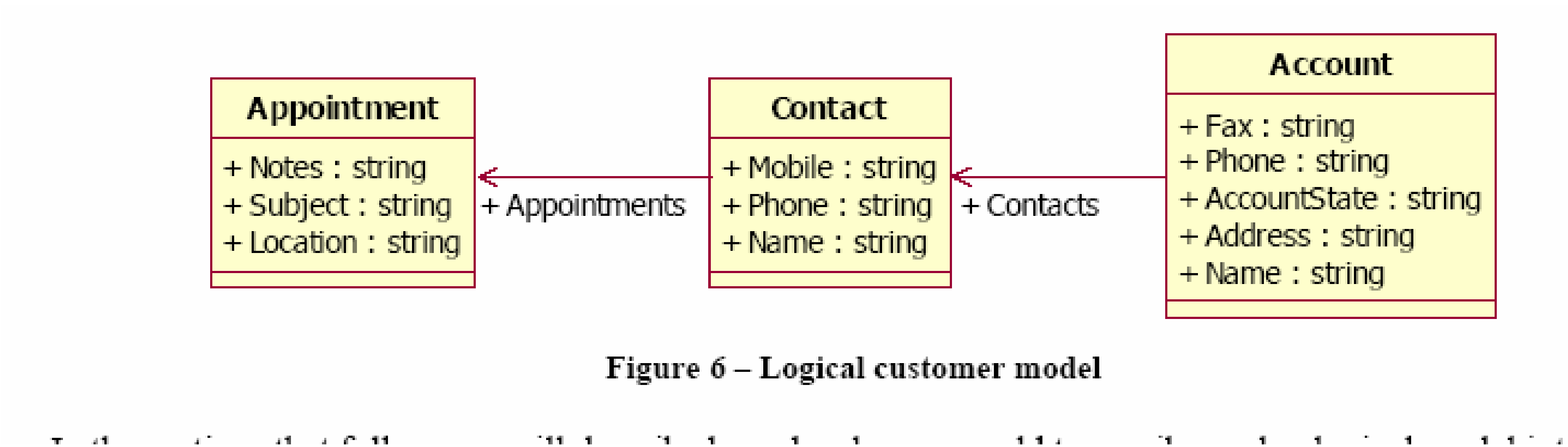


**Figure 5 – Application implementation layers**

# Logical Modeling of Services ‘Entities’/’Classes’



Appointment

+ Notes : string
+ Subject : string
+ Location : string

+ Appointments

Contact

+ Mobile : string
+ Phone : string
+ Name : string

+ Contacts

Account

+ Fax : string
+ Phone : string
+ AccountState : string
+ Address : string
+ Name : string

**Figure 6 – Logical customer model**

# Component Based Design
# UML Components, Classes & Interfaces

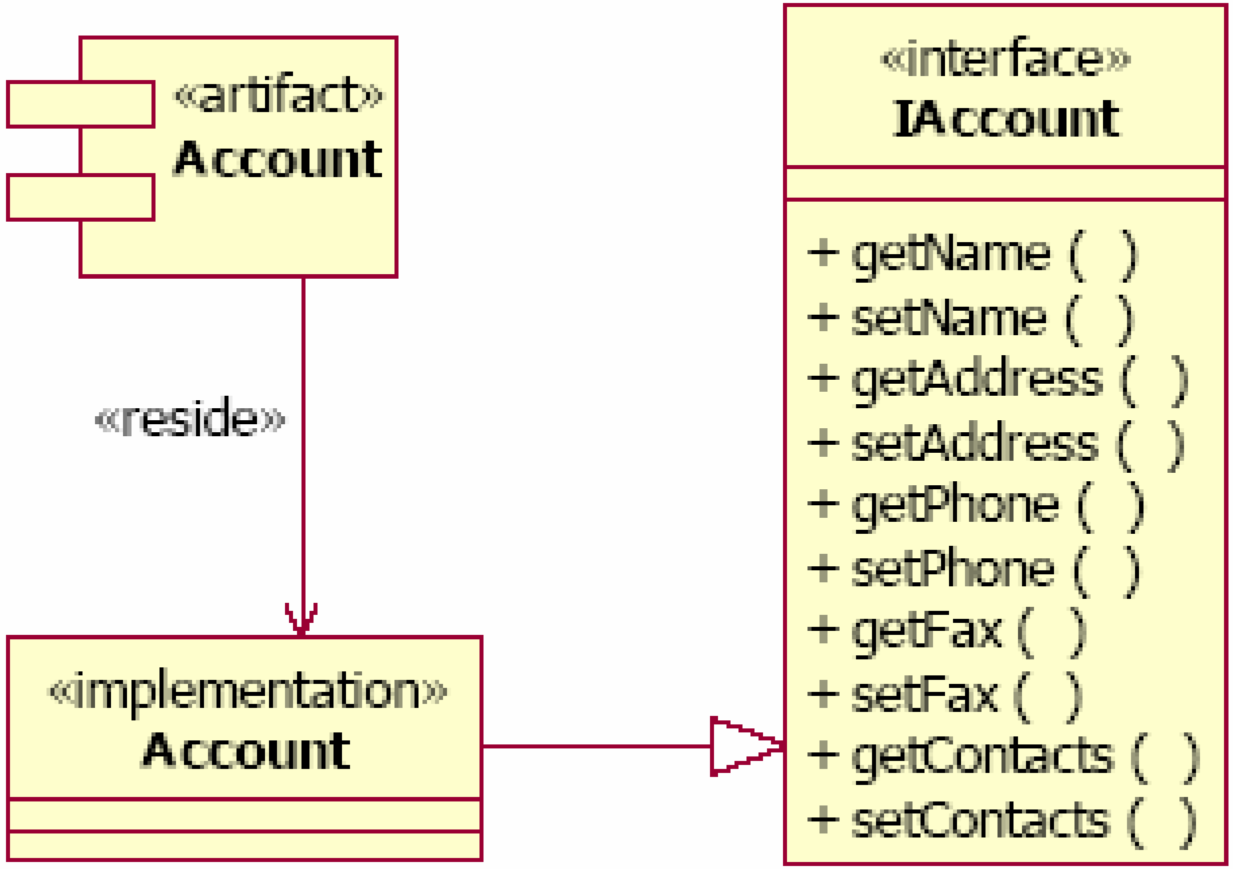


Figure 7 – Generic component diagram

# Service Oriented Design

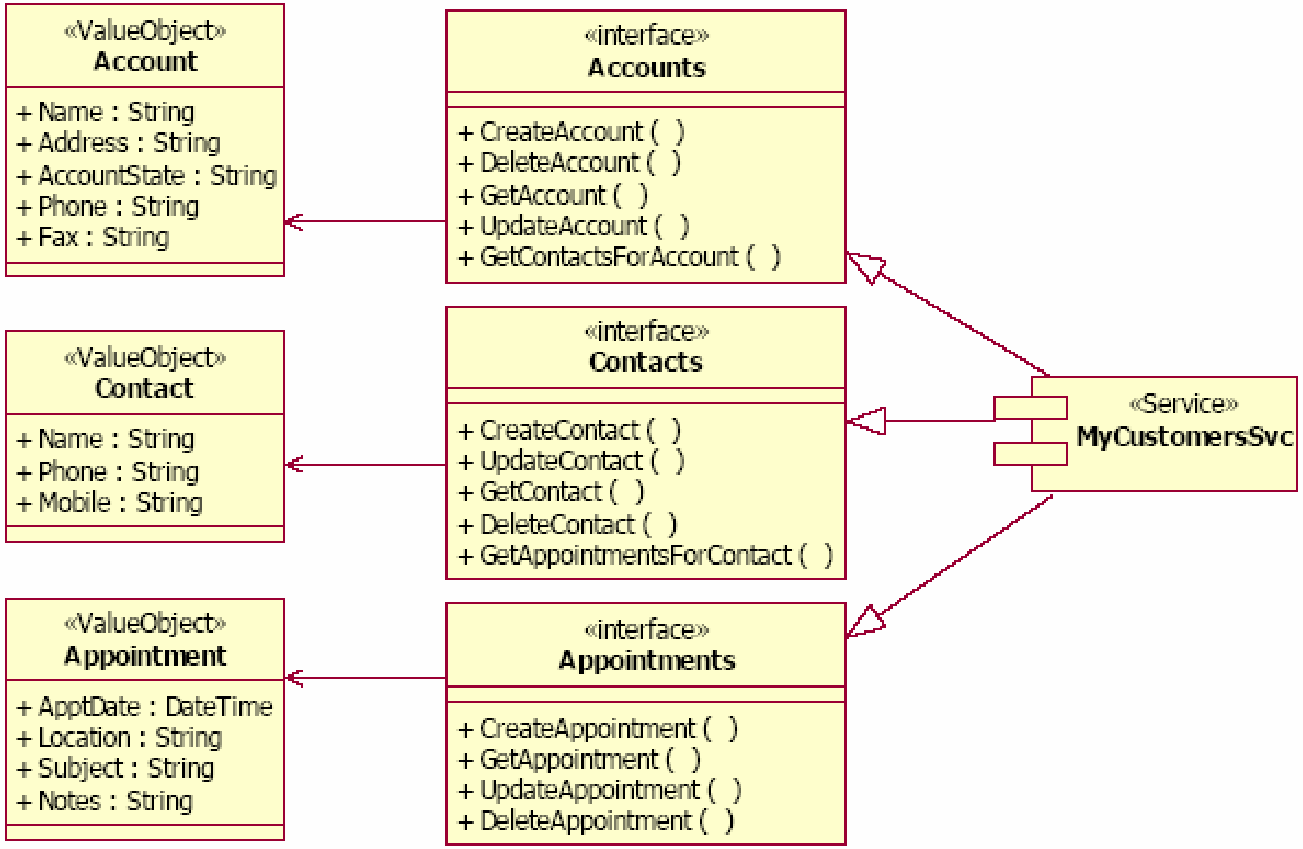


**Figure 8 – Generic service-oriented design**

# Performance & Reliability
# Synch VS Asynch

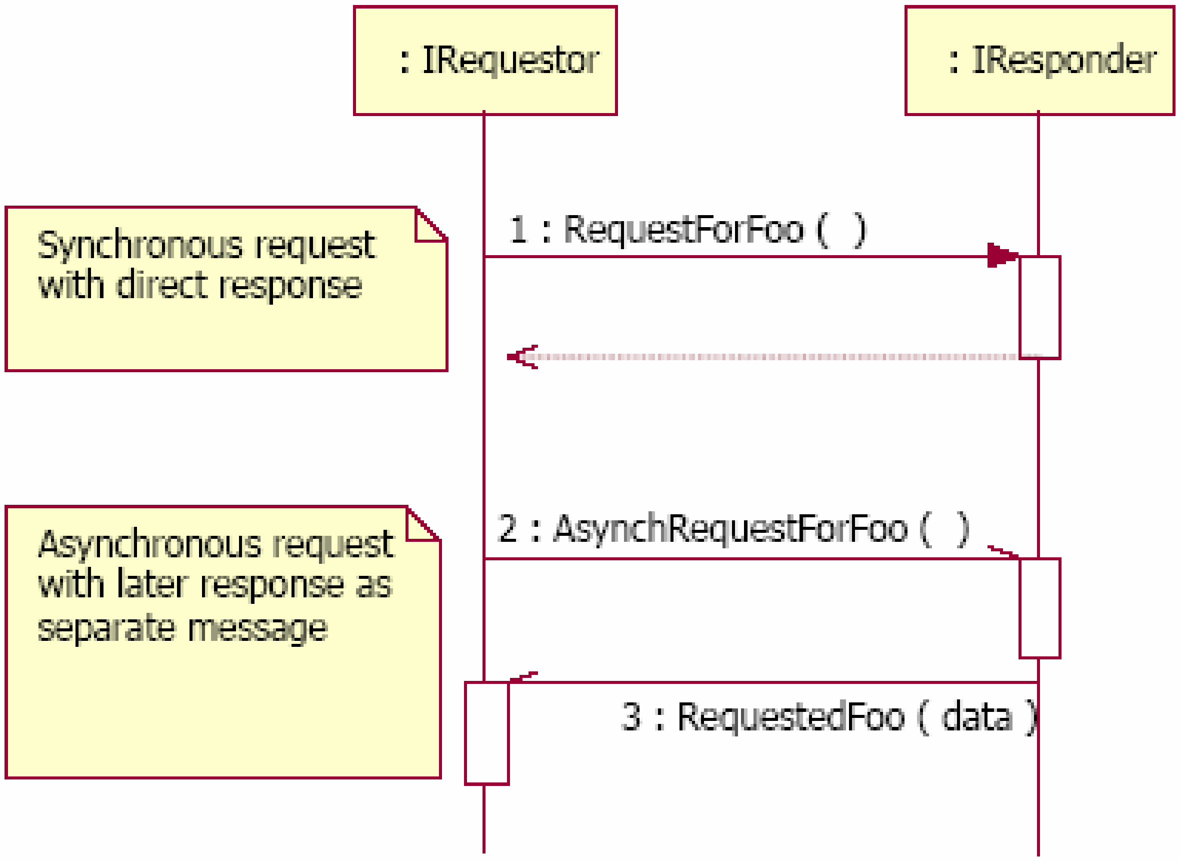

: IRequestor
: IResponder
Synchronous request with direct response
1 : RequestForFoo ( )
Asynchronous request with later response as separate message
2 : AsynchRequestForFoo ( )
3 : RequestedFoo ( data )

