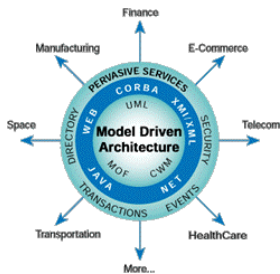


Implementing Model Driven Web Services Architecture using UML, XML, WSDL & BPEL4WS

Web Services for the Integrated Enterprise
Munich, Germany
February 10-13, 2003



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



Presentation Overview

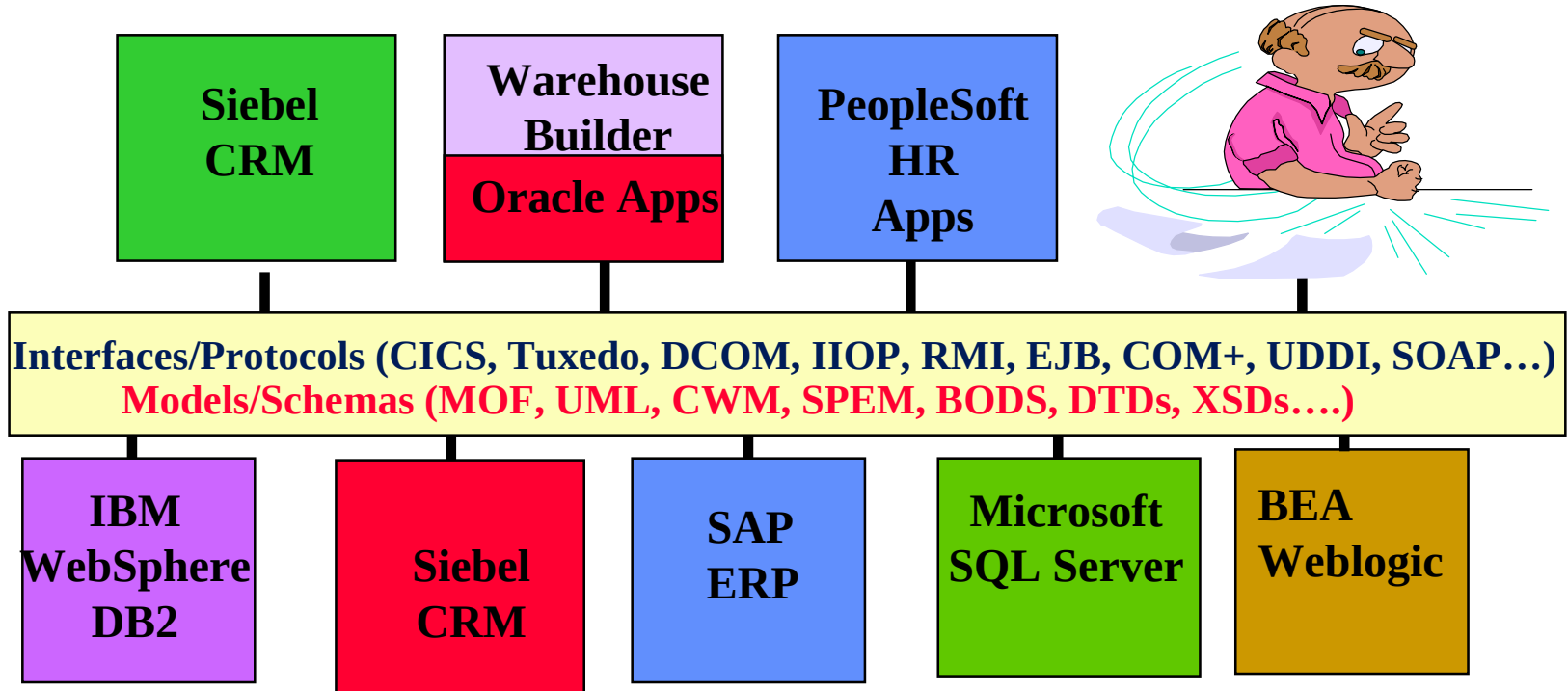
- Introduction - The integration problem
- OMG MDA - Model Driven Architecture Overview
 - What is MDA and why is it important
 - Emerging MDA standards for Model Driven Business Integration
- Web Services Architecture Overview
 - What are Web Services and why are they important
 - Emerging Web Services 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 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

‘Muddleware’ Architect’s Dilemma

Web Services!