**Figure 11 – Synchronous vs. asynchronous**

# Performance & Reliability
# Message Queues

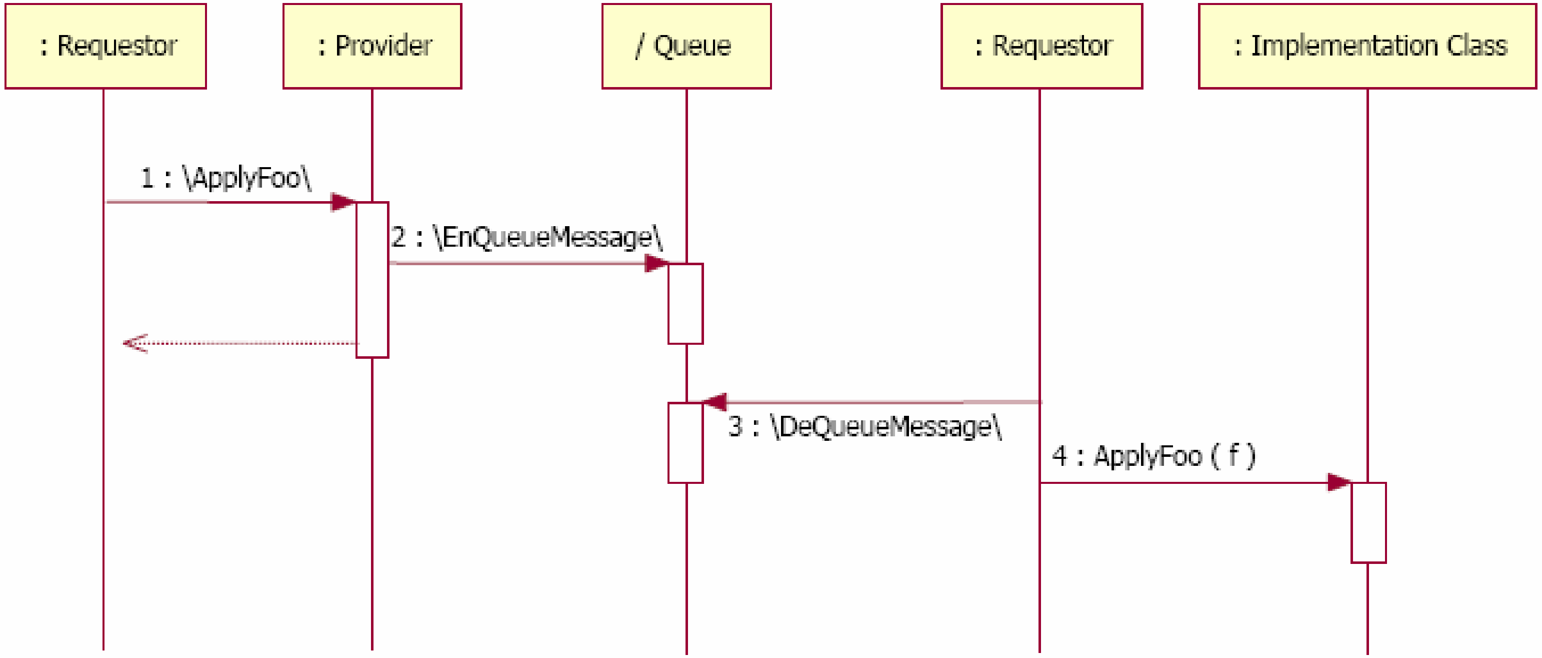


Figure 12 – Queued implementation

# Designing for Web Services Latency 'Inheritance VS Aggregation'

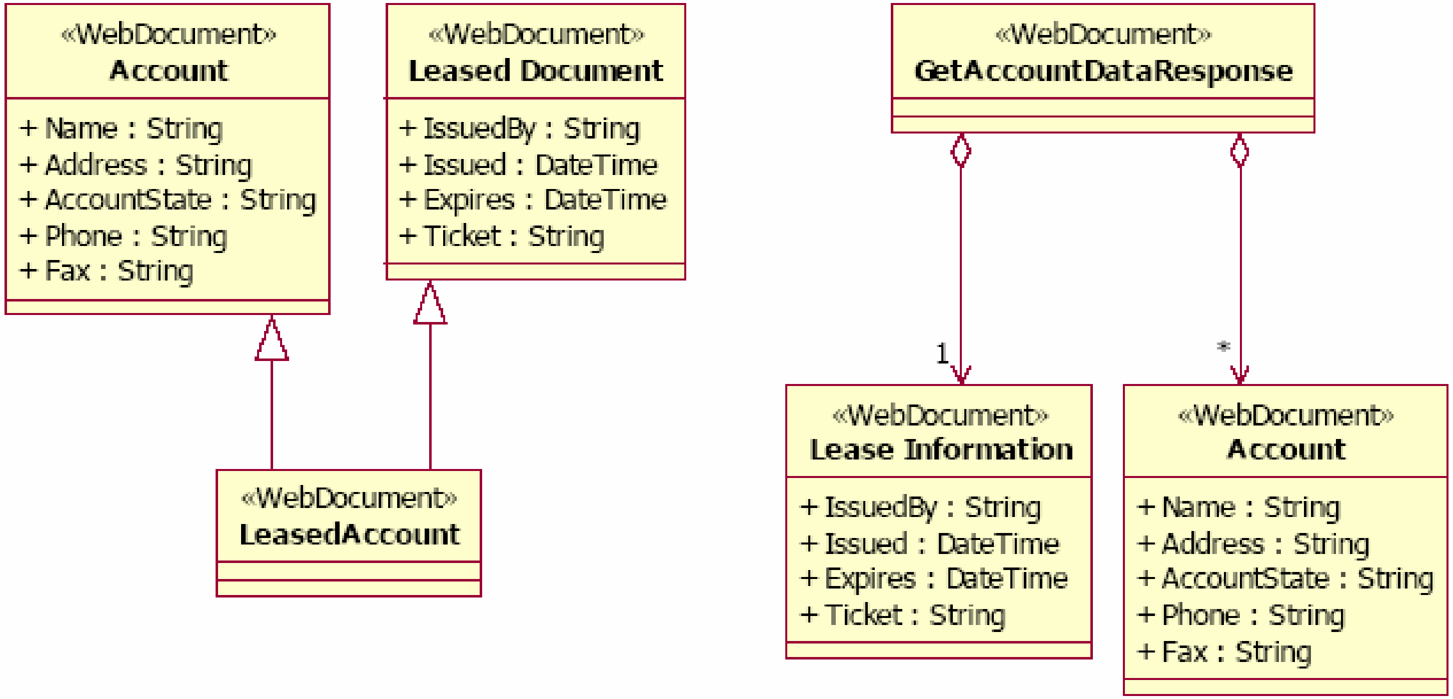


Figure 13 – Two implementations of information leasing

# *Service Oriented Design*

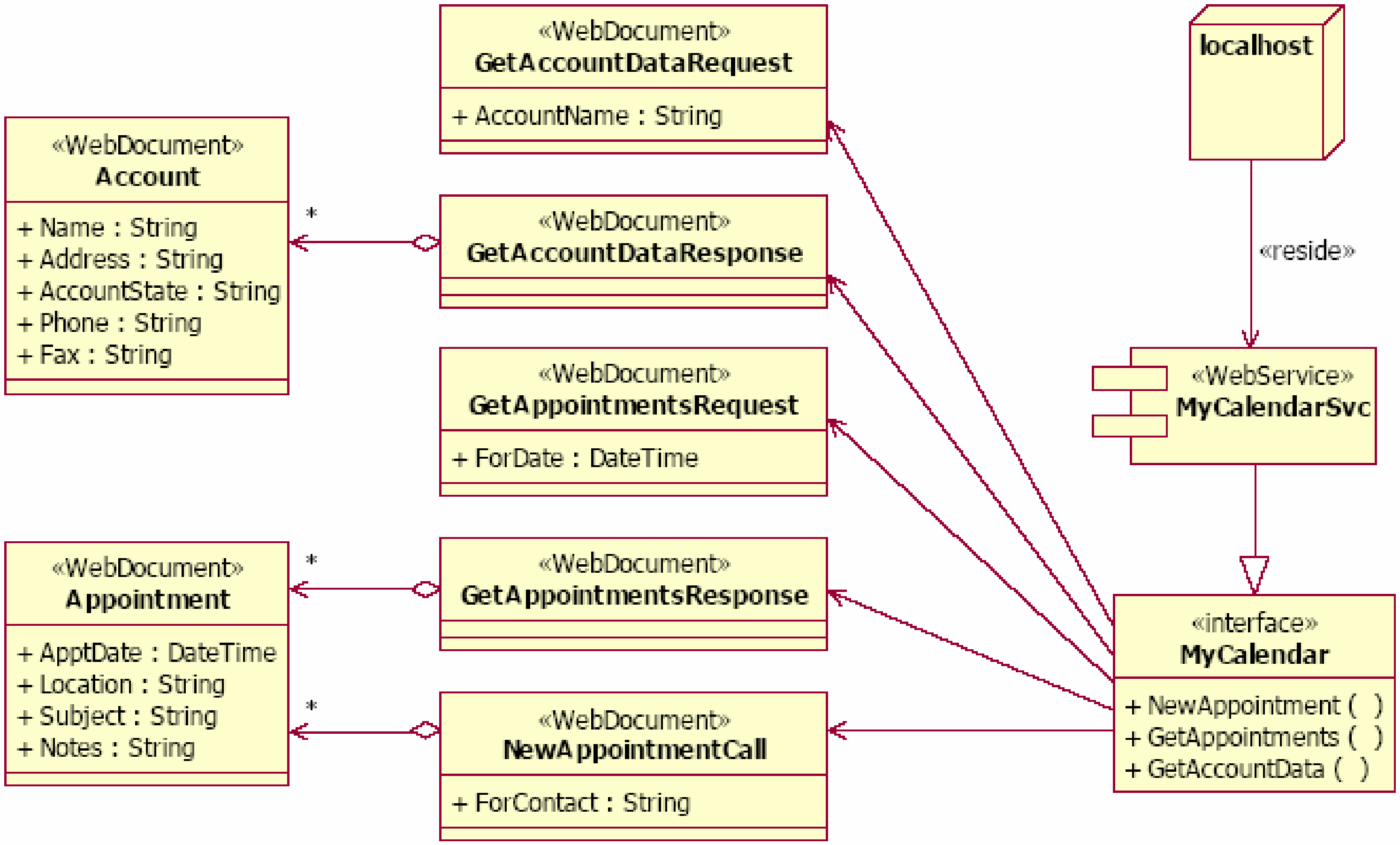
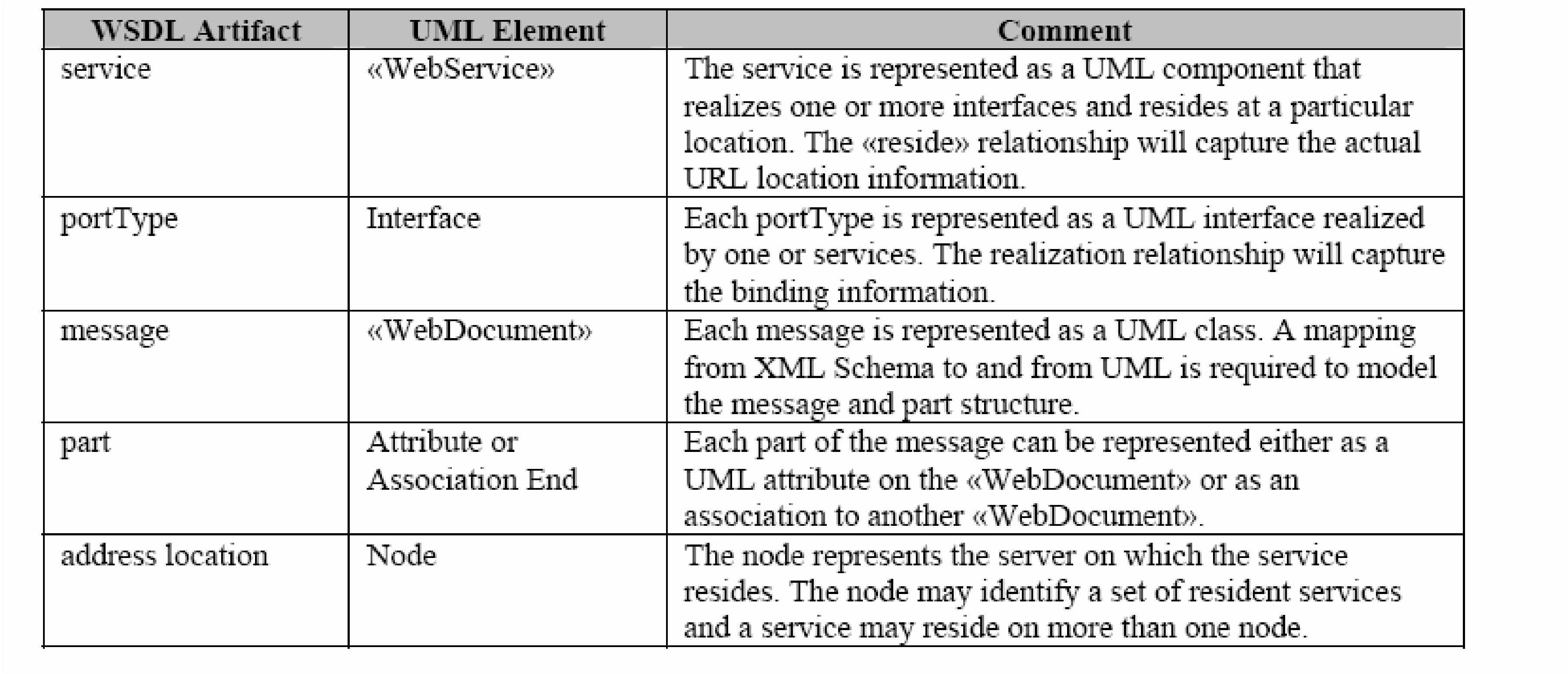


**Figure 14 – XML Web service design**

# Simple UML Web Services Profile



| WSDL Artifact | UML Element | Comment |
|---|---|---|
| service | «WebService» | The service is represented as a UML component that realizes one or more interfaces and resides at a particular location. The «reside» relationship will capture the actual URL location information. |
| portType | Interface | Each portType is represented as a UML interface realized by one or services. The realization relationship will capture the binding information. |
| message | «WebDocument» | Each message is represented as a UML class. A mapping from XML Schema to and from UML is required to model the message and part structure. |
| part | Attribute or Association End | Each part of the message can be represented either as a UML attribute on the «WebDocument» or as an association to another «WebDocument». |
| address location | Node | The node represents the server on which the service resides. The node may identify a set of resident services and a service may reside on more than one node. |

# Use of Patterns for SOA

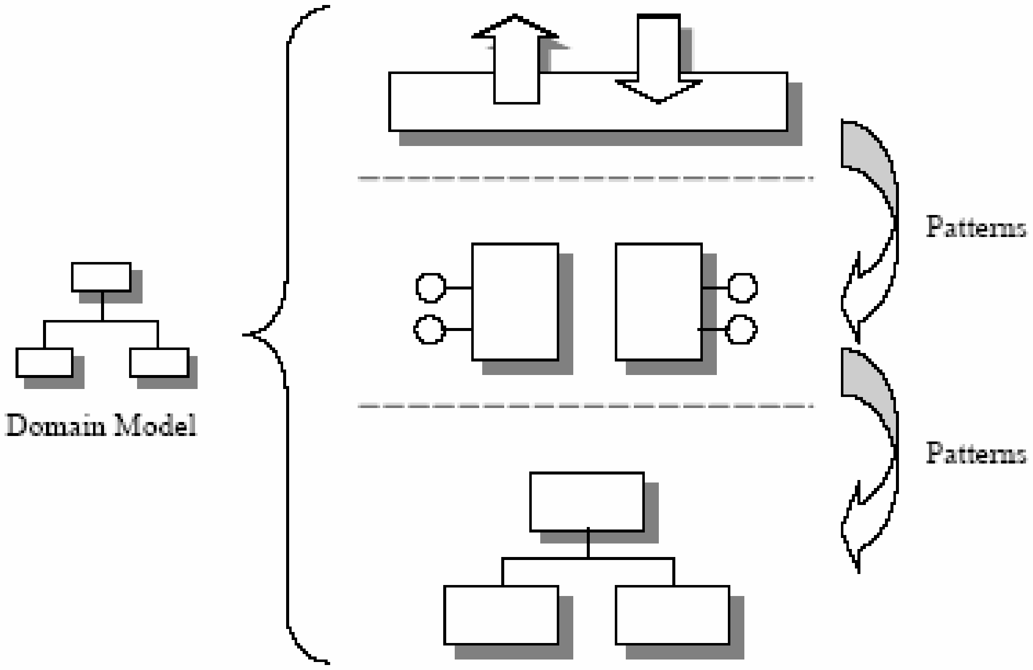
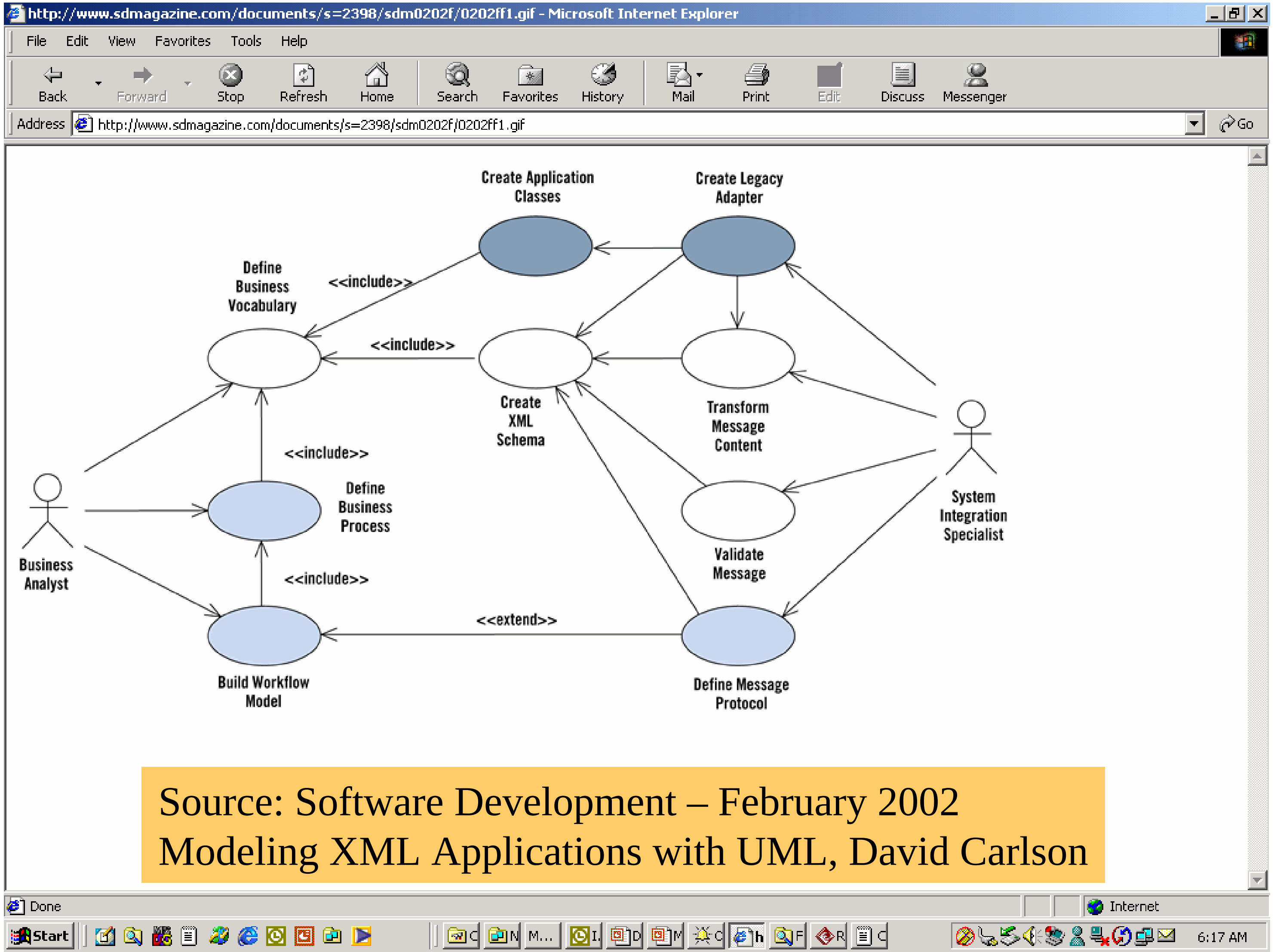


Figure 15 – Service implementation approach

# Business Process Definition Metamodel RFP

- Issued January 2003
- Goal is to define a platform independent model that unifies the modeling concepts in emerging flow/choreography standards
- Proof of concept mappings to emerging choreography standards (BPEL4WS, XPDL…)
- Basis for providing alternate visualizations as well as textual/machine readable syntax
- Link to rest of application development/integration cycle
- 3 proposals including one from IBM, BEA, Unisys, Adaptive…

# E-Business Integration using XML

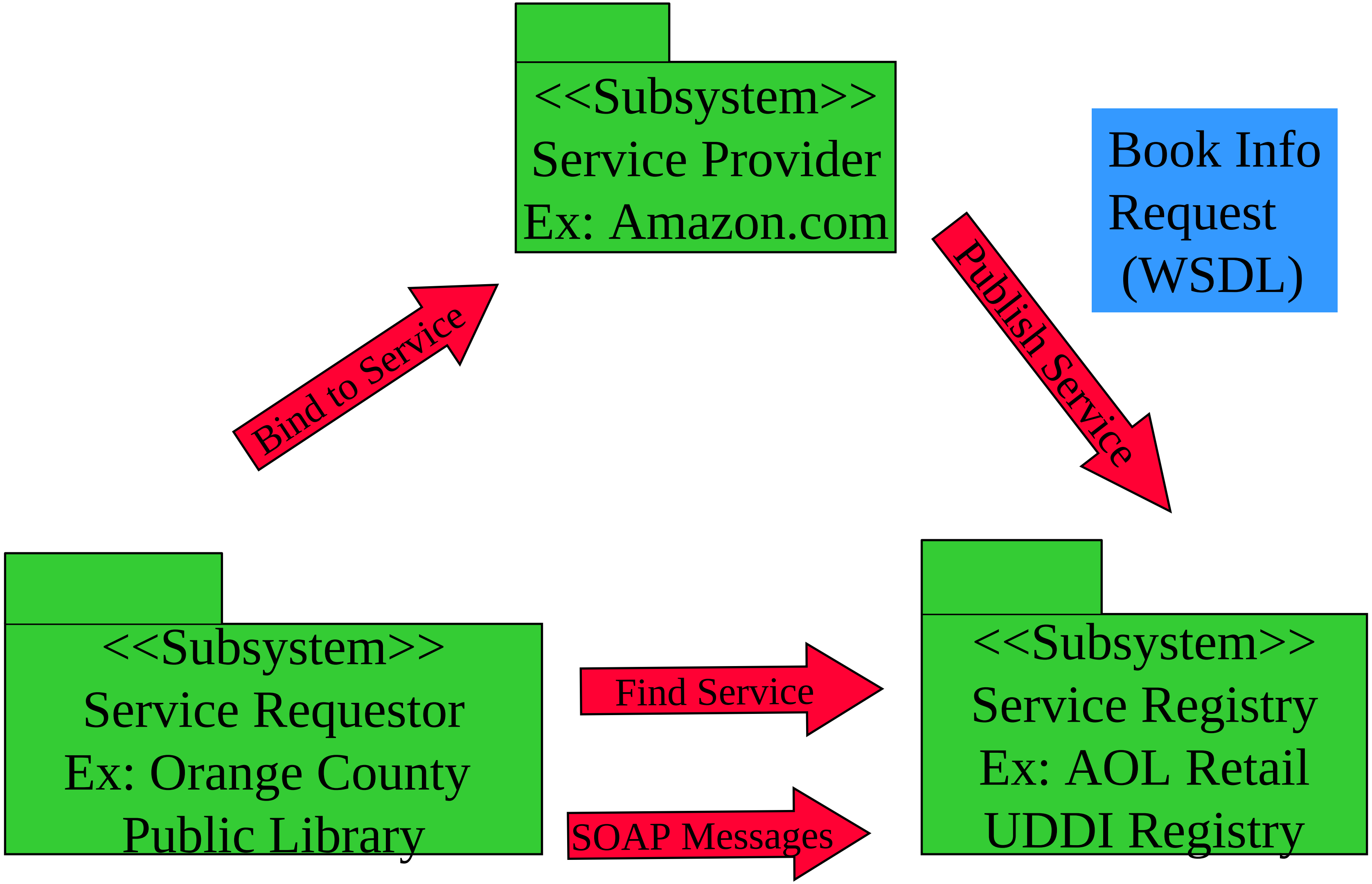


http://www.sdmagazine.com/documents/s=2398/sdm0202f/0202ff1.gif - Microsoft Internet Explorer

Address http://www.sdmagazine.com/documents/s=2398/sdm0202f/0202ff1.gif

Source: Software Development – February 2002
Modeling XML Applications with UML, David Carlson

# Web Services for Book Store – An Example



<<Subsystem>>
Service Provider
Ex: Amazon.com

Book Info Request (WSDL)

Bind to Service

Publish Service

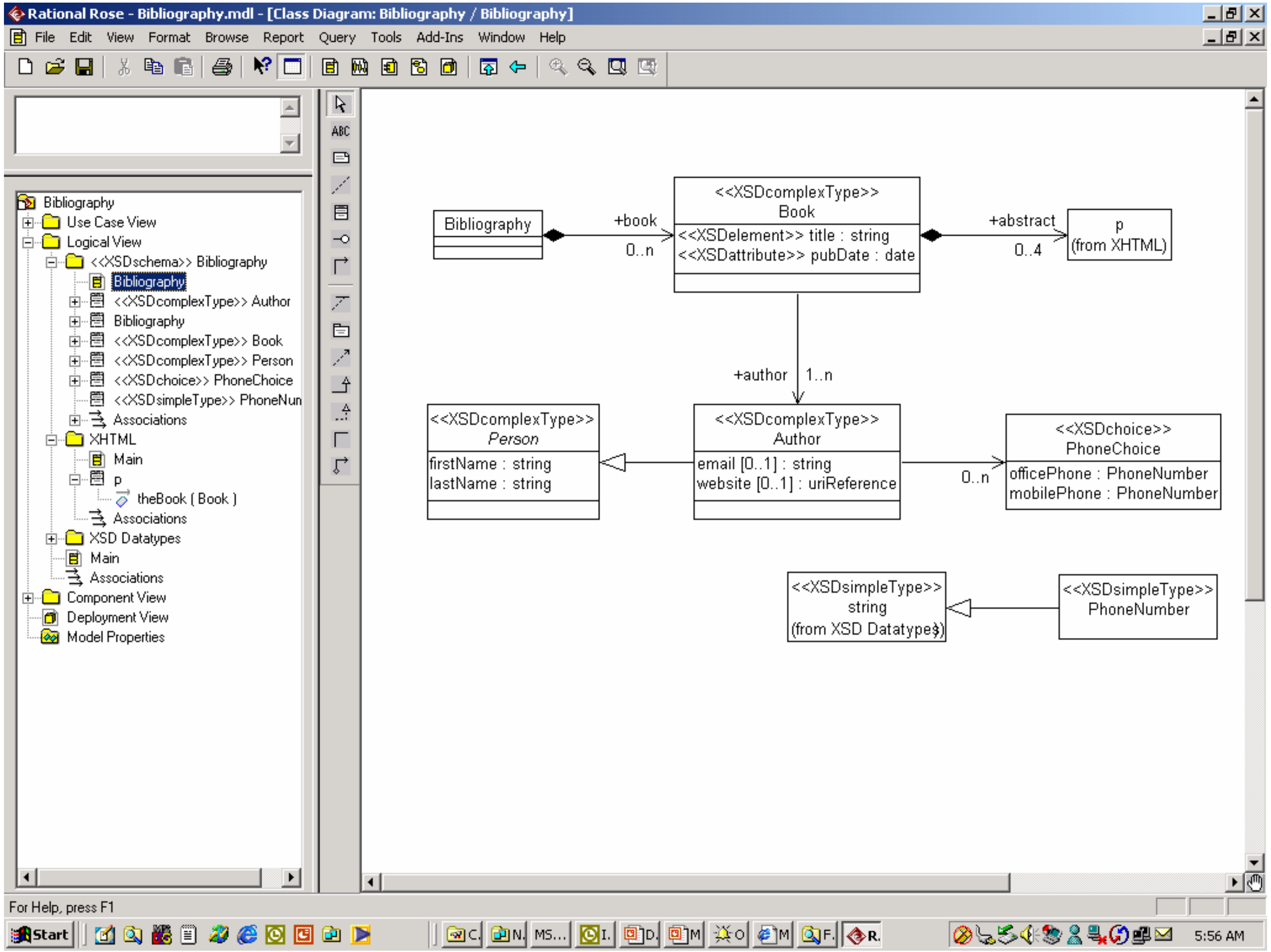
<<Subsystem>>
Service Requestor
Ex: Orange County Public Library

Find Service

SOAP Messages

<<Subsystem>>
Service Registry
Ex: AOL Retail UDDI Registry

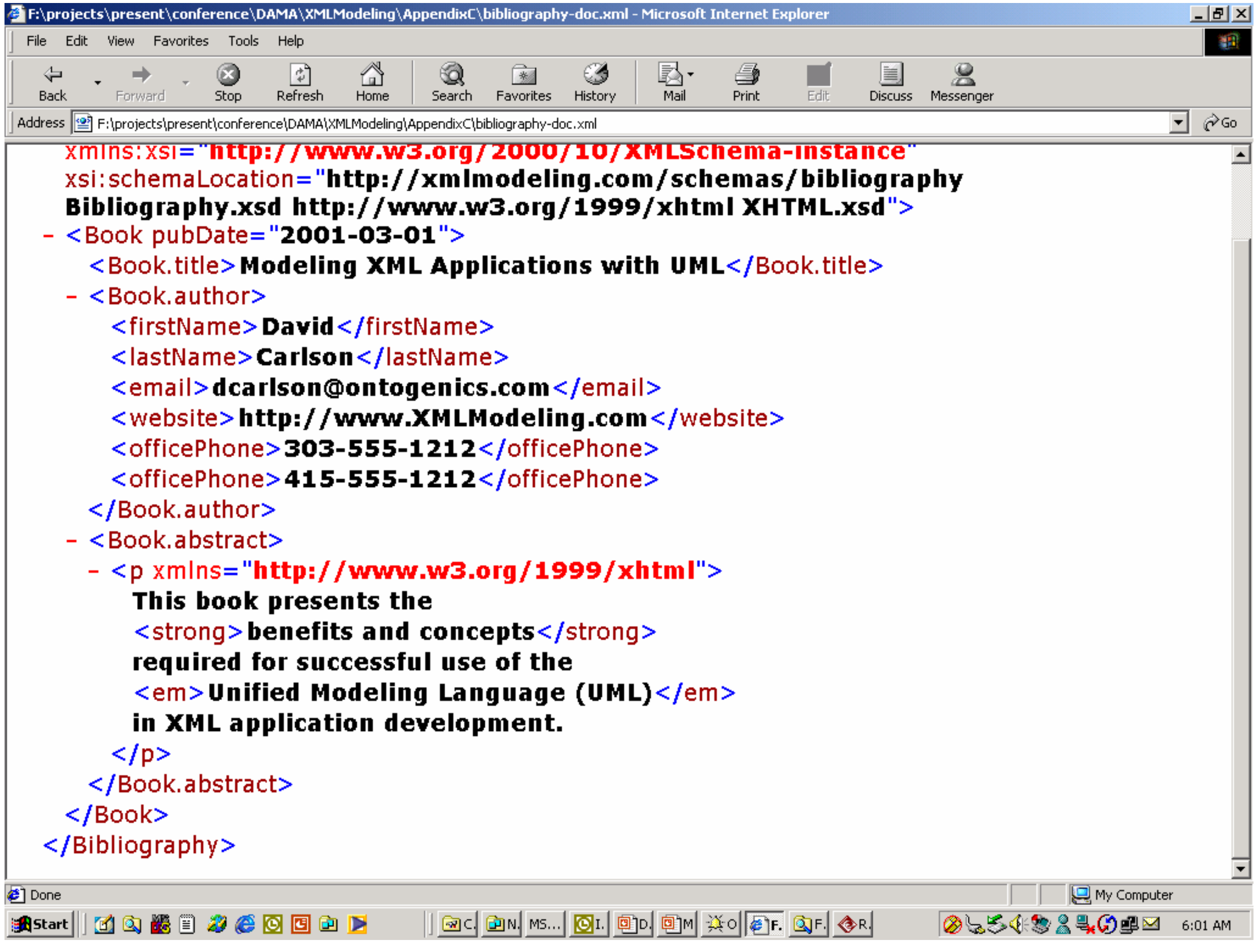
# Modeling Web Services and XML



Usage : Model the Application

Source : David Carlson www.xmlmodeling.com

# An XML Document that conforms to the UML model

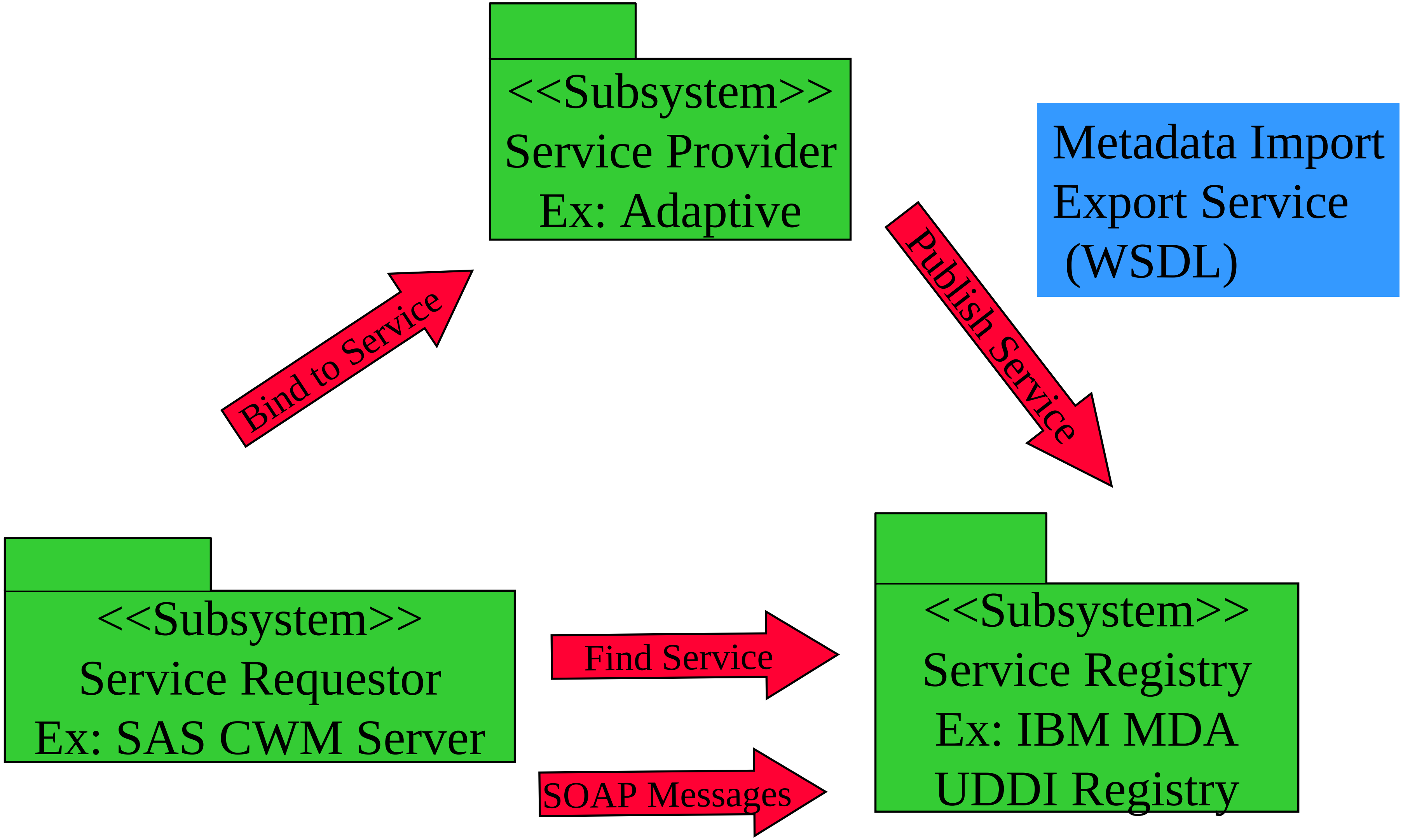

F:\projects\present\conference\DAMA\XMLModeling\AppendixC\bibliography-doc.xml - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss Messenger
Address F:\projects\present\conference\DAMA\XMLModeling\AppendixC\bibliography-doc.xml
Go
Done
My Computer
Start
6:01 AM

```
    xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
    xsi:schemaLocation="http://xmlmodeling.com/schemas/bibliography
    Bibliography.xsd http://www.w3.org/1999/xhtml XHTML.xsd">
  - <Book pubDate="2001-03-01">
      <Book.title>Modeling XML Applications with UML</Book.title>
    - <Book.author>
        <firstName>David</firstName>
        <lastName>Carlson</lastName>
        <email>dcarlson@ontogenics.com</email>
        <website>http://www.XMLModeling.com</website>
        <officePhone>303-555-1212</officePhone>
        <officePhone>415-555-1212</officePhone>
      </Book.author>
    - <Book.abstract>
      - <p xmlns="http://www.w3.org/1999/xhtml">
          This book presents the
          <strong>benefits and concepts</strong>
          required for successful use of the
          <em>Unified Modeling Language (UML)</em>
          in XML application development.
        </p>
      </Book.abstract>
    </Book>
</Bibliography>
```

Usage : Generate XML Schemas and Documents

Source : David Carlson www.xmlmodeling.com

# *Web Services for Metadata Management – An Example*



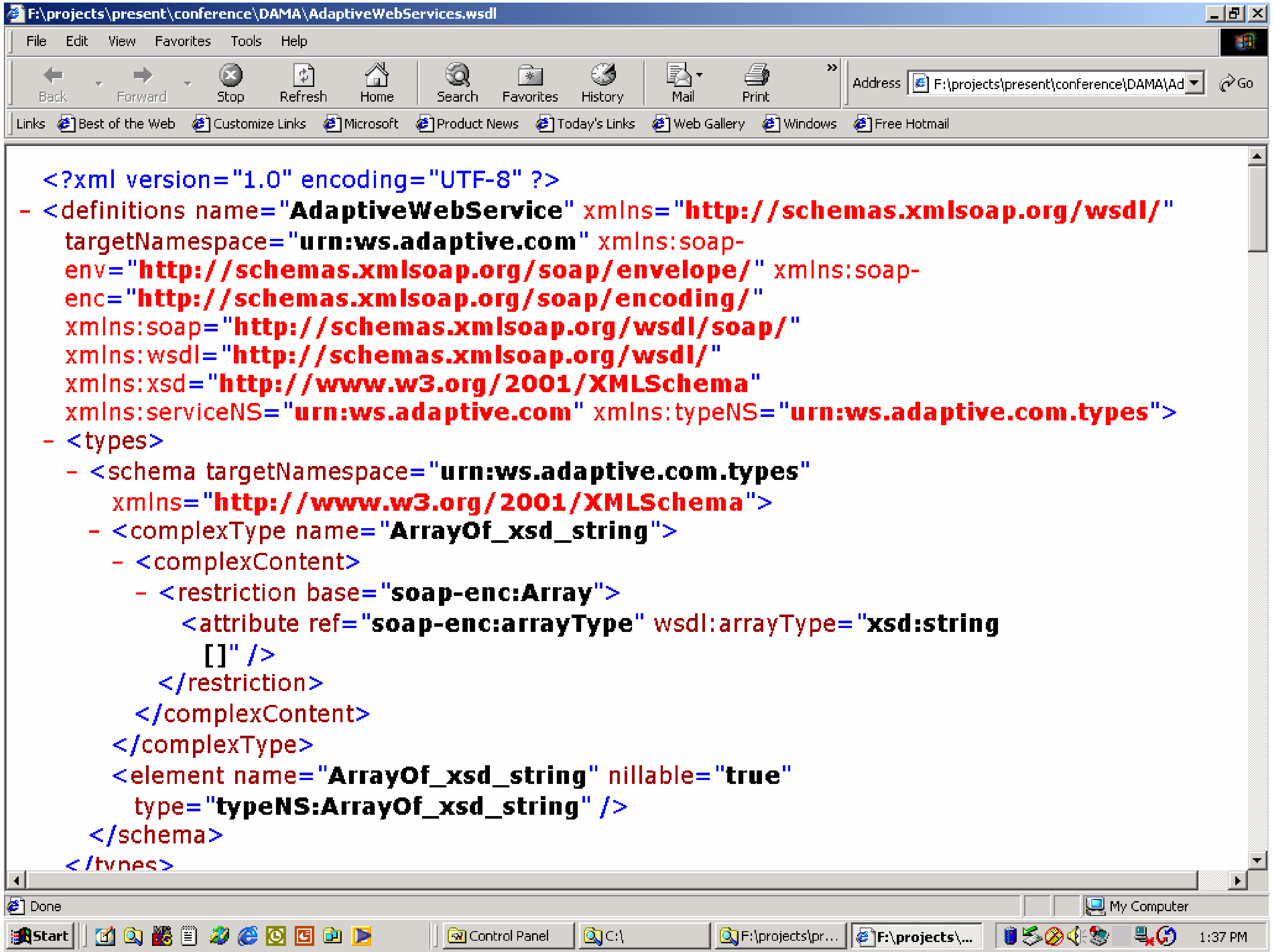
<<Subsystem>>
Service Provider
Ex: Adaptive

Metadata Import Export Service (WSDL)

<<Subsystem>>
Service Requestor
Ex: SAS CWM Server

<<Subsystem>>
Service Registry
Ex: IBM MDA UDDI Registry

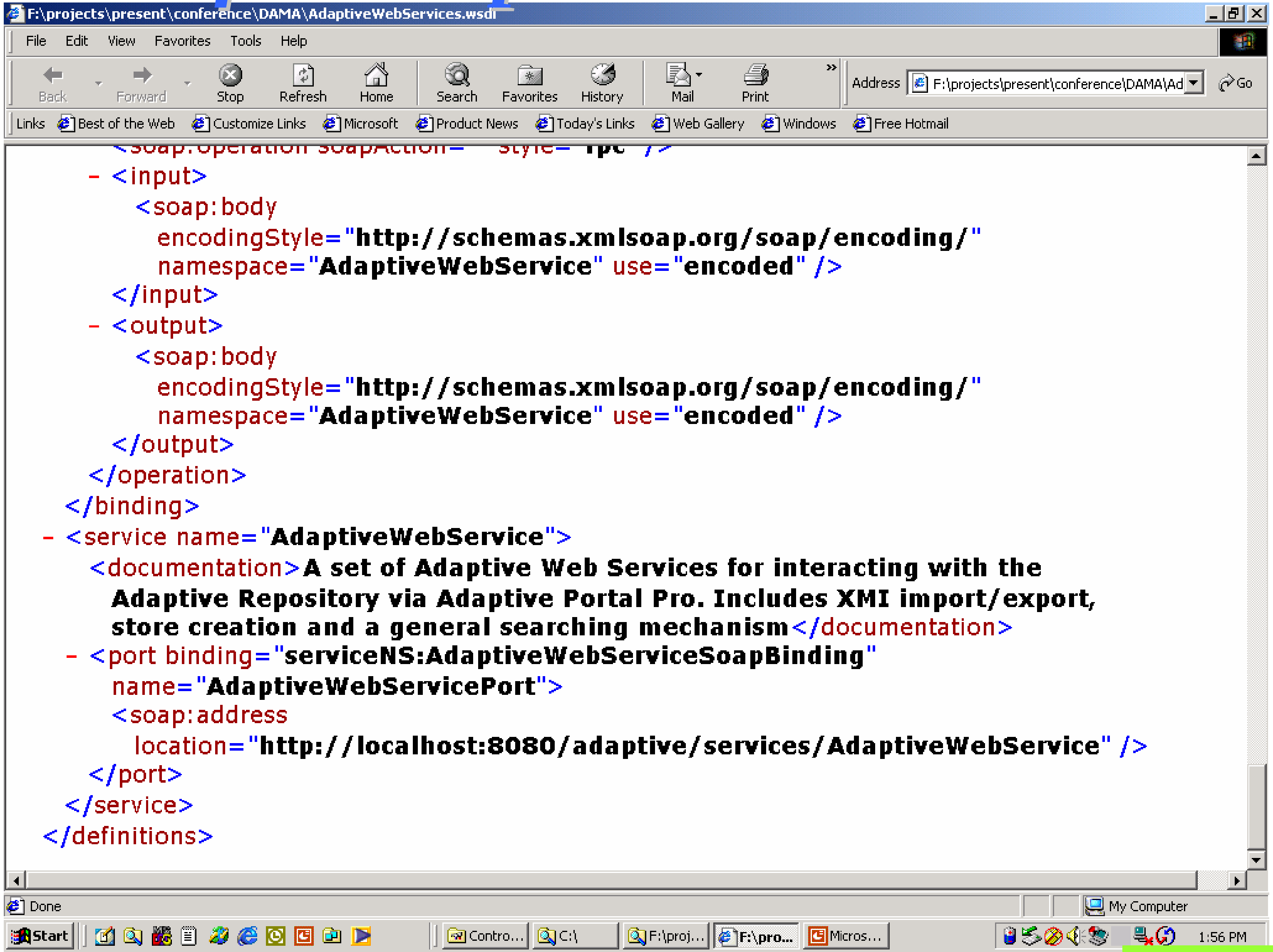
# Web Services for Metadata Management
# WSDL Header



```
  <?xml version="1.0" encoding="UTF-8" ?>
- <definitions name="AdaptiveWebService" xmlns="http://schemas.xmlsoap.org/wsdl/"
    targetNamespace="urn:ws.adaptive.com" xmlns:soap-
    env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap-
    enc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:serviceNS="urn:ws.adaptive.com" xmlns:typeNS="urn:ws.adaptive.com.types">
  - <types>
    - <schema targetNamespace="urn:ws.adaptive.com.types"
        xmlns="http://www.w3.org/2001/XMLSchema">
      - <complexType name="ArrayOf_xsd_string">
        - <complexContent>
          - <restriction base="soap-enc:Array">
              <attribute ref="soap-enc:arrayType" wsdl:arrayType="xsd:string
                []" />
            </restriction>
          </complexContent>
        </complexType>
        <element name="ArrayOf_xsd_string" nillable="true"
          type="typeNS:ArrayOf_xsd_string" />
      </schema>
    </types>
```