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

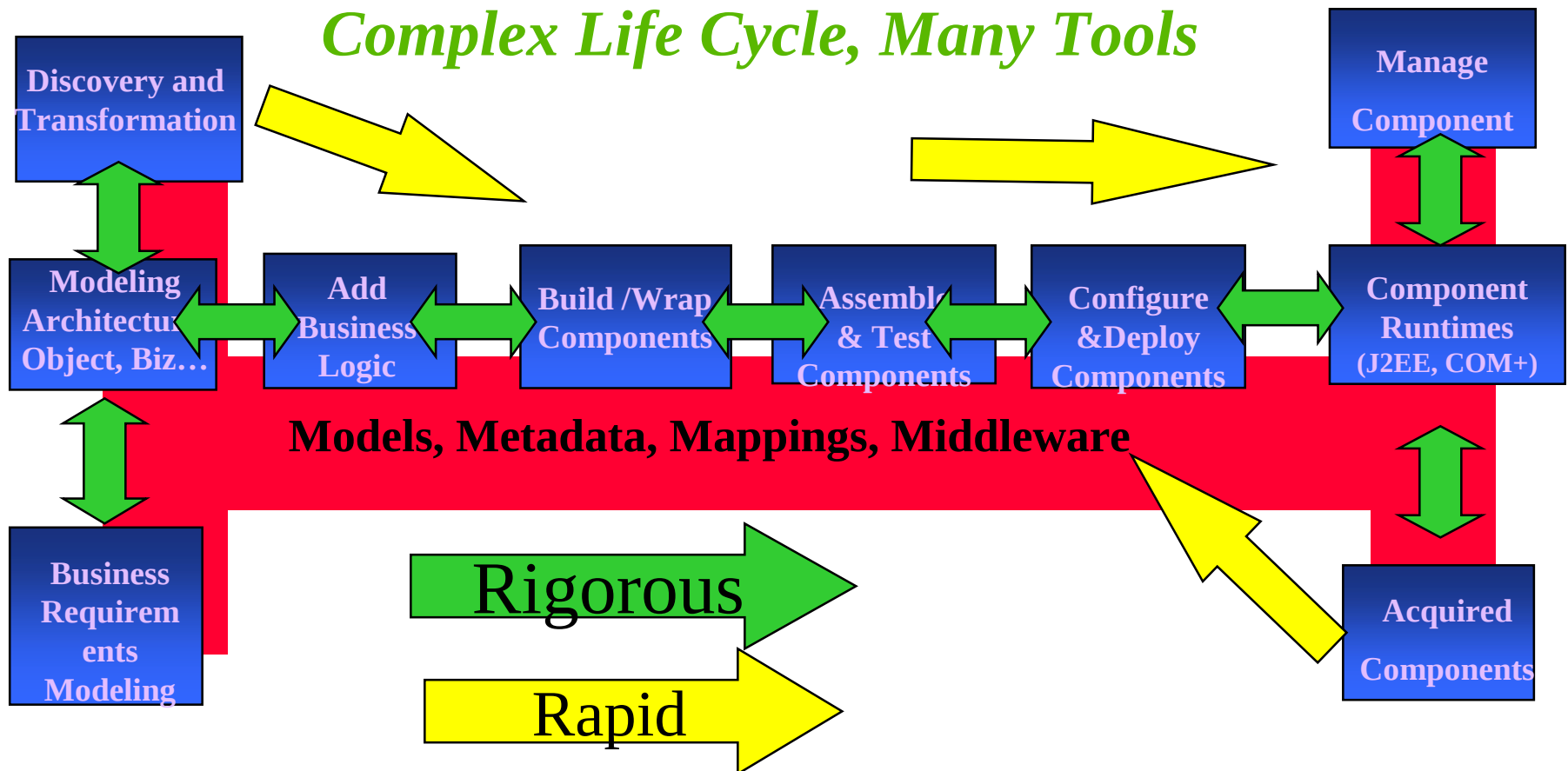
Solving the Integration Problem

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, ebXML...
- 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

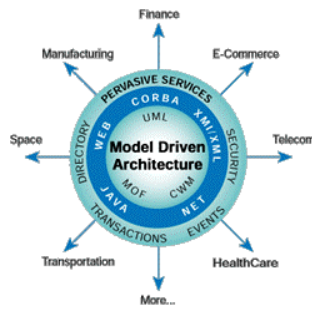
*Architecture Centric, Business driven,
Complex Life Cycle, Many Tools*



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 **M**odels, **M**etadata, **M**appings (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.

Sridhar's Usage

OMG History

- 700+ 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 **IOP** - 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

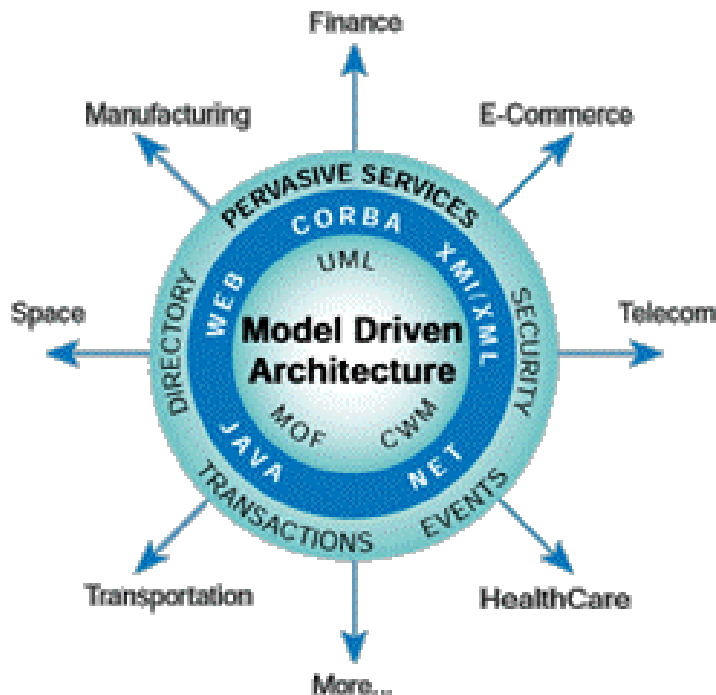
OMG Analysis & Design Platform Task Force Premise

- 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 : $PIM \rightleftharpoons PSM$,
 - 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)

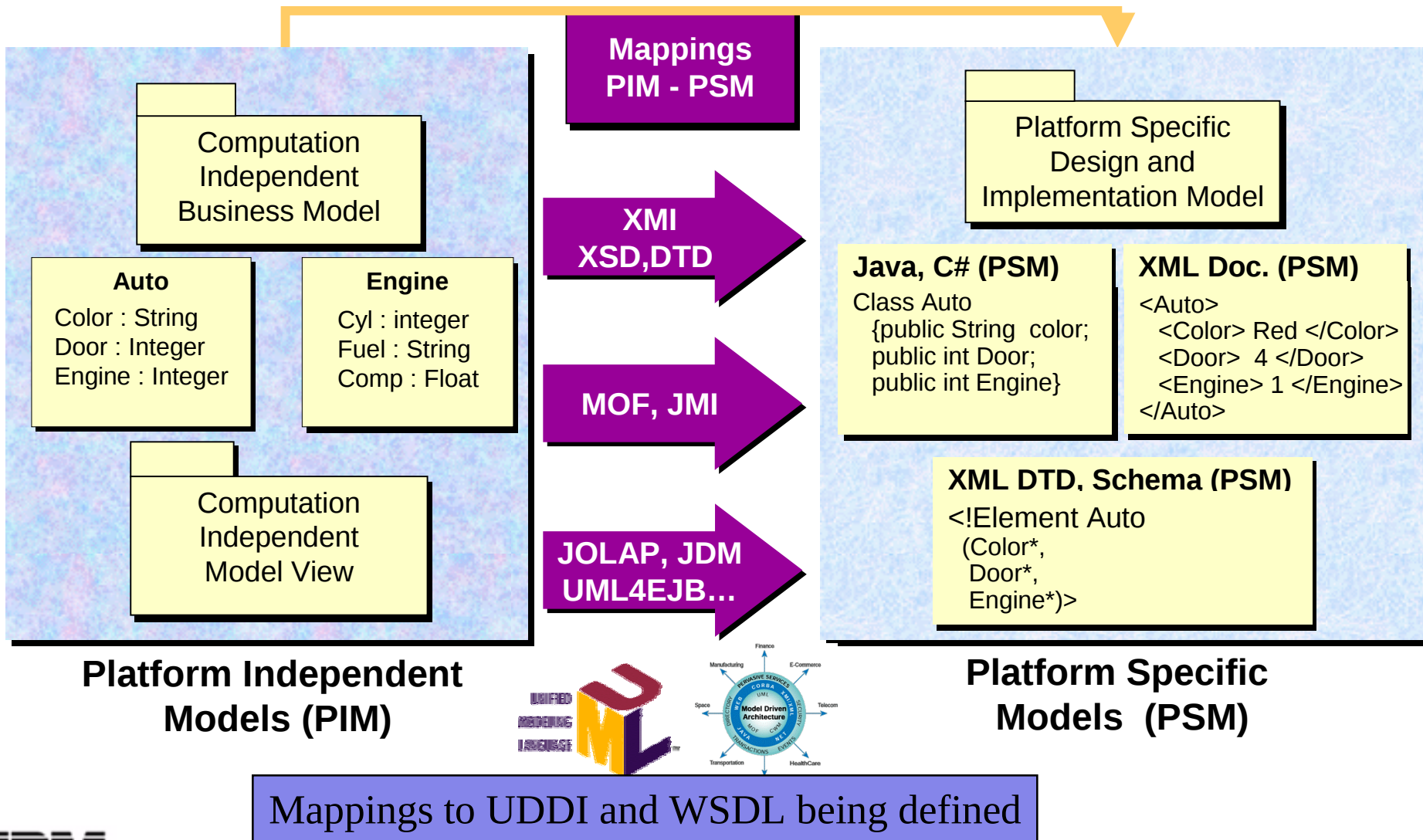
Copyright 2003, IBM Corporation



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 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)



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...)