Courtesy : Adaptive

# Web Services for Metadata Management
# WSDL for Adaptive Web Services Port

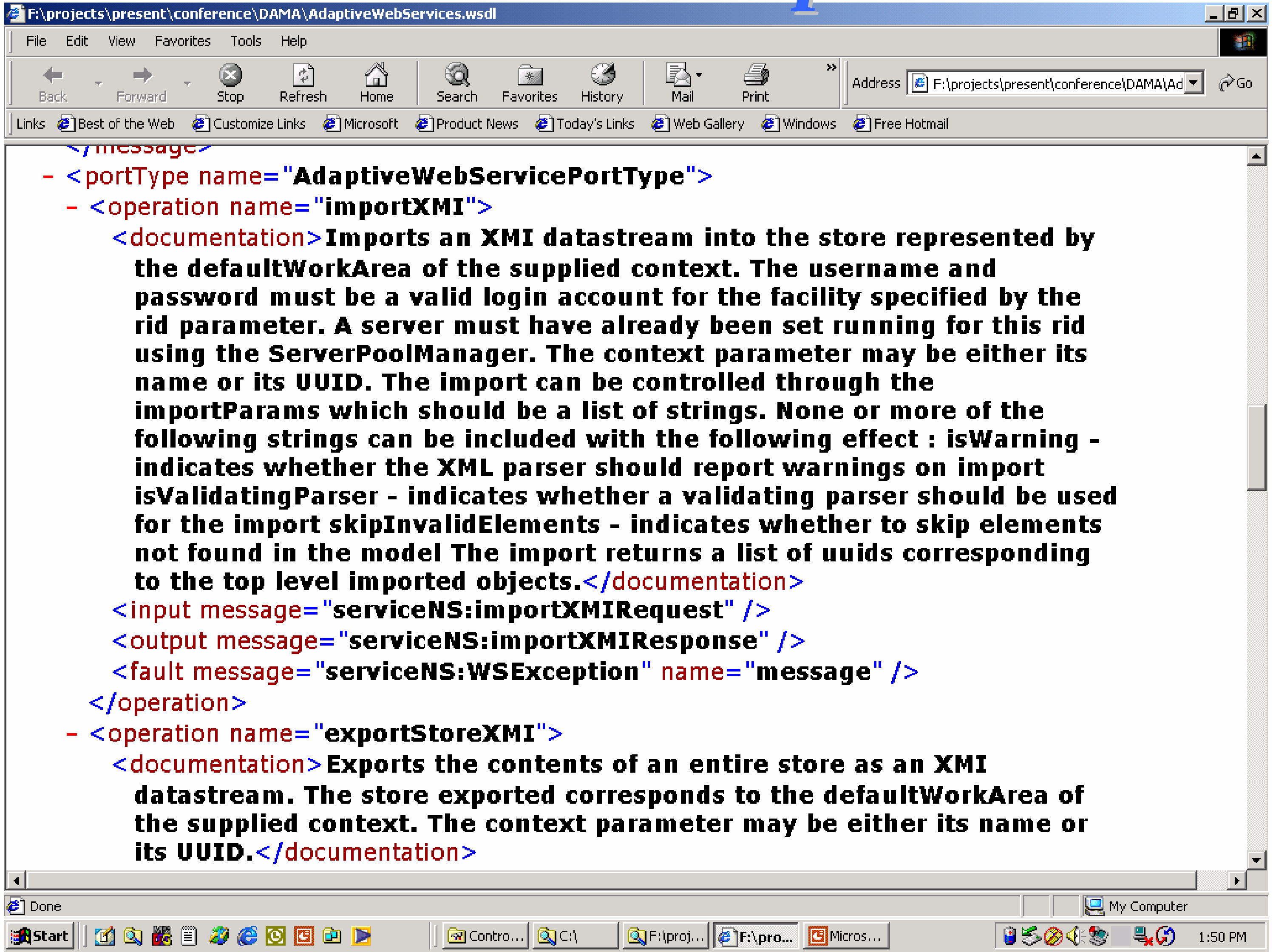


F:\projects\present\conference\DAMA\AdaptiveWebServices.wsdl
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address F:\projects\present\conference\DAMA\Ad
Go
Links Best of the Web Customize Links Microsoft Product News Today's Links Web Gallery Windows Free Hotmail
Done
My Computer
Start
Contro... C:\ F:\proj... F:\pro... Micros...
1:56 PM

```
      <soap:operation soapAction="" style="rpc" />
    - <input>
        <soap:body
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="AdaptiveWebService" use="encoded" />
      </input>
    - <output>
        <soap:body
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="AdaptiveWebService" use="encoded" />
      </output>
    </operation>
  </binding>
- <service name="AdaptiveWebService">
    <documentation>A set of Adaptive Web Services for interacting with the
      Adaptive Repository via Adaptive Portal Pro. Includes XMI import/export,
      store creation and a general searching mechanism</documentation>
  - <port binding="serviceNS:AdaptiveWebServiceSoapBinding"
      name="AdaptiveWebServicePort">
      <soap:address
        location="http://localhost:8080/adaptive/services/AdaptiveWebService" />
    </port>
  </service>
</definitions>
```

Courtesy : Adaptive

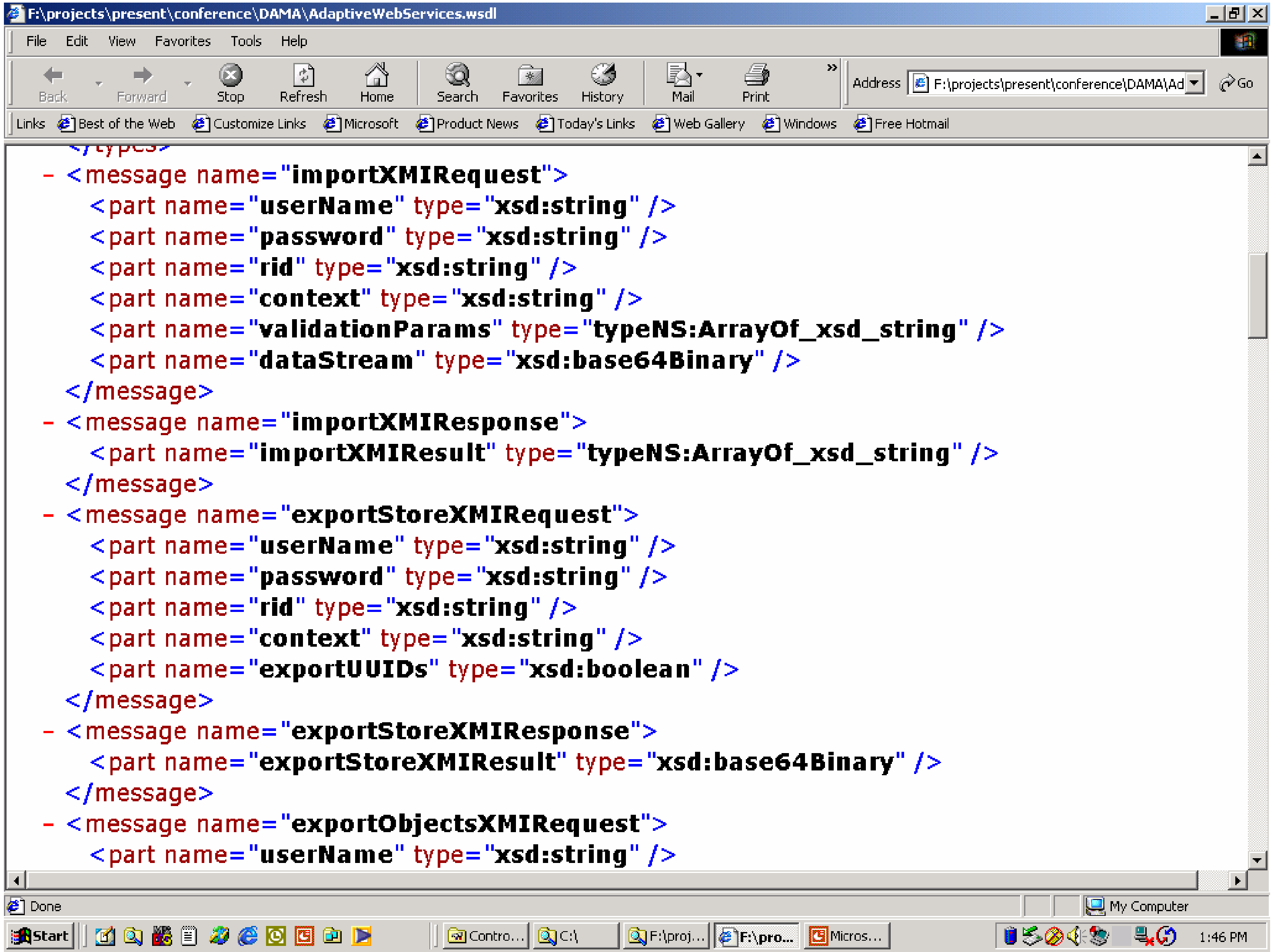
# Web Services for Metadata Management
# WSDL Ports and Operations



F:\projects\present\conference\DAMA\AdaptiveWebServices.wsdl
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address F:\projects\present\conference\DAMA\Ad
Go
Links Best of the Web Customize Links Microsoft Product News Today's Links Web Gallery Windows Free Hotmail
Done
My Computer
Start
Contro... C:\ F:\proj... F:\pro... Micros...
1:50 PM

```
    </message>
  - <portType name="AdaptiveWebServicePortType">
    - <operation name="importXMI">
        <documentation>Imports an XMI datastream into the store represented by
          the defaultWorkArea of the supplied context. The username and
          password must be a valid login account for the facility specified by the
          rid parameter. A server must have already been set running for this rid
          using the ServerPoolManager. The context parameter may be either its
          name or its UUID. The import can be controlled through the
          importParams which should be a list of strings. None or more of the
          following strings can be included with the following effect : isWarning -
          indicates whether the XML parser should report warnings on import
          isValidatingParser - indicates whether a validating parser should be used
          for the import skipInvalidElements - indicates whether to skip elements
          not found in the model The import returns a list of uuids corresponding
          to the top level imported objects.</documentation>
        <input message="serviceNS:importXMIRequest" />
        <output message="serviceNS:importXMIResponse" />
        <fault message="serviceNS:WSException" name="message" />
      </operation>
    - <operation name="exportStoreXMI">
        <documentation>Exports the contents of an entire store as an XMI
          datastream. The store exported corresponds to the defaultWorkArea of
          the supplied context. The context parameter may be either its name or
          its UUID.</documentation>
```

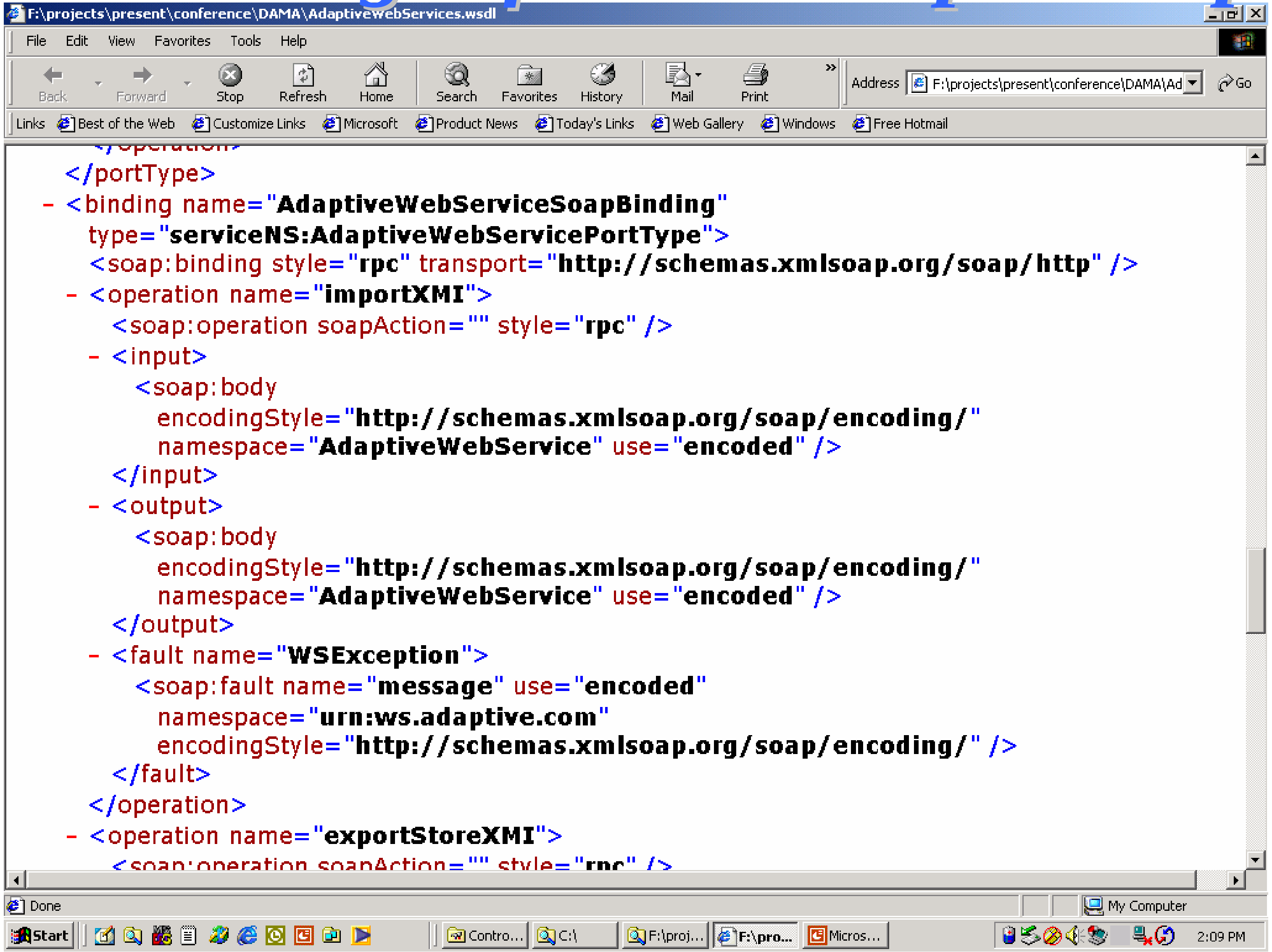
# Web Services for Metadata Management
# WSDL messages for XMI Import/Export



F:\projects\present\conference\DAMA\AdaptiveWebServices.wsdl
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address F:\projects\present\conference\DAMA\Ad Go
Links Best of the Web Customize Links Microsoft Product News Today's Links Web Gallery Windows Free Hotmail
Done
My Computer
Start Contro... C:\ F:\proj... F:\pro... Micros...
1:46 PM

```
  </types>
- <message name="importXMIRequest">
    <part name="userName" type="xsd:string" />
    <part name="password" type="xsd:string" />
    <part name="rid" type="xsd:string" />
    <part name="context" type="xsd:string" />
    <part name="validationParams" type="typeNS:ArrayOf_xsd_string" />
    <part name="dataStream" type="xsd:base64Binary" />
  </message>
- <message name="importXMIResponse">
    <part name="importXMIResult" type="typeNS:ArrayOf_xsd_string" />
  </message>
- <message name="exportStoreXMIRequest">
    <part name="userName" type="xsd:string" />
    <part name="password" type="xsd:string" />
    <part name="rid" type="xsd:string" />
    <part name="context" type="xsd:string" />
    <part name="exportUUIDs" type="xsd:boolean" />
  </message>
- <message name="exportStoreXMIResponse">
    <part name="exportStoreXMIResult" type="xsd:base64Binary" />
  </message>
- <message name="exportObjectsXMIRequest">
    <part name="userName" type="xsd:string" />
```

# Web Services for Metadata Management
# WSDL messages for XMI Import/Export


F:\projects\present\conference\DAMA\AdaptiveWebServices.wsdl
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address F:\projects\present\conference\DAMA\Ad
Go
Links Best of the Web Customize Links Microsoft Product News Today's Links Web Gallery Windows Free Hotmail
Done
My Computer
Start
Contro... C:\ F:\proj... F:\pro... Micros...
2:09 PM

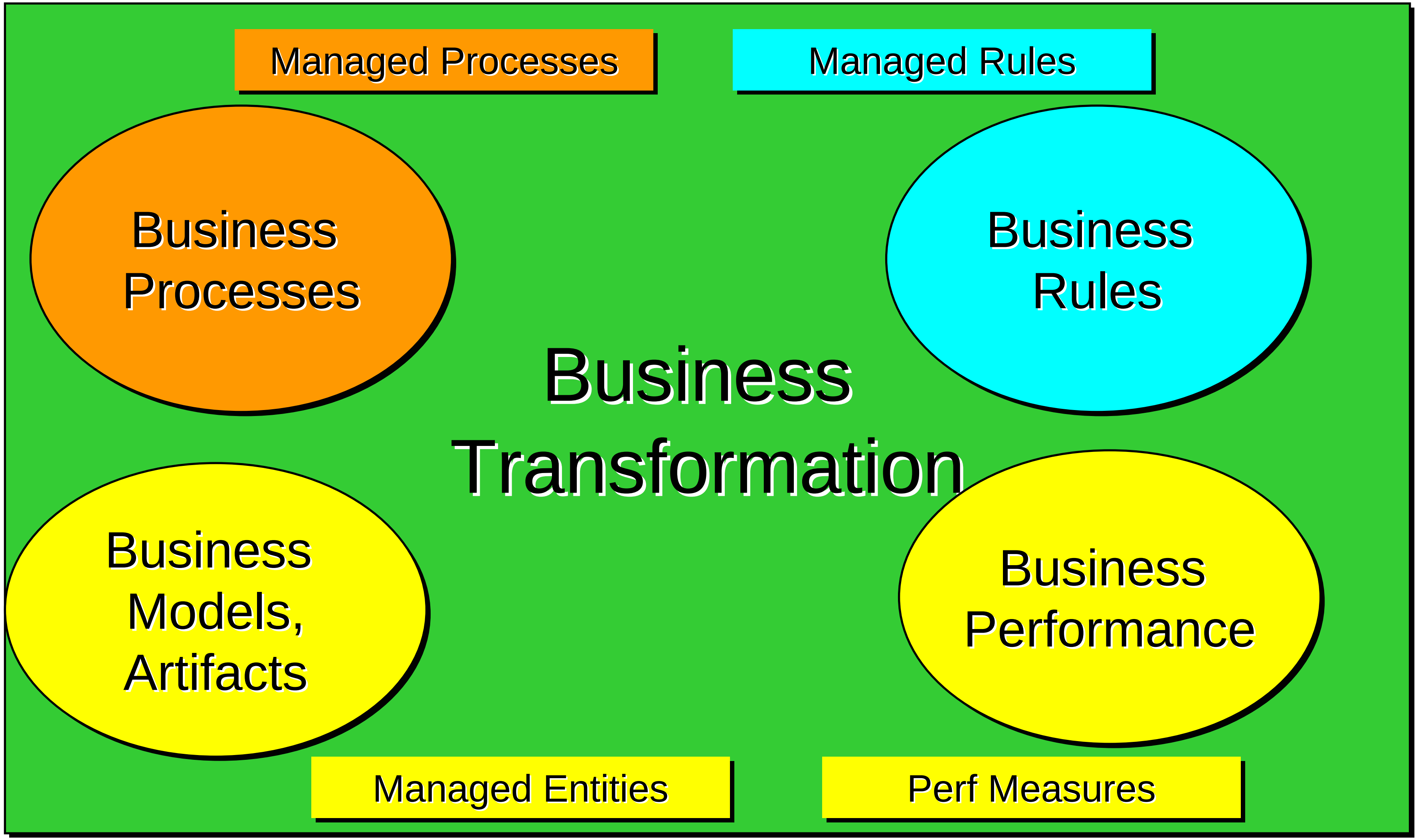
```
      </operation>
    </portType>
  - <binding name="AdaptiveWebServiceSoapBinding"
      type="serviceNS:AdaptiveWebServicePortType">
      <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
    - <operation name="importXMI">
        <soap:operation soapAction="" style="rpc" />
      - <input>
          <soap:body
            encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            namespace="AdaptiveWebService" use="encoded" />
        </input>
      - <output>
          <soap:body
            encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            namespace="AdaptiveWebService" use="encoded" />
        </output>
      - <fault name="WSException">
          <soap:fault name="message" use="encoded"
            namespace="urn:ws.adaptive.com"
            encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
        </fault>
      </operation>
    - <operation name="exportStoreXMI">
        <soap:operation soapAction="" style="rpc" />
```

Courtesy : Adaptive

# How do Web Services fit into existing Software Architectures

- W3C Architecture Working Group
  - Architectural framework and liaison with other groups
- OMG Model Driven Architecture
  - OMG CWM for Web Services, MOF to WSDL, UDDI
  - IDL/WSDL and WSDL/IDL Mappings
  - OMG Web Services SIG
- J2EE – Web Services Architectural extensions
  - JAXP, JAXM, JAXB, JAXR, JAX-RPC
- Microsoft .Net ‘connected’
  - Weaving Web Services everywhere
- OMG  – Integrating Business Models, Processes & rules with J2EE,Web Services and MDA
  - UML2, Business Process Definition Metamodel, Business Rules RFP (coming)

# Business Models, Processes and Rules



Managed Processes

Managed Rules

Business Processes

Business Rules

Business Transformation

Business Models, Artifacts

Business Performance

Managed Entities

Perf Measures

# A peek at integrating Business Processes using UML and BPEL4WS

This is early work and is intended to spur Discussion!

See the BPEL4WS spec for the examples, BPEL4WS syntax

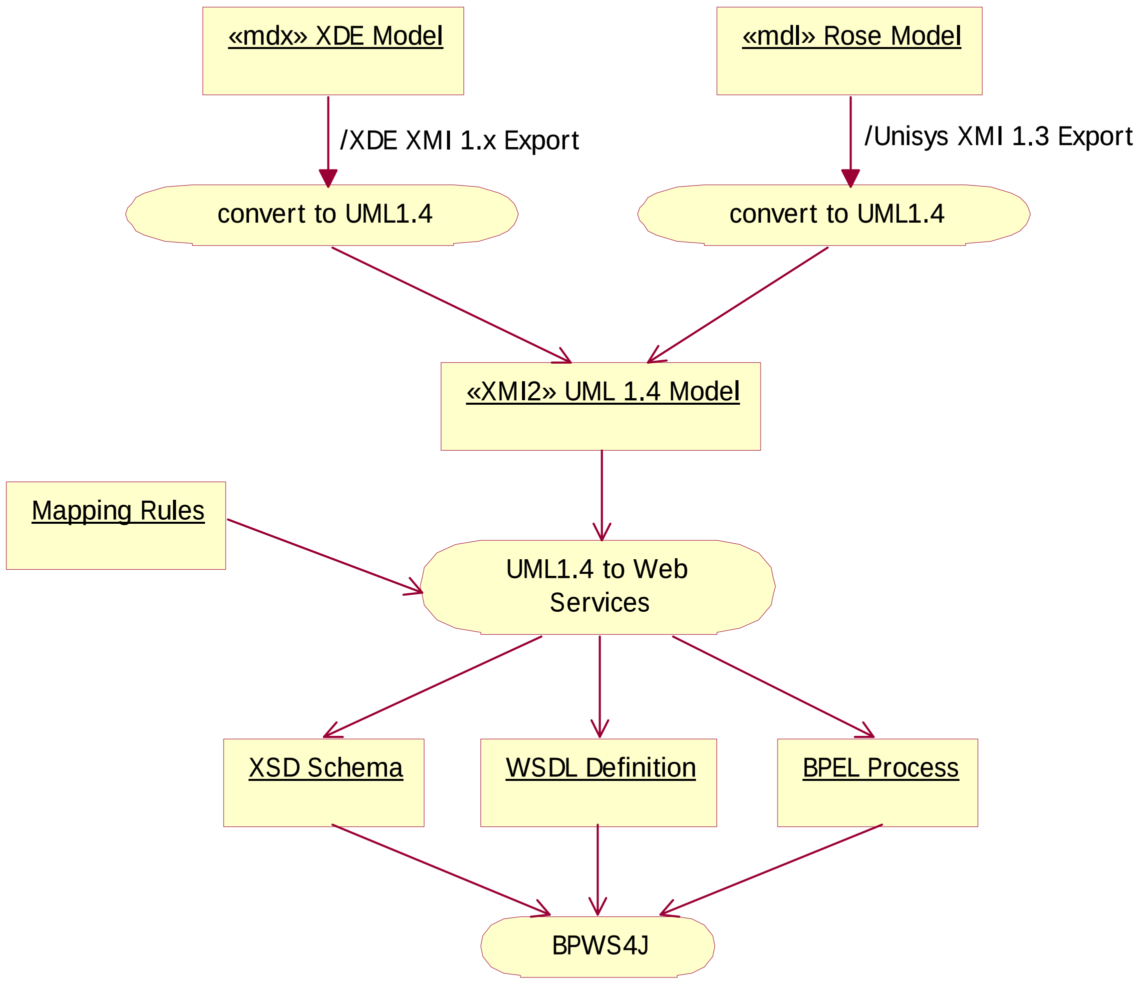
May be this will result in a standard UML profile for BPEL

Update : BPD Profile submitted to OMG

# BPEL4WS

- The Business Process Execution Language for Web Services (BPEL4WS) provides an XML notation and semantics for specifying business process behavior based on Web Services.
- A BPEL4WS process is defined in terms of its interactions with partners. A partner may provide services to the process, require services from the process, or participate in a two-way interaction with the process.
- A short summary of research underway at IBM is described to illustrate how MDA and models can be used for business process integration

# ‘Business Integration Profile’ using MDA and Web Services



# Profile Principles

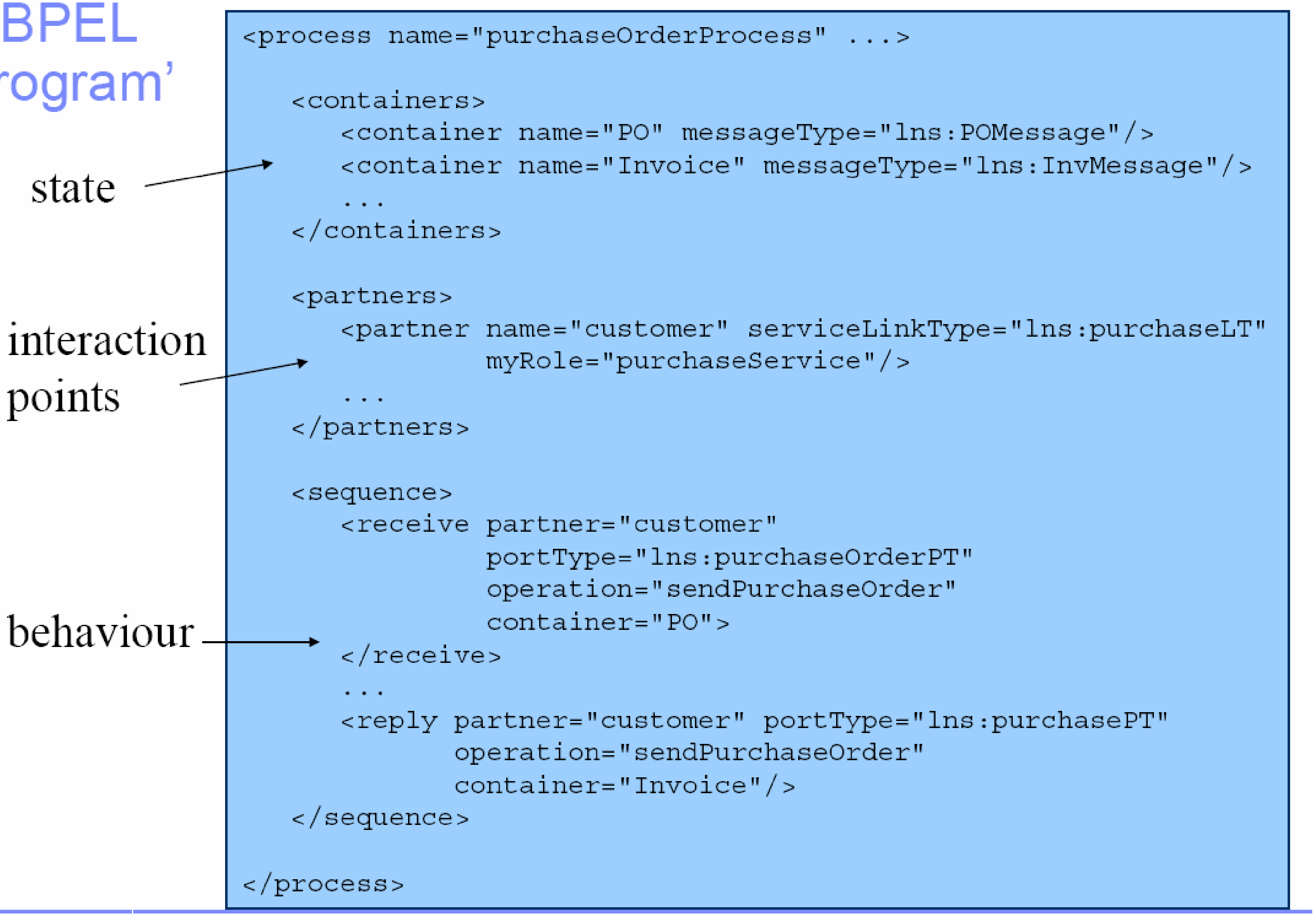
- The profile should cover broadly the same set of concepts as BPEL
- Support the concepts of XSD and WSDL that are required to support BPEL, but don't cover the whole of service oriented architecture in this profile
- Standard UML terminology for concepts is used where available, e.g. Interface rather than PortType
- Where UML 2 will have more direct support for concepts then the profile adopts a UML 2 style (e.g. introducing a notion of ports)
- In areas of UML that are better defined in UML 2 then the UML 2 semantics is assumed
- It should be possible to create models conforming to the profile using multiple UML editors, specifically Rose and XDE

# Purchase Order (PO) Process

- Customer sends purchase order to purchase service provided by PO process
- PO process asks for initial price, shipping information, and production schedule
- Shipping requestor determines where order items will be obtained and creates source and destination shipping information
- When price and shipping information are available, invoice provider calculates final price and sends invoice to PO process
- The PO process asks a scheduling provider to determine when each order item will be produced and instructs the scheduling provider to send a schedule to the customer
- Finally, the PO process replies to the customer with an invoice

# The PO Process in  BPEL4WS

A BPEL
‘program’



state

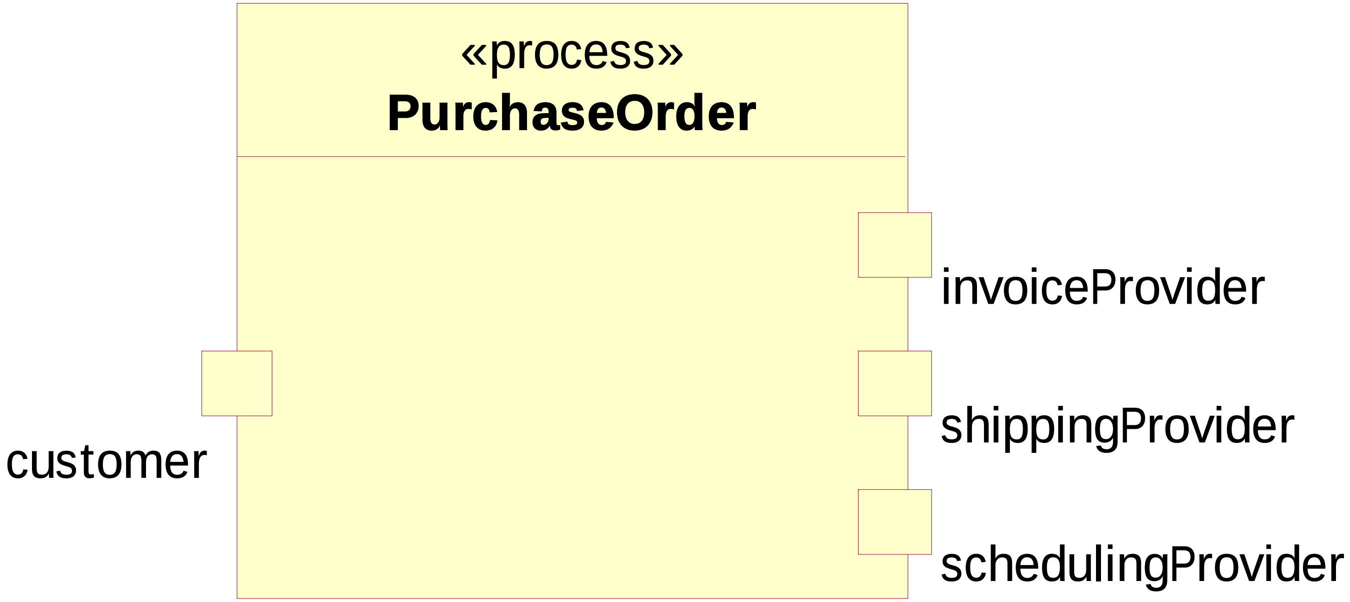
interaction
points

behaviour

```
<process name="purchaseOrderProcess" ...>

   <containers>
      <container name="PO" messageType="lns:POMessage"/>
      <container name="Invoice" messageType="lns:InvMessage"/>
      ...
   </containers>

   <partners>
      <partner name="customer" serviceLinkType="lns:purchaseLT"
               myRole="purchaseService"/>
      ...
   </partners>

   <sequence>
      <receive partner="customer"
               portType="lns:purchaseOrderPT"
               operation="sendPurchaseOrder"
               container="PO">
      </receive>
      ...
      <reply partner="customer" portType="lns:purchasePT"
             operation="sendPurchaseOrder"
             container="Invoice"/>
   </sequence>

</process>
```

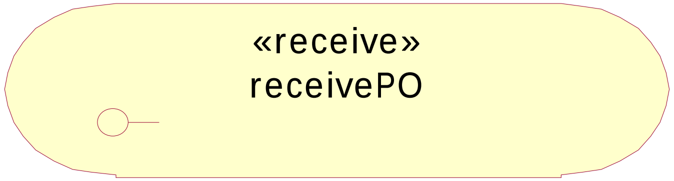
# BPEL partners as ports

- This is the UML 2 version, we approximate this in UML 1.4 (more detail later)



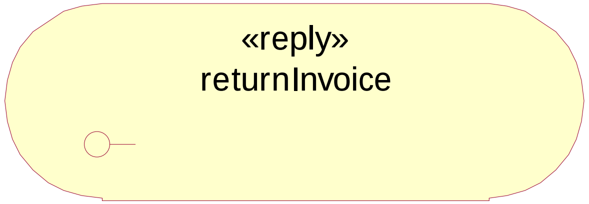
# Behavior Basics

Receive a synchronous operation call:



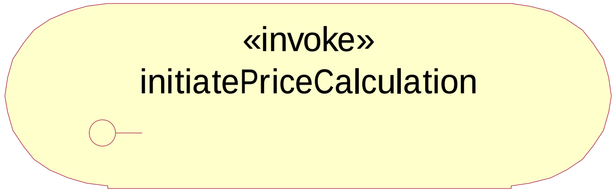
«receive»
receivePO

Reply to a previous synchronous request:



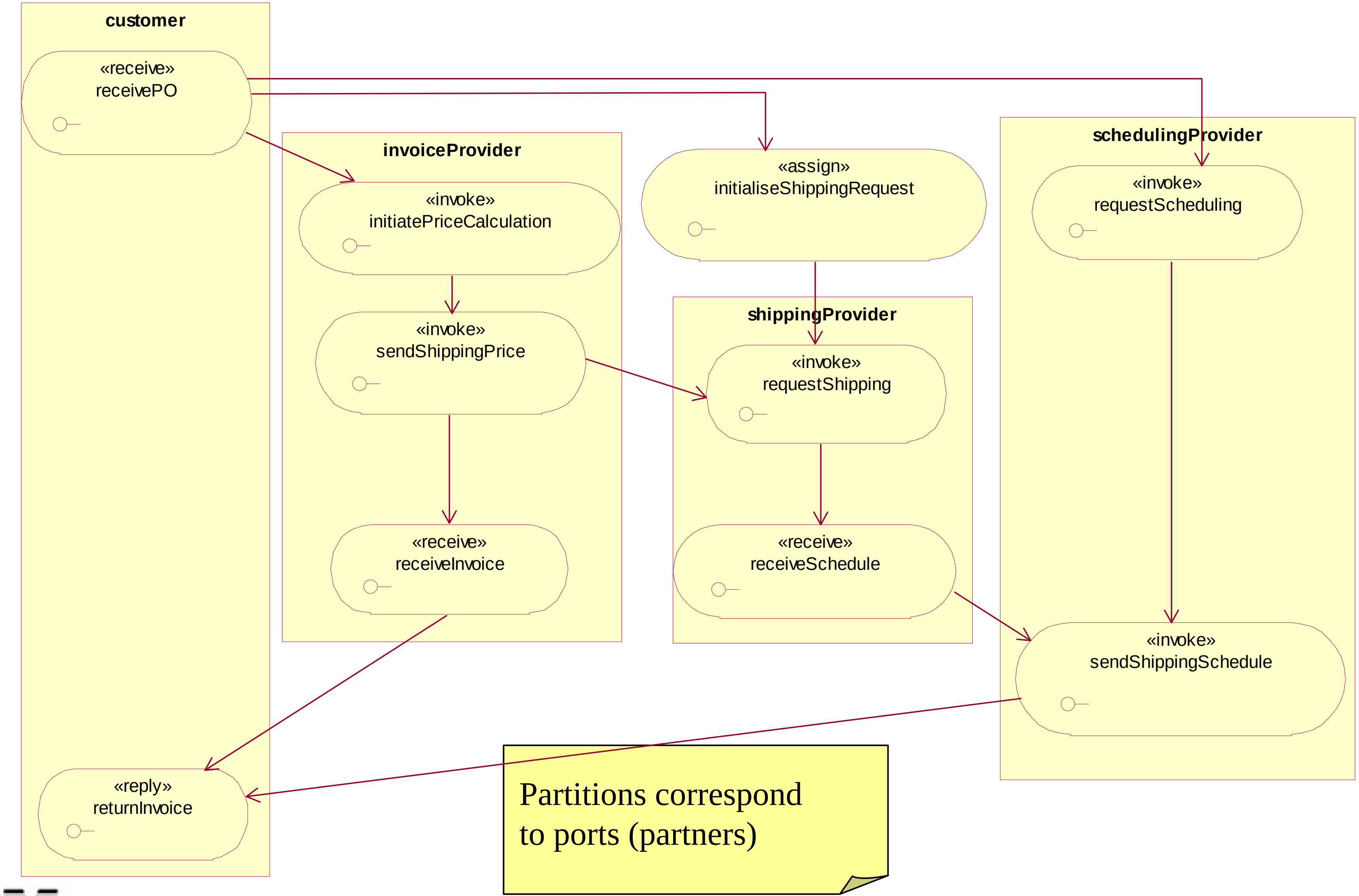
«reply»
returnInvoice

Invoke an operation synchronously or asynchronously on a partner:



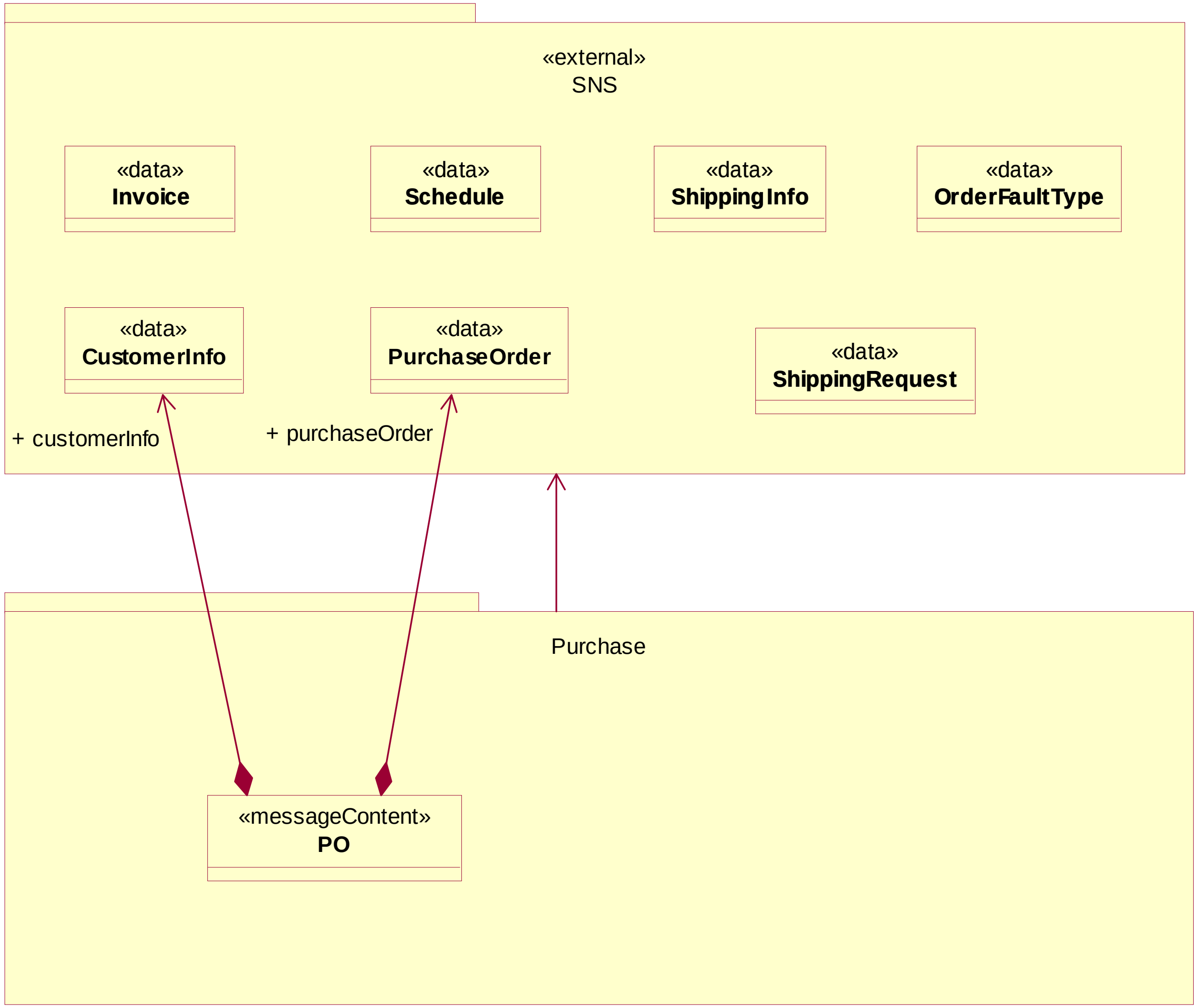
«invoke»
initiatePriceCalculation

# Purchase Order Process

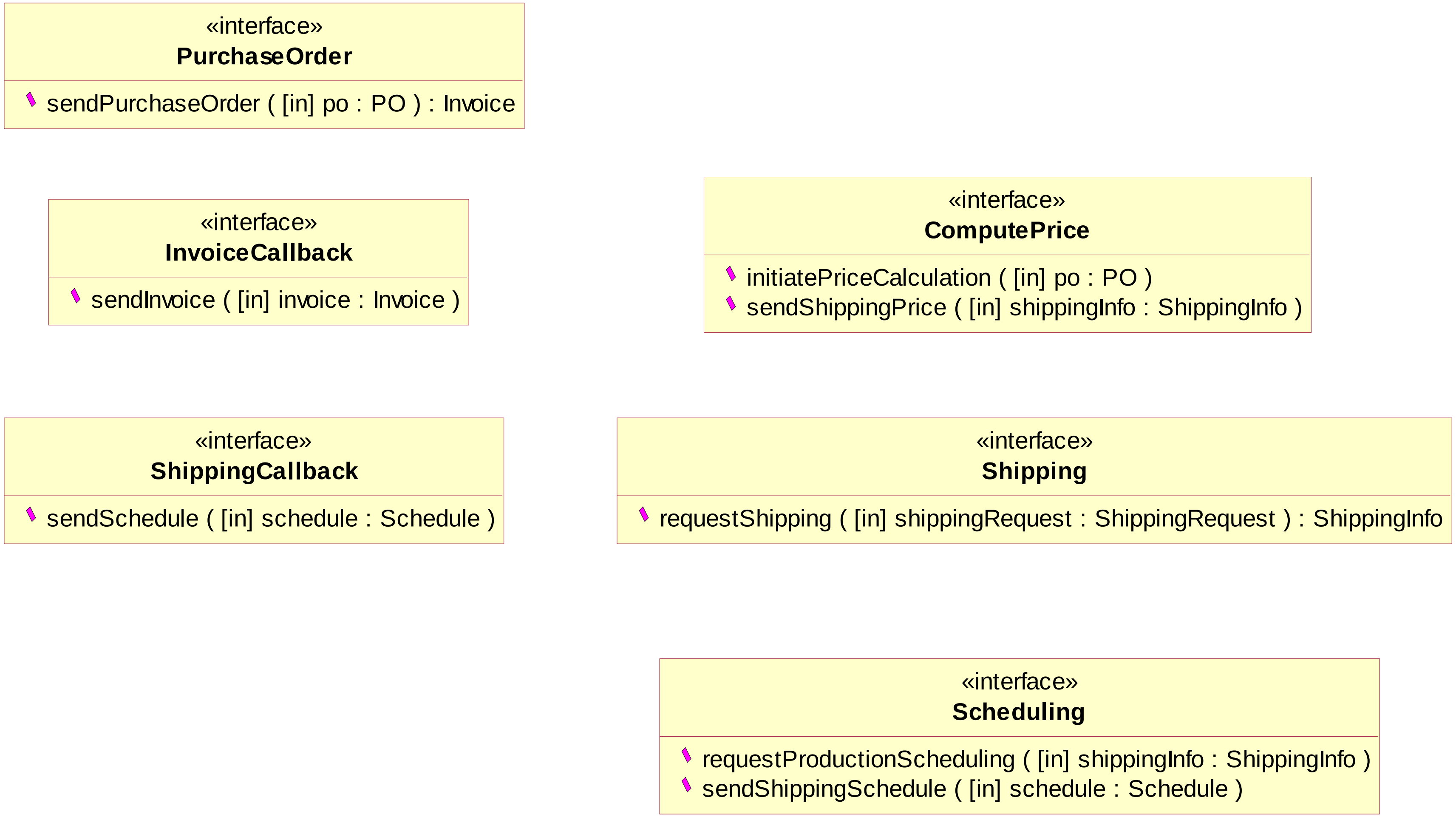


Partitions correspond to ports (partners)