P
L
A
T
F
O
R
M

A
G
N
O
S
T
I
C

S
P
E
C
I
F
I
C

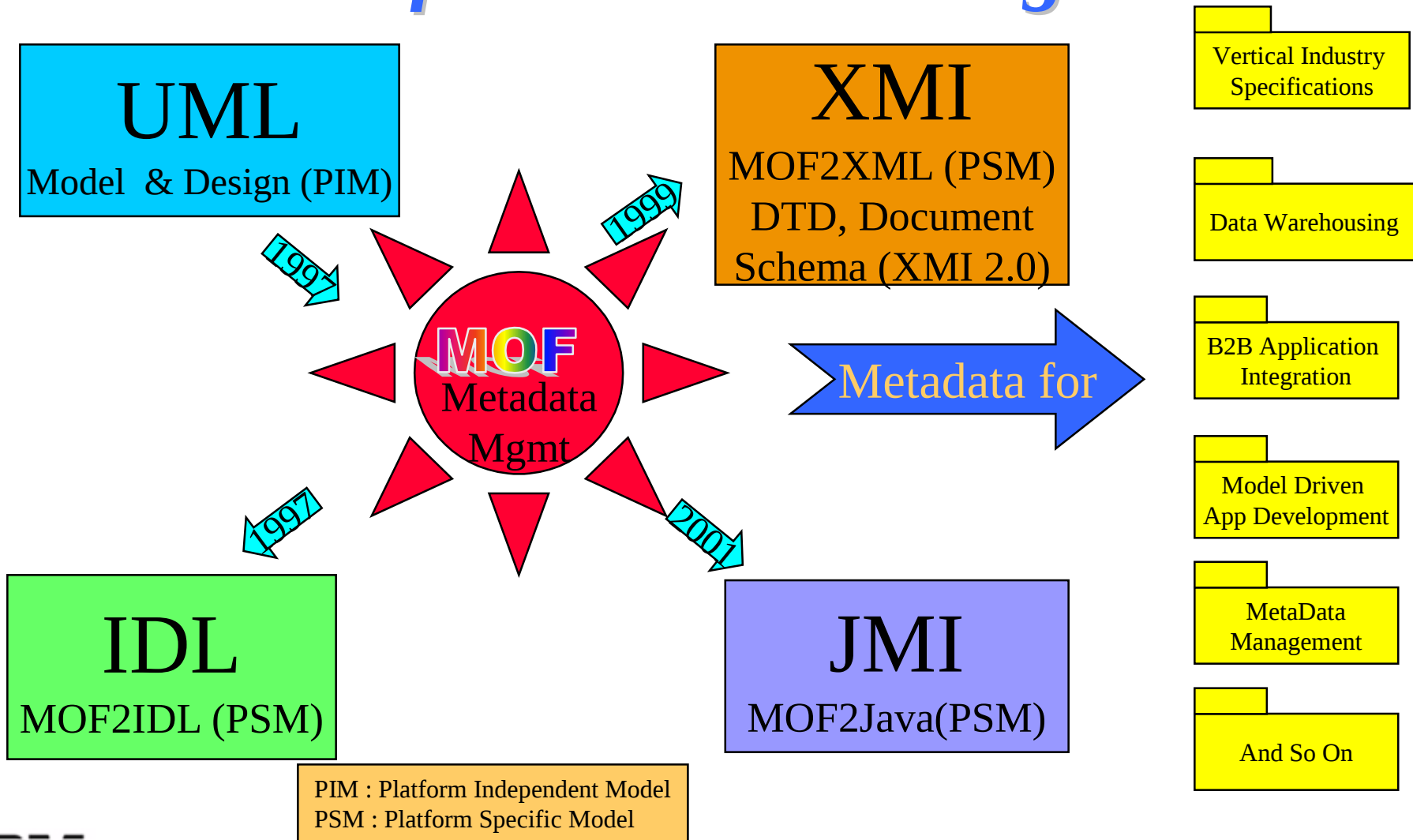
* Opportunity

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
- 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), 'Business Objects' (UML4EDOC)