# Data and Message Types



# Interfaces



«interface»
**PurchaseOrder**

- sendPurchaseOrder ( [in] po : PO ) : Invoice

«interface»
**InvoiceCallback**

- sendInvoice ( [in] invoice : Invoice )

«interface»
**ComputePrice**

- initiatePriceCalculation ( [in] po : PO )
- sendShippingPrice ( [in] shippingInfo : ShippingInfo )

«interface»
**ShippingCallback**

- sendSchedule ( [in] schedule : Schedule )

«interface»
**Shipping**

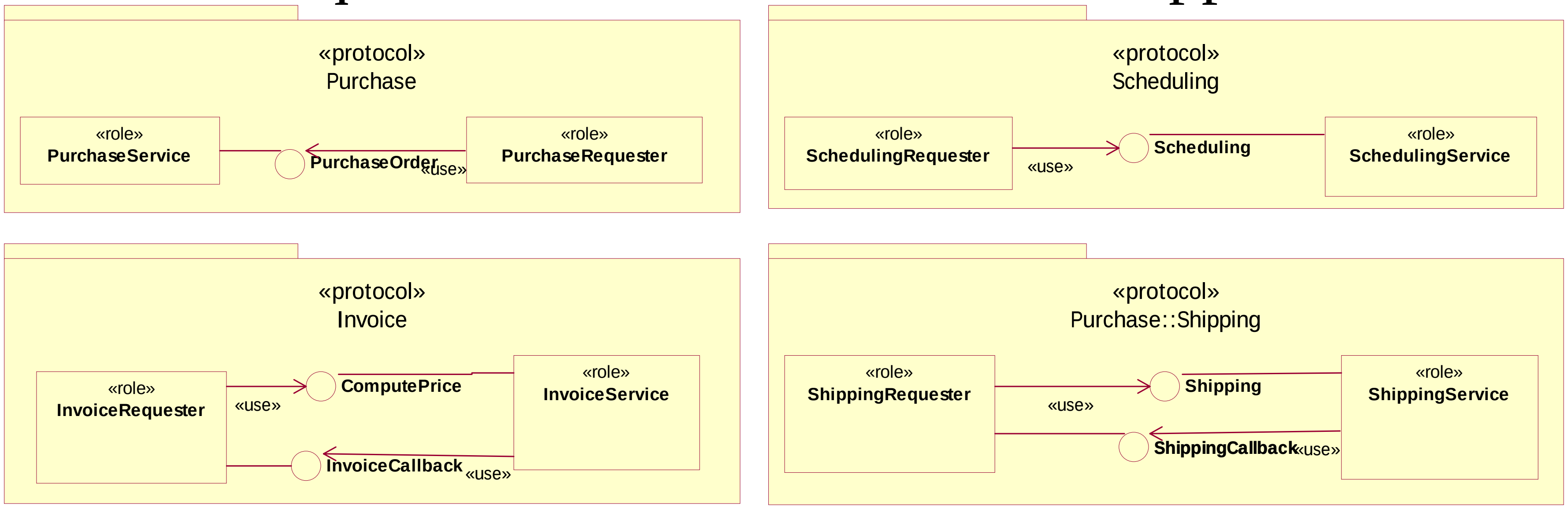
- requestShipping ( [in] shippingRequest : ShippingRequest ) : ShippingInfo

«interface»
**Scheduling**

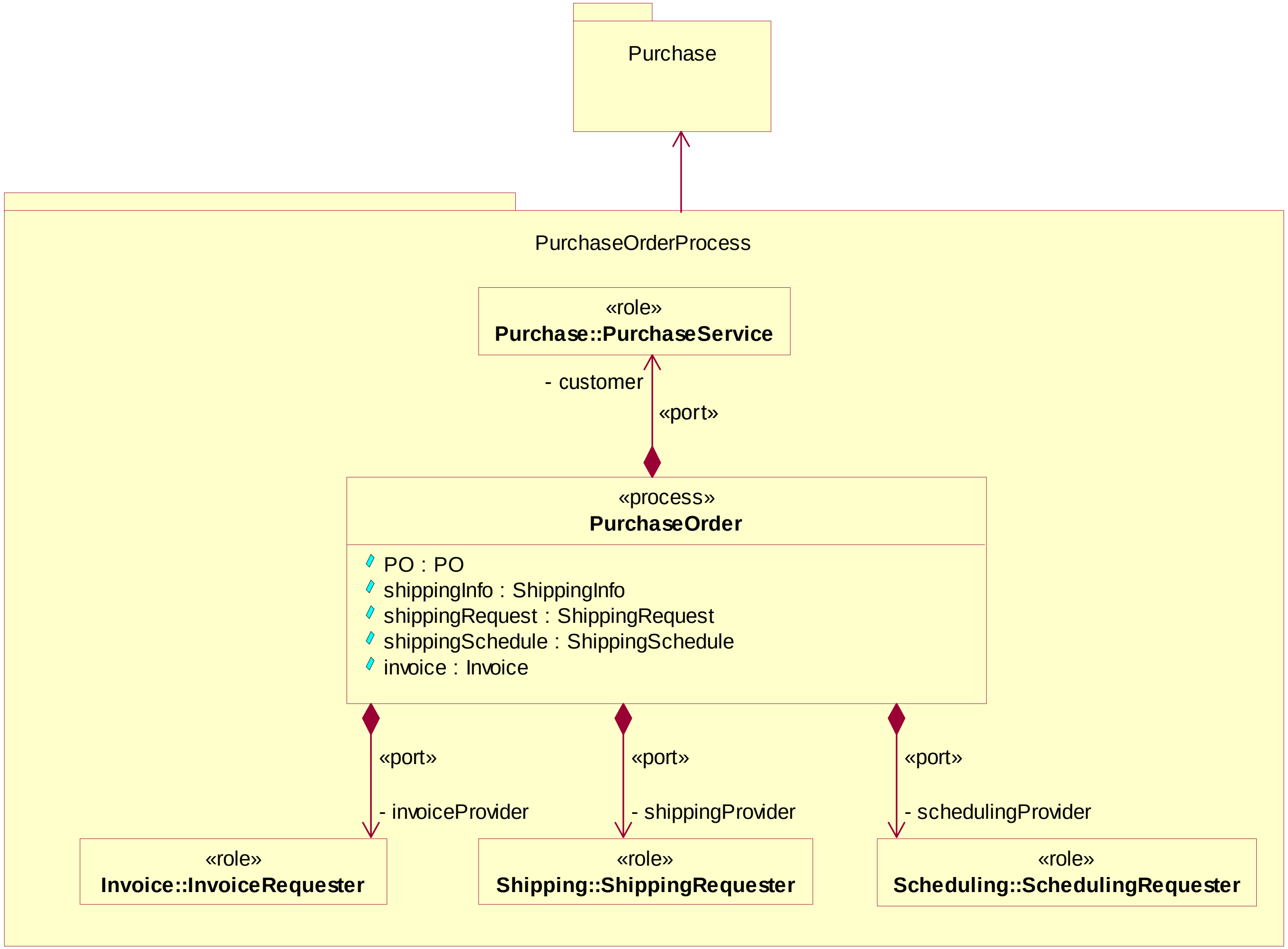
- requestProductionScheduling ( [in] shippingInfo : ShippingInfo )
- sendShippingSchedule ( [in] schedule : Schedule )

# Protocols

- Protocols are defined independently of the processes that use them
- Roles provide ‘port types’ – groupings of provided and required interfaces that must be supported



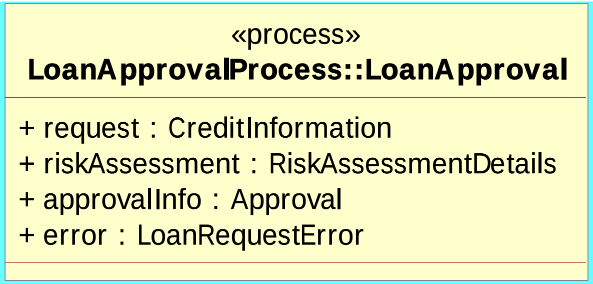
# Purchase Order Process - Structure


Purchase
PurchaseOrderProcess
«role»
Purchase::PurchaseService
- customer
«port»
«process»
PurchaseOrder
PO : PO
shippingInfo : ShippingInfo
shippingRequest : ShippingRequest
shippingSchedule : ShippingSchedule
invoice : Invoice
«port»
- invoiceProvider
«port»
- shippingProvider
«port»
- schedulingProvider
«role»
Invoice::InvoiceRequester
«role»
Shipping::ShippingRequester
«role»
Scheduling::SchedulingRequester

# BPEL Mapping Overview

| | |
|---|---|
| <<process>> class | BPEL process definition |
| Activity graph on a <<process>> class | BPEL activity hierarchy |
| <<port>> associations | BPEL partner declarations |
| <<process>> class attributes | BPEL containers |
| Hierarchical structure and control flow | BPEL sequence and flow activities |
| Decision nodes | BPEL switch activities and transition conditions |
| <<receive>>, <<reply>>, <<invoke>> activities | BPEL receive, reply, invoke activities |
| <<protocol>> package with <<role>> classes | BPEL service links types and roles |

# Mapping: Process



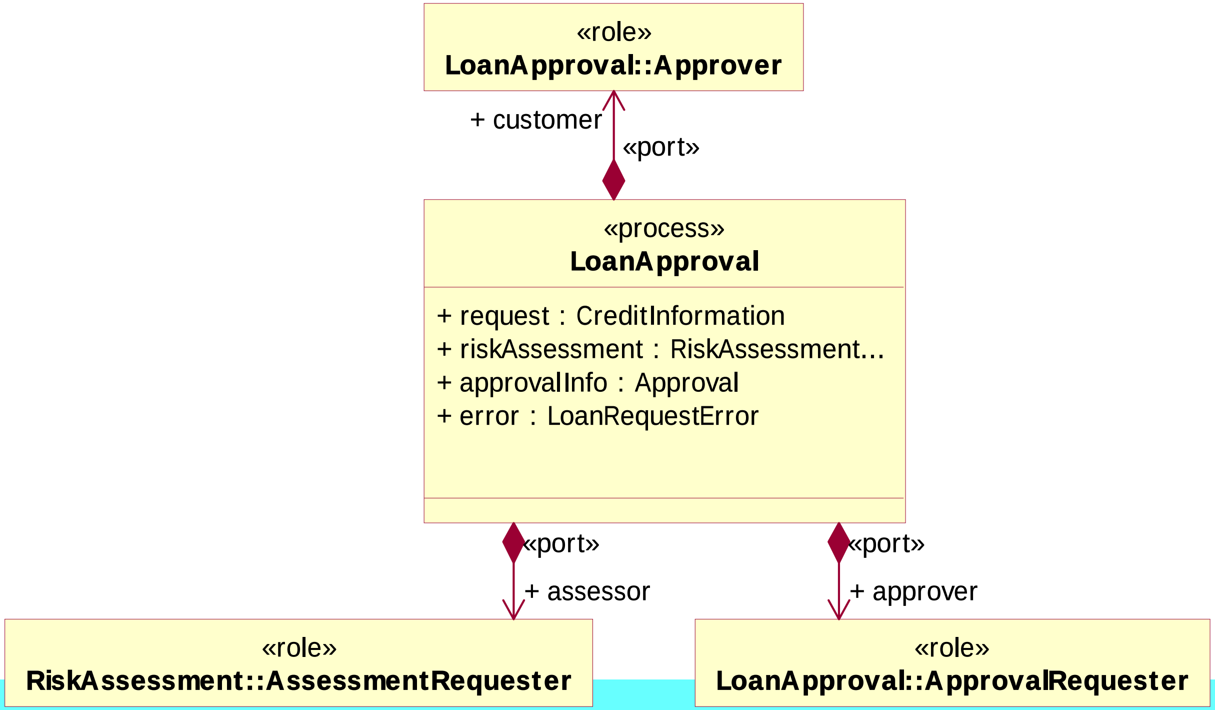
«process»

**LoanApprovalProcess::LoanApproval**

+ request : CreditInformation
+ riskAssessment : RiskAssessmentDetails
+ approvalInfo : Approval
+ error : LoanRequestError

```
<process
    abstractProcess="no"
    containerAccessSerializable="no"
    enableInstanceCompensation="no"
    name="LoanApproval"
    suppressJoinFailure="yes"
    targetNamespace="http://www.bpel-examples.ibm.com/
LoanApproval/LoanApprovalProcess/LoanApproval.bpel">
   ...
   <containers>
        <container messageType="LoanDefinitions:CreditInformation"
                   name="request"/>
        <container messageType="LoanAssessor:RiskAssessmentDetails"
                   name="riskAssessment"/>
        <container messageType="LoanApprover:Approval"
                   name="approvalInfo"/>
        <container messageType="LoanDefinitions:LoanRequestError"
                   name="error"/>
    </containers>
   …
</process>
```

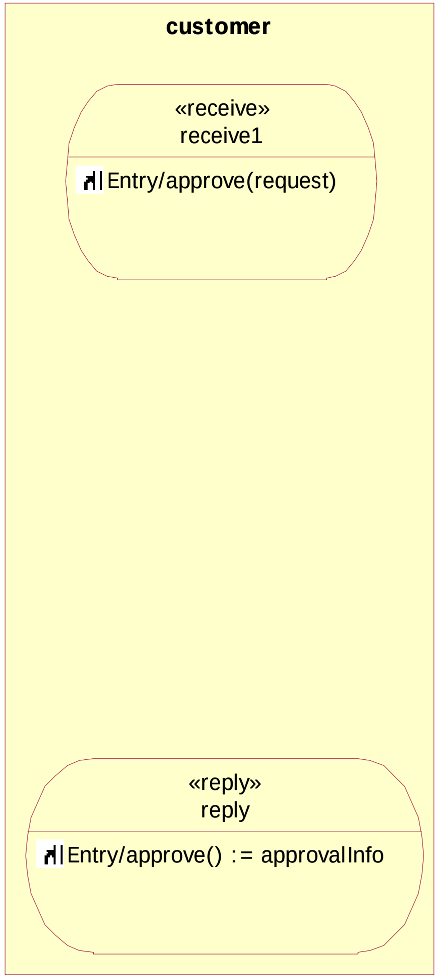
# Mapping: Partners


«role»
LoanApproval::Approver
+ customer
«port»
«process»
LoanApproval
+ request : CreditInformation
+ riskAssessment : RiskAssessment...
+ approvalInfo : Approval
+ error : LoanRequestError
«port»
+ assessor
«port»
+ approver
«role»
RiskAssessment::AssessmentRequester
«role»
LoanApproval::ApprovalRequester

```
   <partners>
         <partner myRole="Approver" name="customer"
                  serviceLinkType="LoanApprovalProtocols:LoanApproval"/>
         <partner name="approver" partnerRole="Approver"
                  serviceLinkType="LoanApprovalProtocols:LoanApproval"/>
         <partner name="assessor" partnerRole="Assessor"
                  serviceLinkType="LoanApprovalProtocols:RiskAssessment"/>
     </partners>
```

# Mapping: Receive/Reply Activities

```
<receive container="request" createInstance="yes"
         name="receive1" operation="approve"
   partner="customer"
         portType="LoanApprover:LoanApproval"
   suppressJoinFailure="no">
   <source linkName="receive1_to_invokeApprover"
        transitionCondition="bpws:getContainerData
   ('request','amount') &gt;= 10000"/>
   <source linkName="receive1_to_invokeAssessor"
        transitionCondition="bpws:getContainerData
   ('request','amount') &lt; 10000"/>
</receive>
    …
<reply container="approvalInfo" name="reply"
   operation="approve"
       partner="customer"
   portType="LoanApprover:LoanApproval"
        suppressJoinFailure="no">
   <target linkName="assign_to_reply"/>
   <target linkName="invokeApprover_to_reply"/>
</reply>
```



**customer**

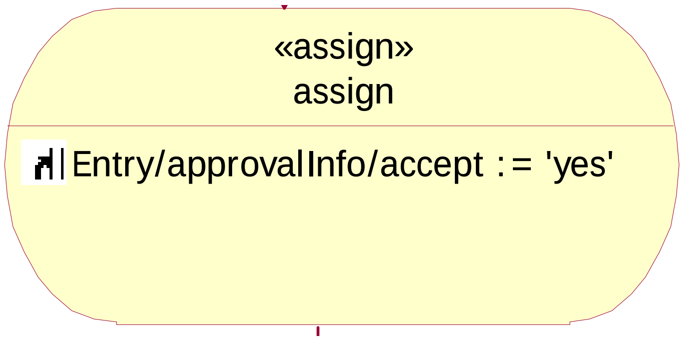
«receive»
receive1

Entry/approve(request)