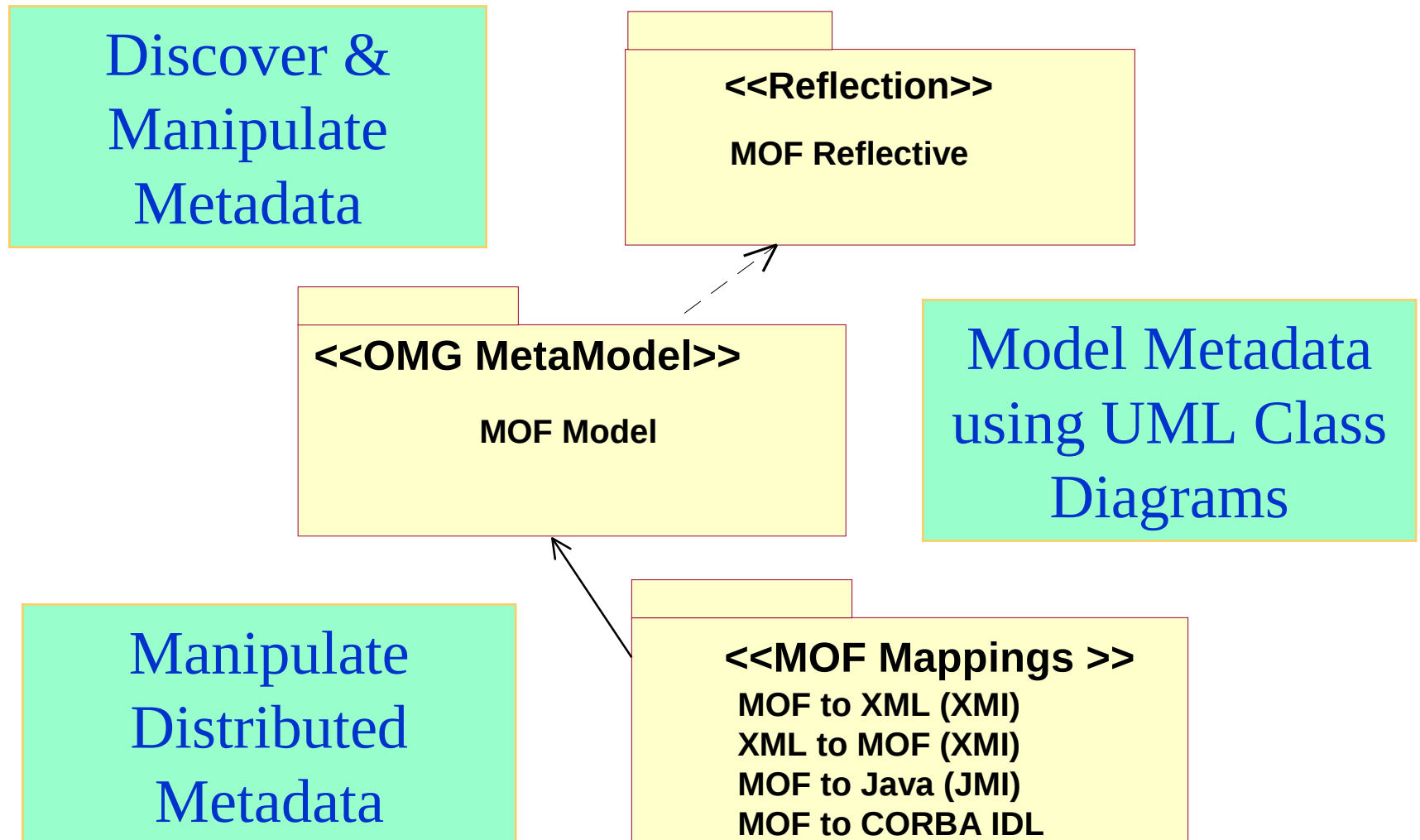
OMG Model Driven Architecture

MDA for Metadata Integration

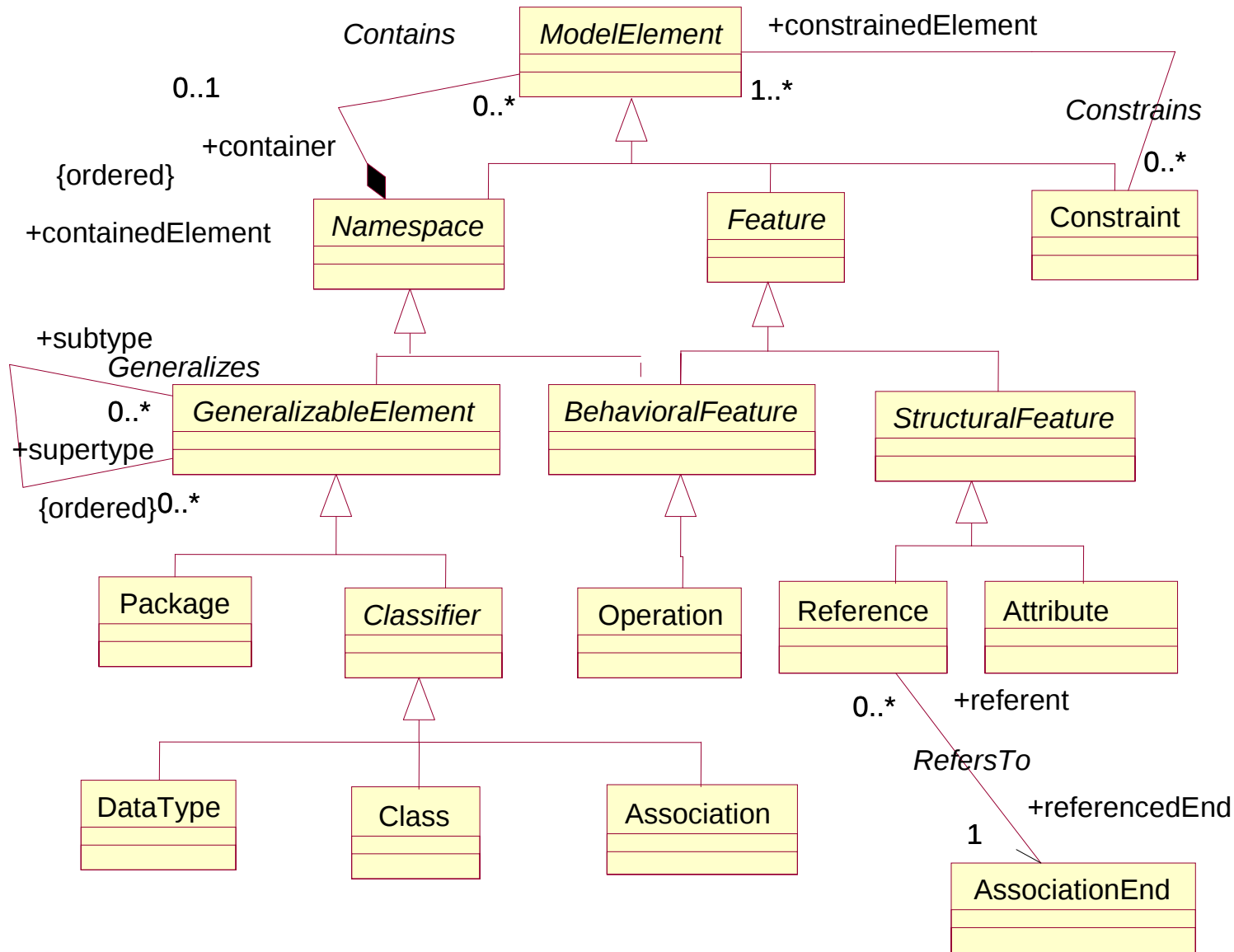


Meta Object Facility Overview

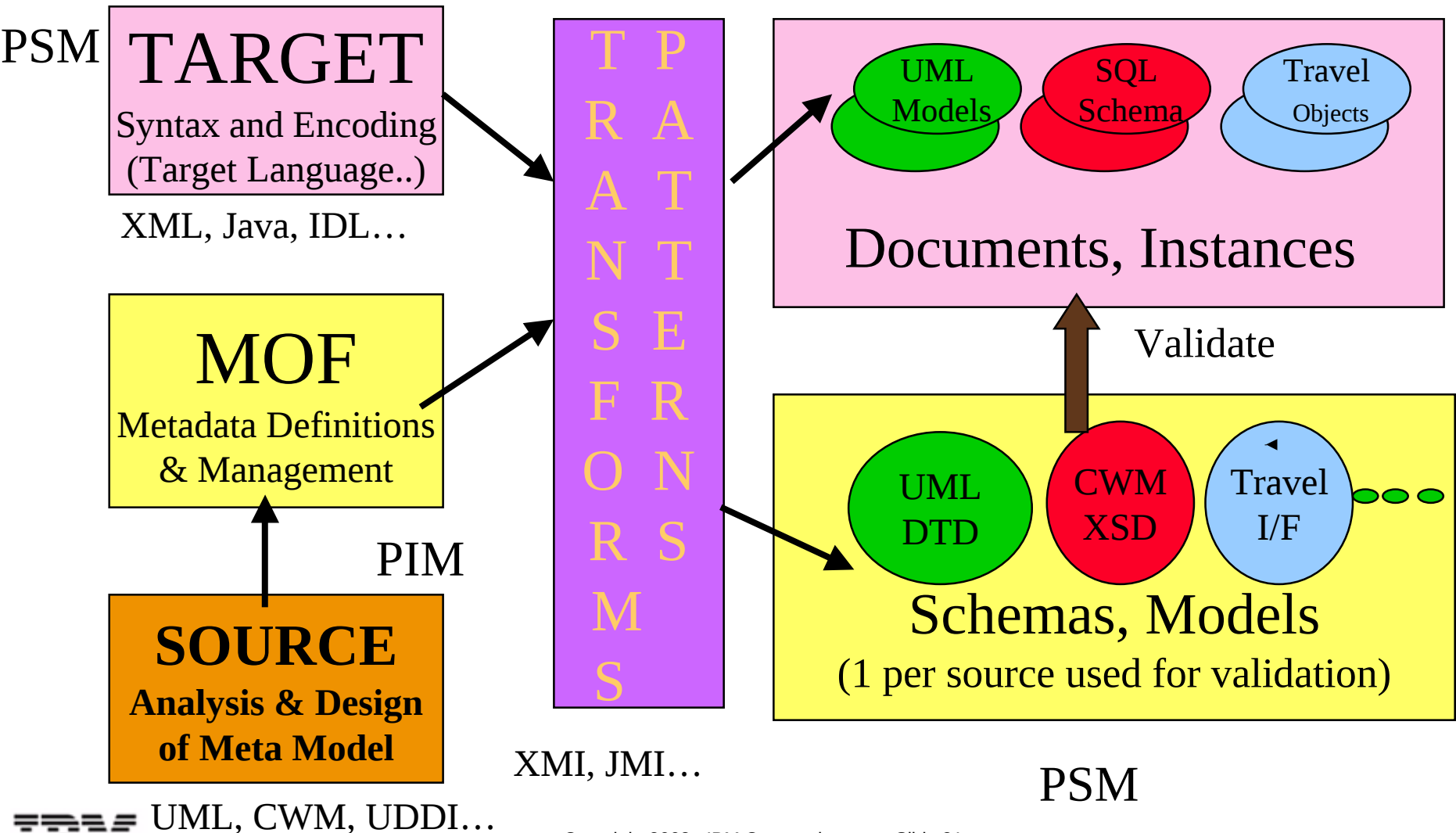
Current Revision : MOF 1.4