«reply»
reply

Entry/approve() := approvalInfo

# Mapping: Assign Activities



«assign»
assign

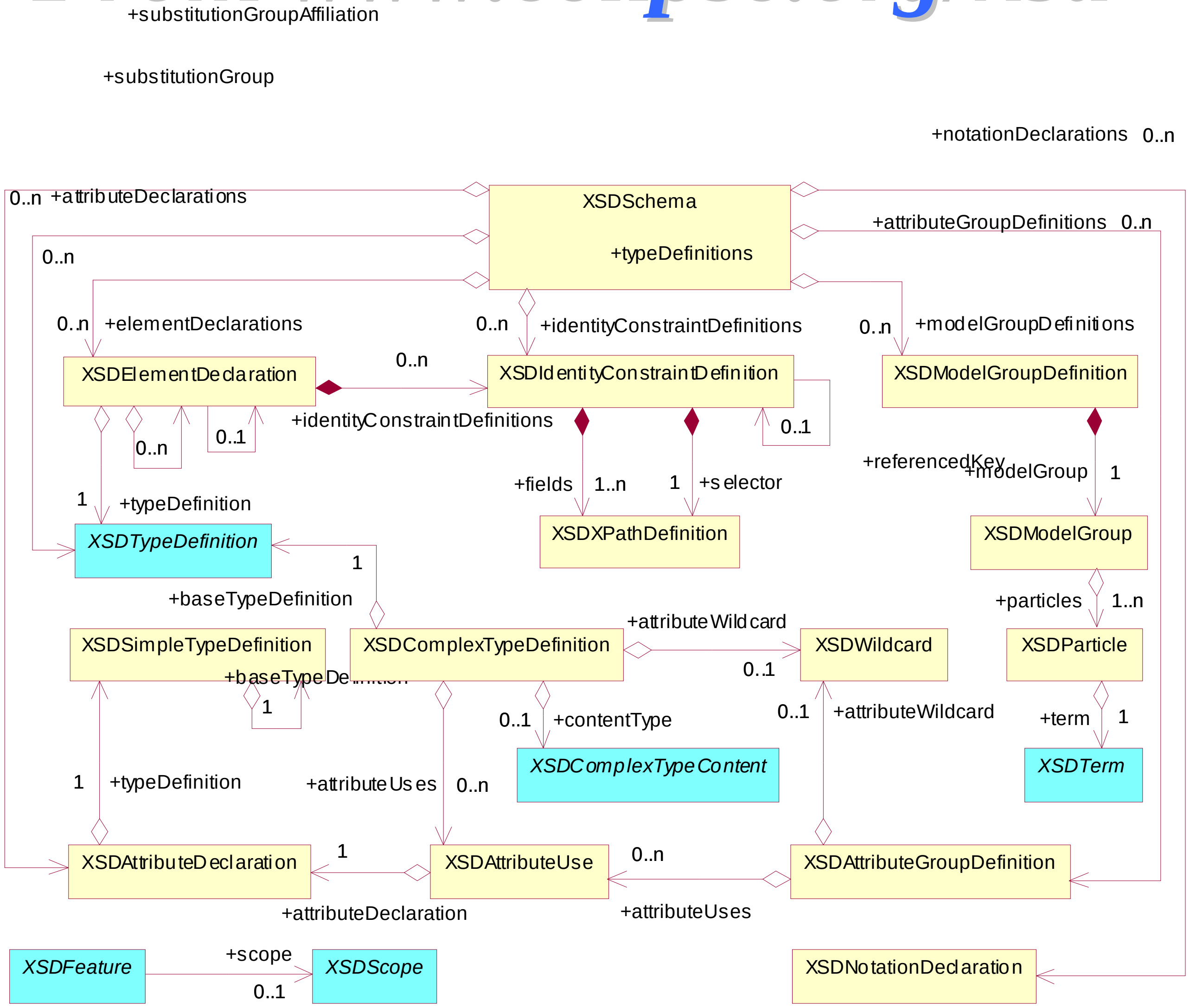
Entry/approvalInfo/accept := 'yes'

```
<assign name="assign">
     <copy>
         <from expression="'yes'"/>
         <to container="approvalInfo" part="accept"/>
     </copy>
 </assign>
```

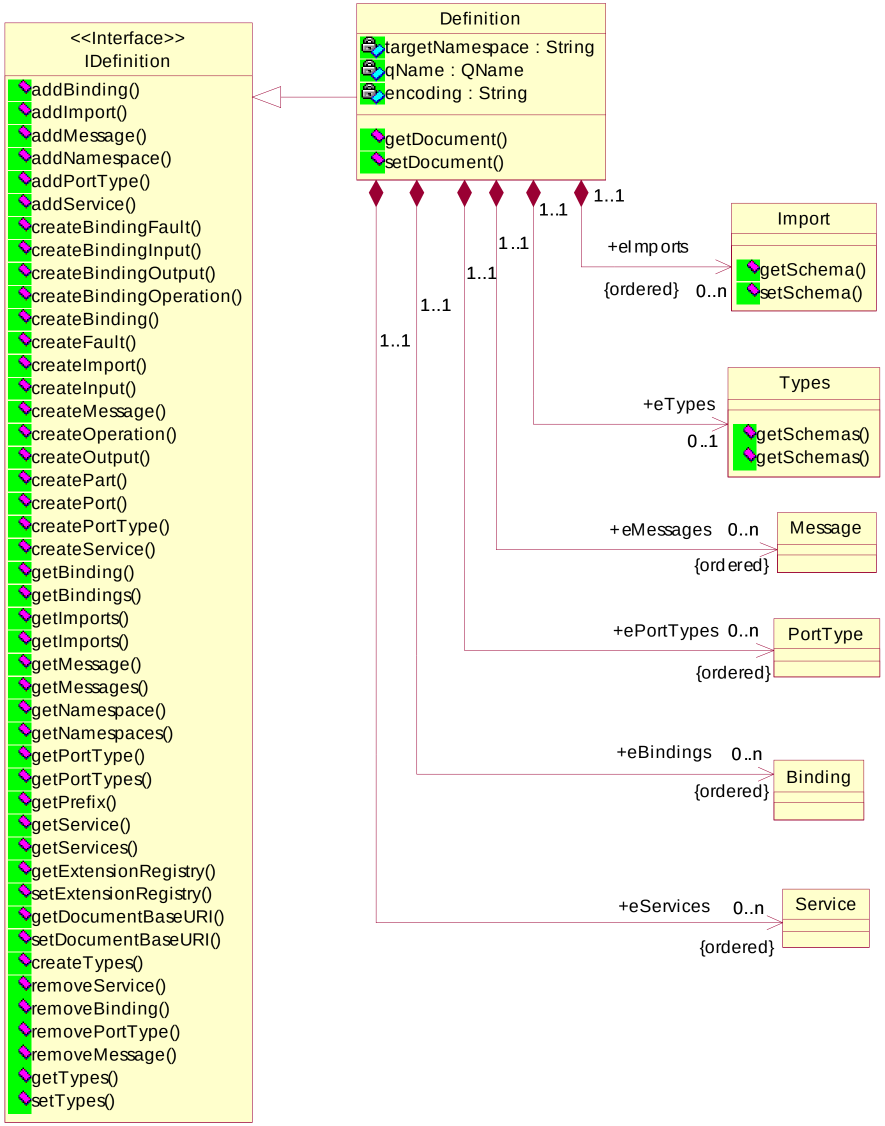
# *Using MDA to execute the Mapping*

- Start with EMF models of UML, XML Schema, WSDL, and BPEL
- Use EMF to generate Java APIs for all the EMF models
  - Each model uses a serializer for its preferred external representation
- Create an EMF model of the mapping between UML and the Web Services components
- Use EMF to generate an API for the model-to-model mapping
- Implement the methods to map between source and target models
- Can be deployed as an Eclipse builder, no user input required to do the mappings

# XML Schema model
# From www.eclipse.org/xsd



# WSDL Model – From WSAD



<<Interface>>
IDefinition

- addBinding()
- addImport()
- addMessage()
- addNamespace()
- addPortType()
- addService()
- createBindingFault()
- createBindingInput()
- createBindingOutput()
- createBindingOperation()
- createBinding()
- createFault()
- createImport()
- createInput()
- createMessage()
- createOperation()
- createOutput()
- createPart()
- createPort()
- createPortType()
- createService()
- getBinding()
- getBindings()
- getImports()
- getImports()
- getMessage()
- getMessages()
- getNamespace()
- getNamespaces()
- getPortType()
- getPortTypes()
- getPrefix()
- getService()
- getServices()
- getExtensionRegistry()
- setExtensionRegistry()
- getDocumentBaseURI()
- setDocumentBaseURI()
- createTypes()
- removeService()
- removeBinding()
- removePortType()
- removeMessage()
- getTypes()
- setTypes()

Definition

- targetNamespace : String
- qName : QName
- encoding : String

- getDocument()
- setDocument()

1..1 1..1 1..1 1..1 1..1 1..1

+eImports {ordered} 0..n

Import

- getSchema()
- setSchema()

+eTypes 0..1

Types

- getSchemas()
- getSchemas()

+eMessages 0..n {ordered}

Message

+ePortTypes 0..n {ordered}

PortType

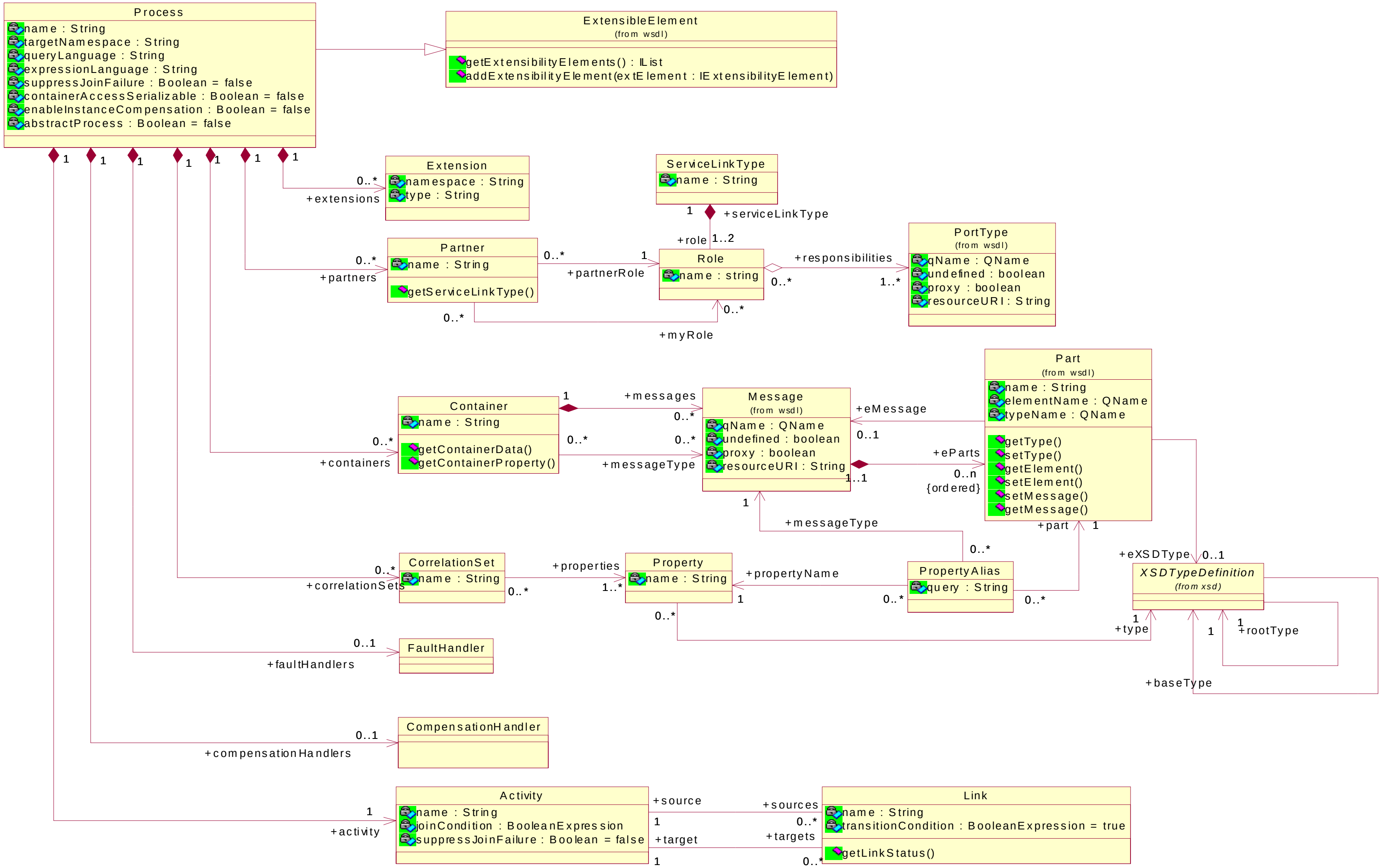
+eBindings 0..n {ordered}

Binding

+eServices 0..n {ordered}

Service

# Draft BPEL Model



Process
name : String
targetNamespace : String
queryLanguage : String
expressionLanguage : String
suppressJoinFailure : Boolean = false
containerAccessSerializable : Boolean = false
enableInstanceCompensation : Boolean = false
abstractProcess : Boolean = false

ExtensibleElement
(from wsdl)
getExtensibilityElements() : IList
addExtensibilityElement(extElement : IExtensibilityElement)

Extension
namespace : String
type : String

ServiceLinkType
name : String

Partner
name : String
getServiceLinkType()

Role
name : string

PortType
(from wsdl)
qName : QName
undefined : boolean
proxy : boolean
resourceURI : String

Part
(from wsdl)
name : String
elementName : QName
typeName : QName
getType()
setType()
getElement()
setElement()
setMessage()
getMessage()

Container
name : String
getContainerData()
getContainerProperty()

Message
(from wsdl)
qName : QName
undefined : boolean
proxy : boolean
resourceURI : String

CorrelationSet
name : String

Property
name : String

PropertyAlias
query : String

XSDTypeDefinition
(from xsd)

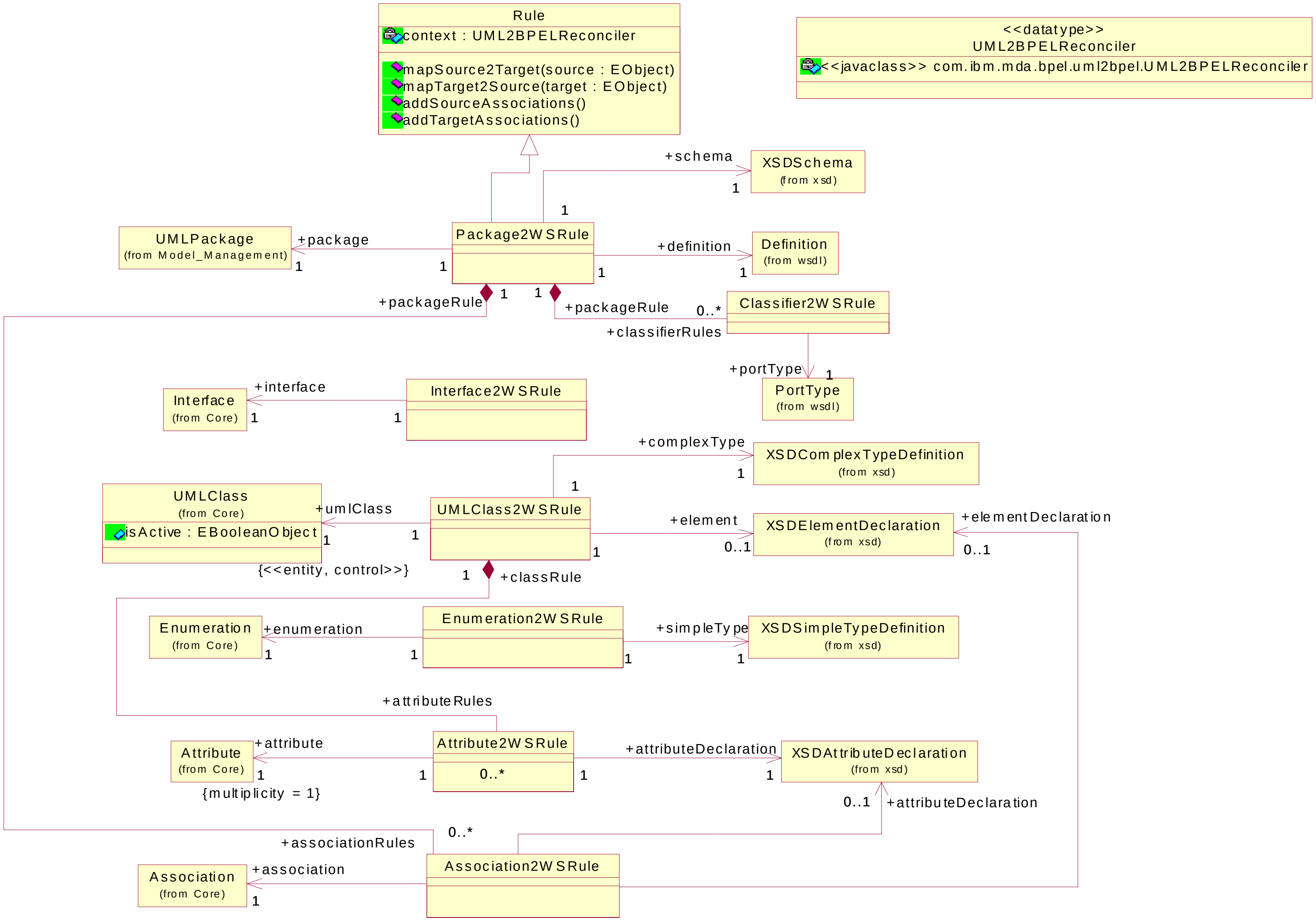
FaultHandler

CompensationHandler

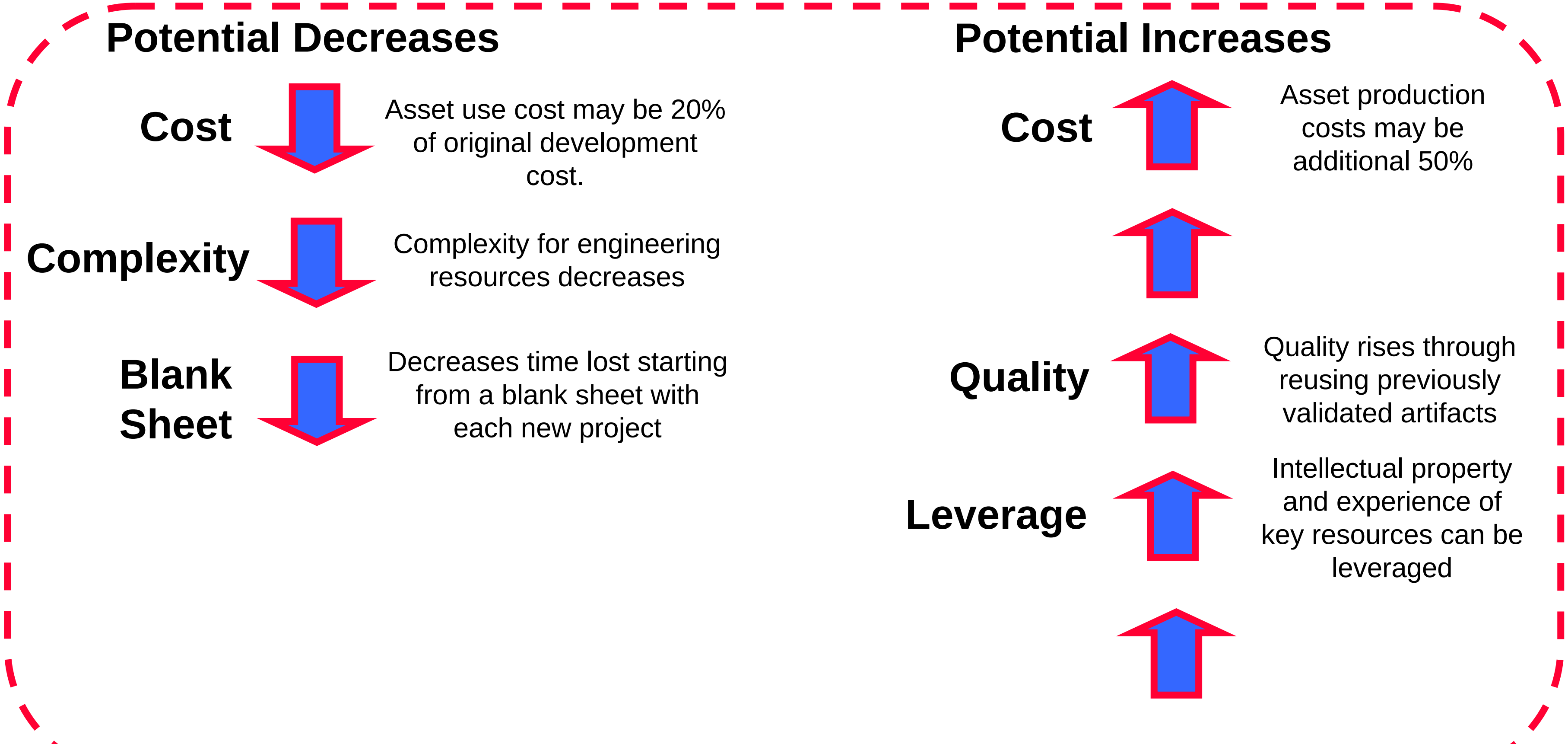
Activity
name : String
joinCondition : BooleanExpression
suppressJoinFailure : Boolean = false

Link
name : String
transitionCondition : BooleanExpression = true
getLinkStatus()