OMG MOF 1.4 Model Subset

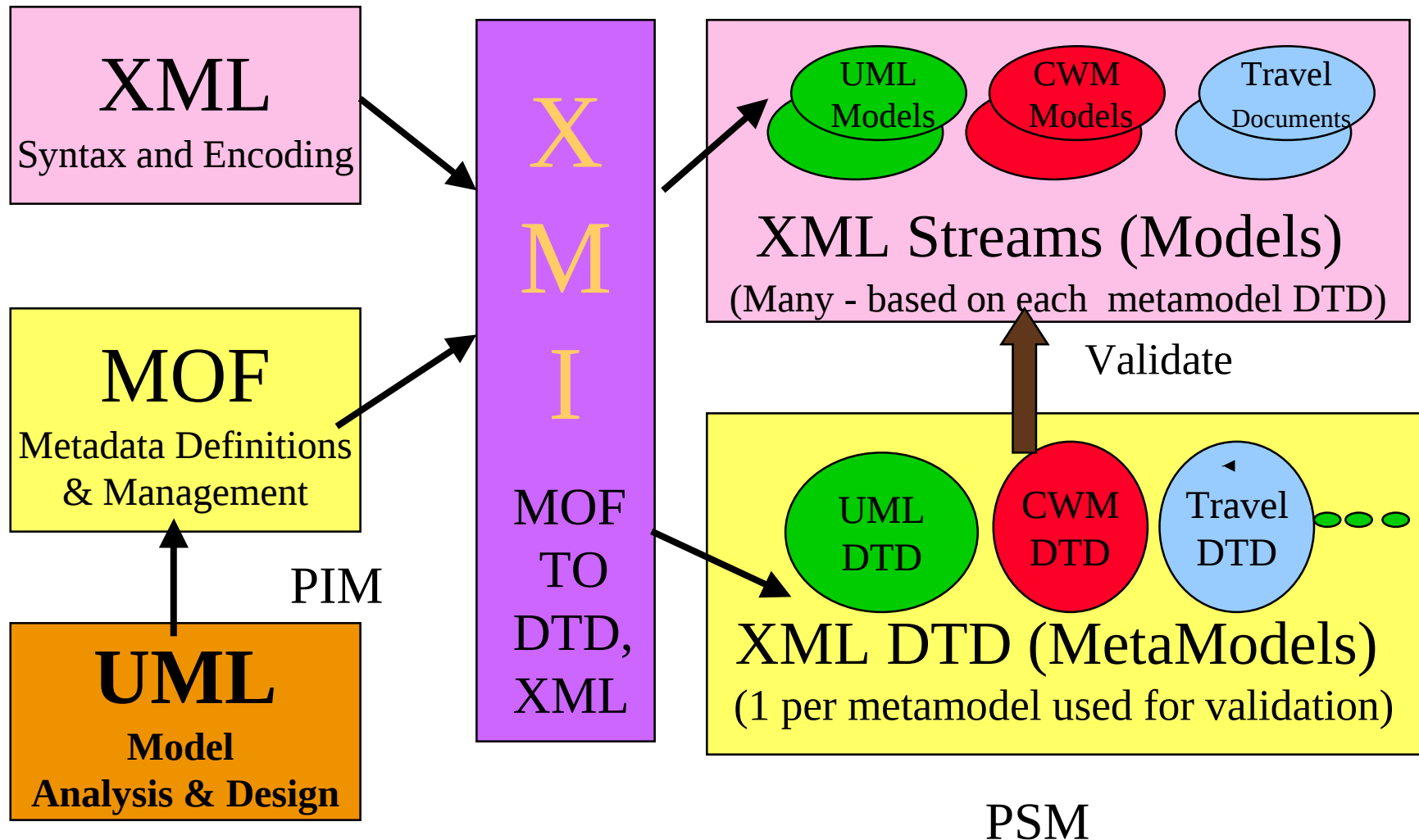


Applying the MDA Architectural Style for Metadata Management