# UML to BPEL Mapping model



# Motivation for Reuse : Asset Based Development



## Potential Decreases

**Cost** – Asset use cost may be 20% of original development cost.

**Complexity** – Complexity for engineering resources decreases

**Blank Sheet** – Decreases time lost starting from a blank sheet with each new project

## Potential Increases

**Cost** – Asset production costs may be additional 50%

**Quality** – Quality rises through reusing previously validated artifacts

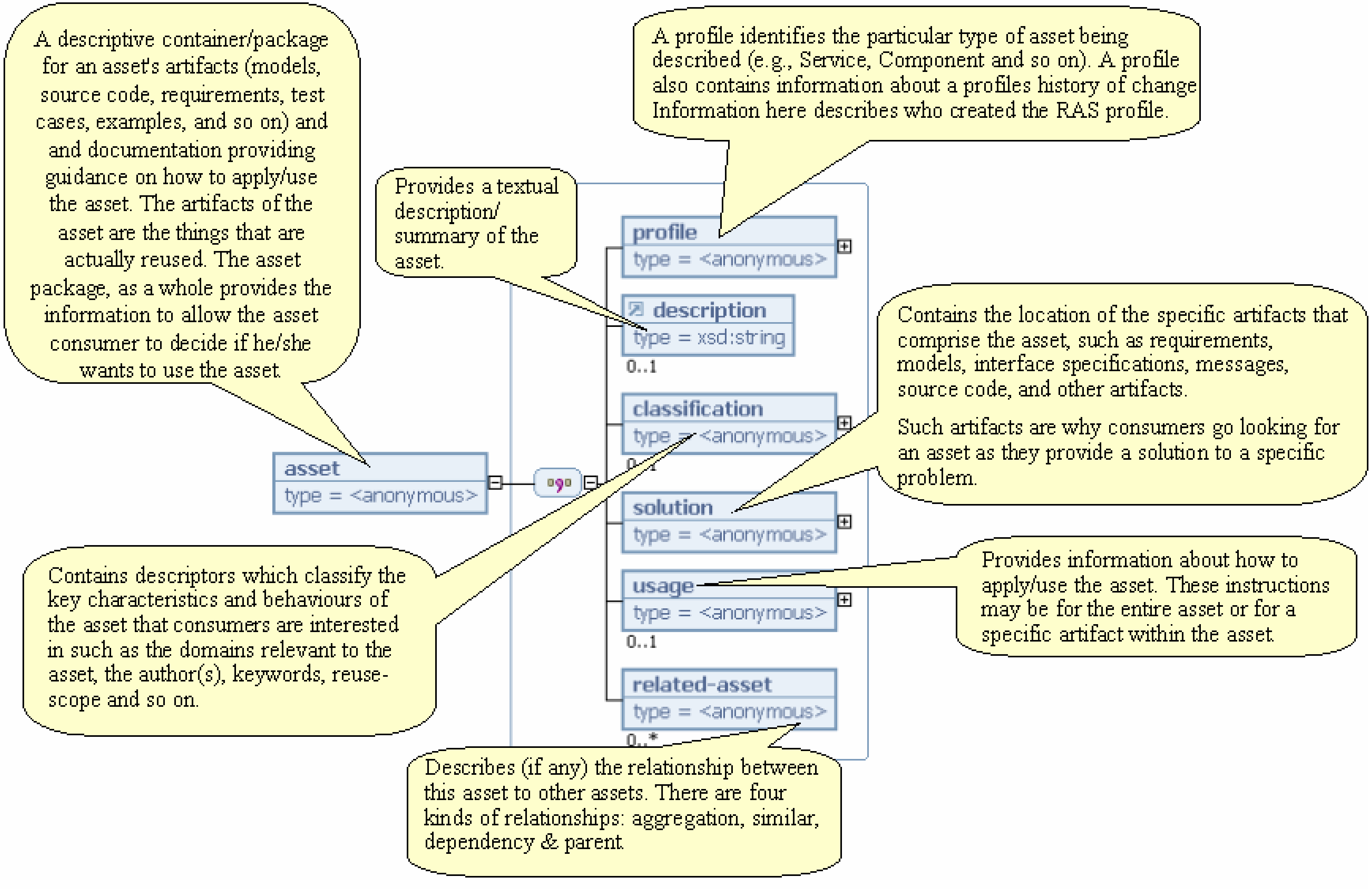
**Leverage** – Intellectual property and experience of key resources can be leveraged

- Took a pragmatic approach to reuse
  - Approach the specification from bottom up
  - Didn't want to build the "mother of all repositories"

# What is Reusable Asset Specification?

- A structure for describing the characteristics of a reusable software asset by using meta-data
- The meta-data used to describe a reusable software asset is based on an underlying XML schema referred to as an asset profile
- **Question**: Why would the meta-data of a software asset be packaged as a XML document?
  - **Answer**: To enable the communication and shipping (exchange) of the software assets meta-data across interested parties e.g.  From the supplier of the software asset to the consumer of the software asset
- Based on the meta-data, a potential consumer of the software asset can quickly determine if the asset will be suitable in satisfying their business requirements
- **Question**: Is the asset’s meta always represented in XML?
  - **Answer**: Not necessarily, the asset’s meta-data can be represented in any database, but when the need arises to have a formal communication/exchange of the assets meta-data, the XML document can be produced in a RAS compliant format

# RAS XML Schema Overview



A descriptive container/package for an asset's artifacts (models, source code, requirements, test cases, examples, and so on) and and documentation providing guidance on how to apply/use the asset. The artifacts of the asset are the things that are actually reused. The asset package, as a whole provides the information to allow the asset consumer to decide if he/she wants to use the asset.

Provides a textual description/ summary of the asset.

A profile identifies the particular type of asset being described (e.g., Service, Component and so on). A profile also contains information about a profiles history of change Information here describes who created the RAS profile.

Contains the location of the specific artifacts that comprise the asset, such as requirements, models, interface specifications, messages, source code, and other artifacts.

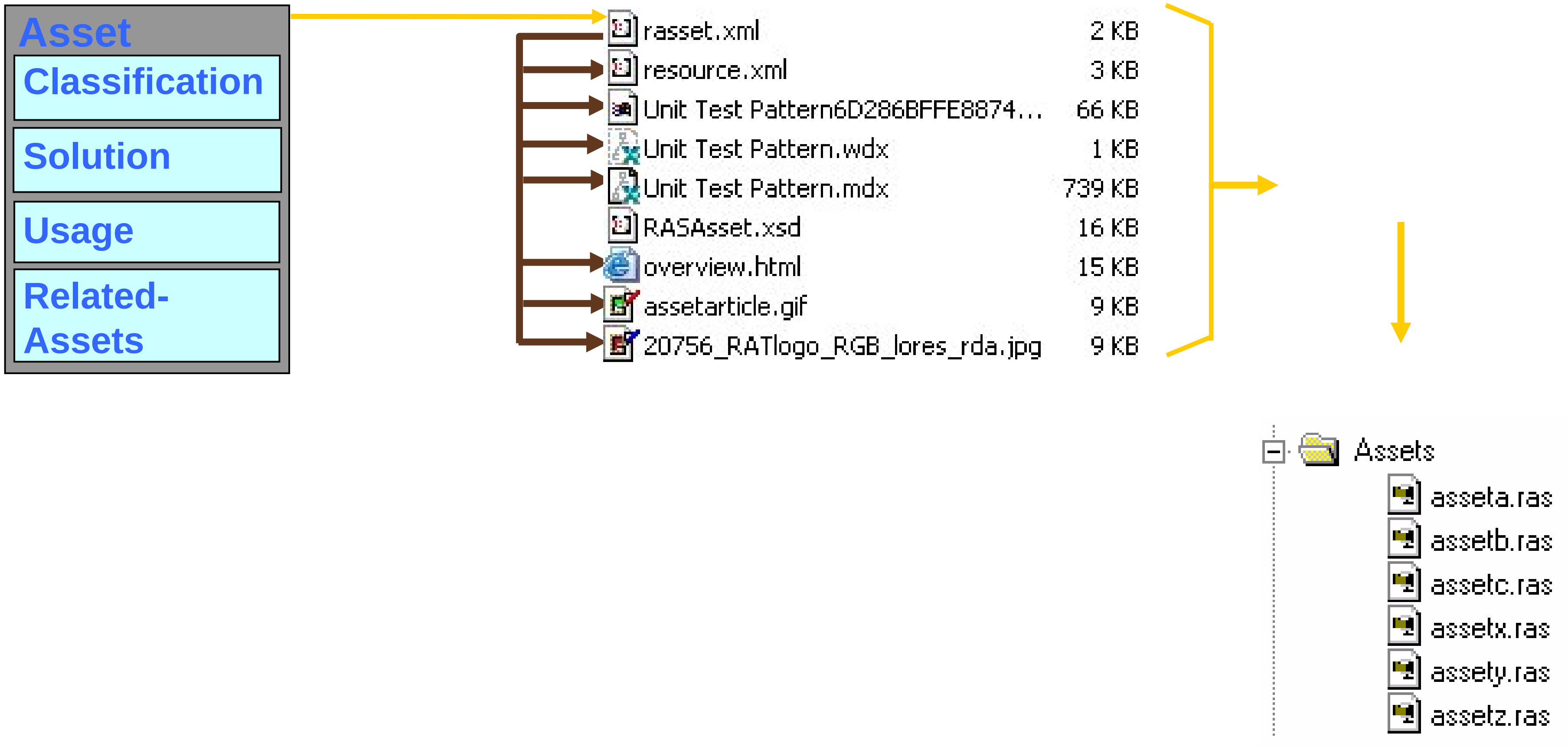
Such artifacts are why consumers go looking for an asset as they provide a solution to a specific problem.

Contains descriptors which classify the key characteristics and behaviours of the asset that consumers are interested in such as the domains relevant to the asset, the author(s), keywords, reuse-scope and so on.

Provides information about how to apply/use the asset. These instructions may be for the entire asset or for a specific artifact within the asset.

Describes (if any) the relationship between this asset to other assets. There are four kinds of relationships: aggregation, similar, dependency & parent.

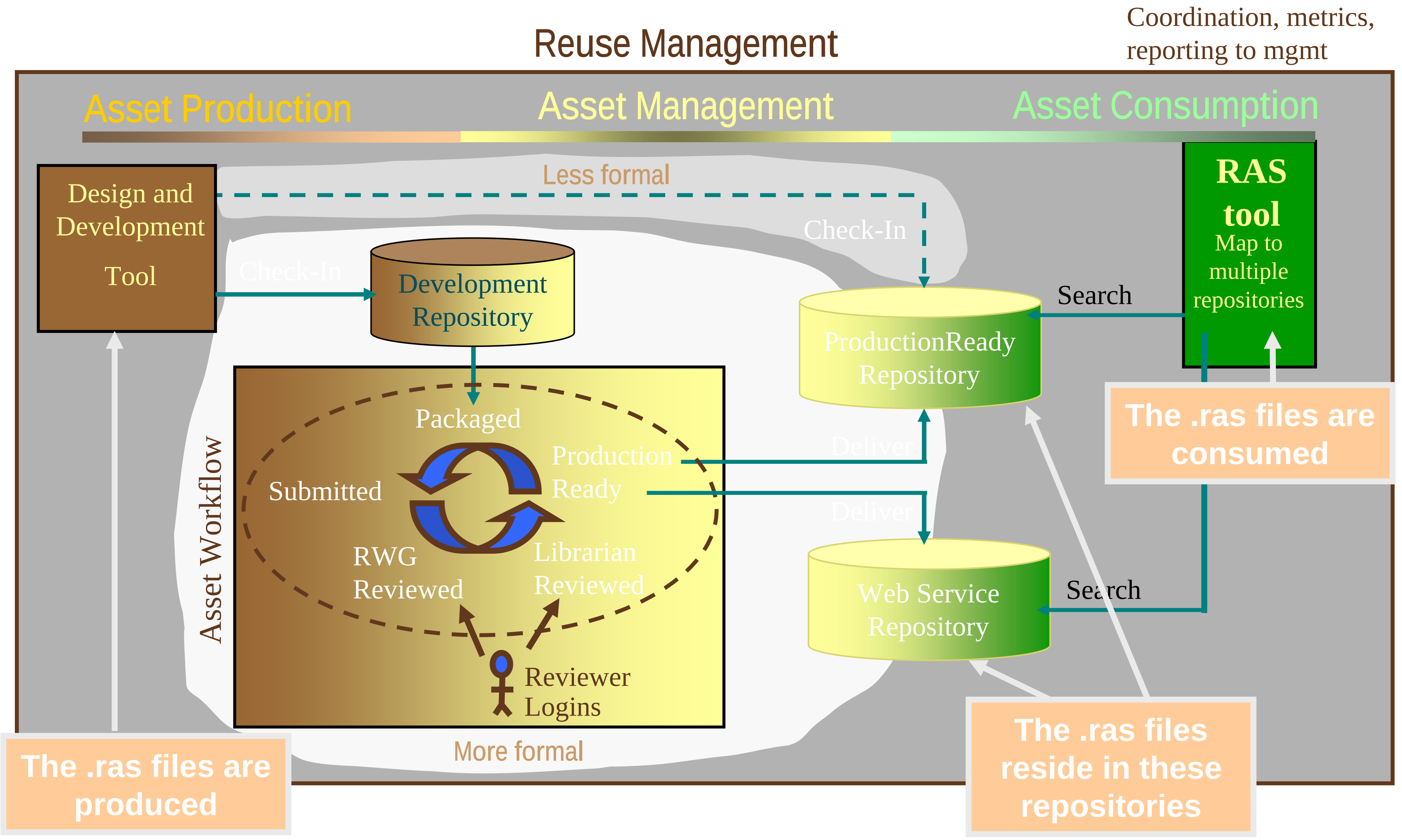
# How is RAS used (1)



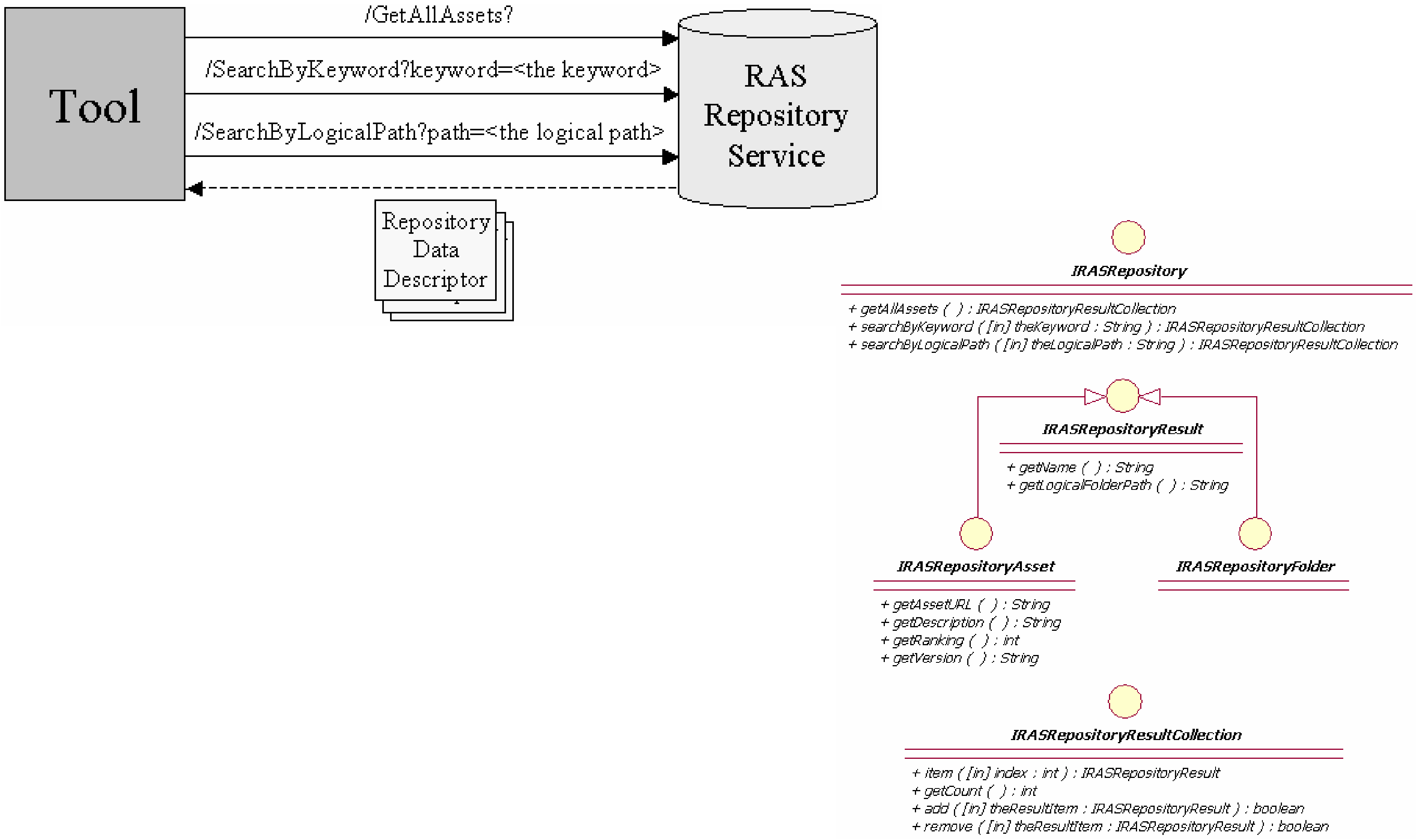
| | |
|---|---|
| rasset.xml | 2 KB |
| resource.xml | 3 KB |
| Unit Test Pattern6D286BFFE8874... | 66 KB |
| Unit Test Pattern.wdx | 1 KB |
| Unit Test Pattern.mdx | 739 KB |
| RASAsset.xsd | 16 KB |
| overview.html | 15 KB |
| assetarticle.gif | 9 KB |
| 20756_RATlogo_RGB_lores_rda.jpg | 9 KB |

Assets
- asseta.ras
- assetb.ras
- assetc.ras
- assetx.ras
- assety.ras
- assetz.ras

# How is RAS used (2)



# RAS Repository Service



Tool

/GetAllAssets?

/SearchByKeyword?keyword=<the keyword>

/SearchByLogicalPath?path=<the logical path>

RAS Repository Service

Repository Data Descriptor

*IRASRepository*

*+ getAllAssets ( ) : IRASRepositoryResultCollection*
*+ searchByKeyword ( [in] theKeyword : String ) : IRASRepositoryResultCollection*
*+ searchByLogicalPath ( [in] theLogicalPath : String ) : IRASRepositoryResultCollection*

*IRASRepositoryResult*

*+ getName ( ) : String*
*+ getLogicalFolderPath ( ) : String*

*IRASRepositoryAsset*

*+ getAssetURL ( ) : String*
*+ getDescription ( ) : String*
*+ getRanking ( ) : int*
*+ getVersion ( ) : String*

*IRASRepositoryFolder*

*IRASRepositoryResultCollection*

*+ item ( [in] index : int ) : IRASRepositoryResult*
*+ getCount ( ) : int*
*+ add ( [in] theResultItem : IRASRepositoryResult ) : boolean*
*+ remove ( [in] theResultItem : IRASRepositoryResult ) : boolean*

# Who is using RAS?

- Customers
  - Volvo, USPTO, Visa, Charles Schwab, …
- Tool Vendors
  - FlashLine
  - LogicLibrary
  - IBM
  - OSTnet

# Modeling for Web Services - Recap

- UML, MOF and XMI can be used as-is to model & manage Web Services
  - Web Service Operations (WSDL Operations) are UML Operations
  - Groups of Web Service structures Requests (WSDL Ports) as UML Interfaces or Classes
  - XML Schemas using UML Class Diagrams
  - Web Services data structures using UML Class Diagrams
  - Web Services flow (E.g.:BPEL4WS) using UML Activity Diagrams
- But some standard transformations, UML profiles etc, would be useful to enable tools interoperability
  - http://www.omg.org (MOF, XMI, UML and CWM specs)
  - XMI - Mappings from MOF/UML to XML DTD, XML Schema
  - Modeling XML applications with UML by David Carlson
    - http://www.xmlmodeling.com
  - UML to BPEL mappings (UML profile for BPEL?)
  - OMG Business Process Definition Metamodel RFP
  - OMG Business Rules RFP Drafts : Rule Expression Language, Rule Management…

# Wrap Up

- Most customers have and will continue to have components and information/data  from multiple sources and formats that need integration
- Web Services promise to be a key enabler for application integration and business process integration
  - Embraced by all industry players
  - Pragmatic loosely coupled integration
  - Promise of ubiquitous availability
  - Modeling and Web Services on a convergence path
- Model Driven Architecture enables architects, designers and developers to use models and metadata with web services and existing technologies with a focus on full life cycle architecture
- Web Services standards (like other standards!)will come and go, but some of the business models and processes will be more lasting (manage and protect your metadata)
- Use models to communicate, understand, analyze and design, use metadata to integrate flexibly within an architetural context

# For More Information

- OMG : www.omg.org
- Web Services : http://www-106.ibm.com/developerworks/webservices/newto
- MDA : www.omg.org/mda
- IBM : www.ibm.com
- IBM Alphaworks : www.ibm.com/alphaworks (UDDI4J, BPWS4J)
- UML : www.omg.org/uml, www.rational.com
- CWM : www.cwmforum.org, www.omg.org/uml
- W3C : www.w3c.org
- DSTC : www.dstc.edu.au
- Sridhar : siyengar@us.ibm.com
- UML RTF : uml-rtf@omg.org
- MOF RTF : mof-rtf@omg.org
- XMI RTF : xmi-rtf@omg.org
- CWM RTF : cwm-rtf@omg.org
- JMI : http://java.sun.com/aboutJava/communityprocess/jsr-40
- Eclipse : http://www.eclipse.org (/emf, /xsd)