Transform Models to DTD

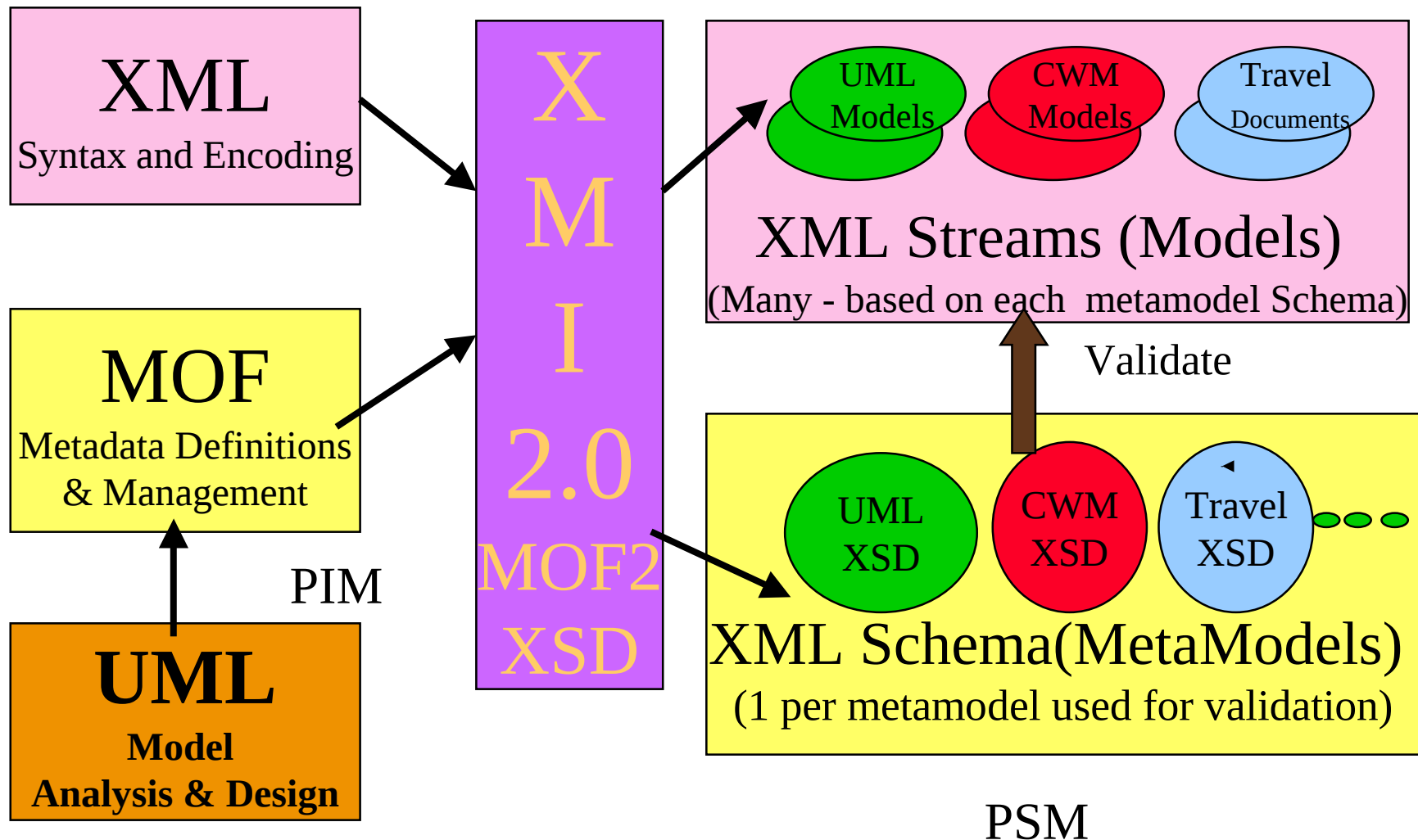
XMI 1.1 or XMI 1.2 (1999, 2001)



PSM

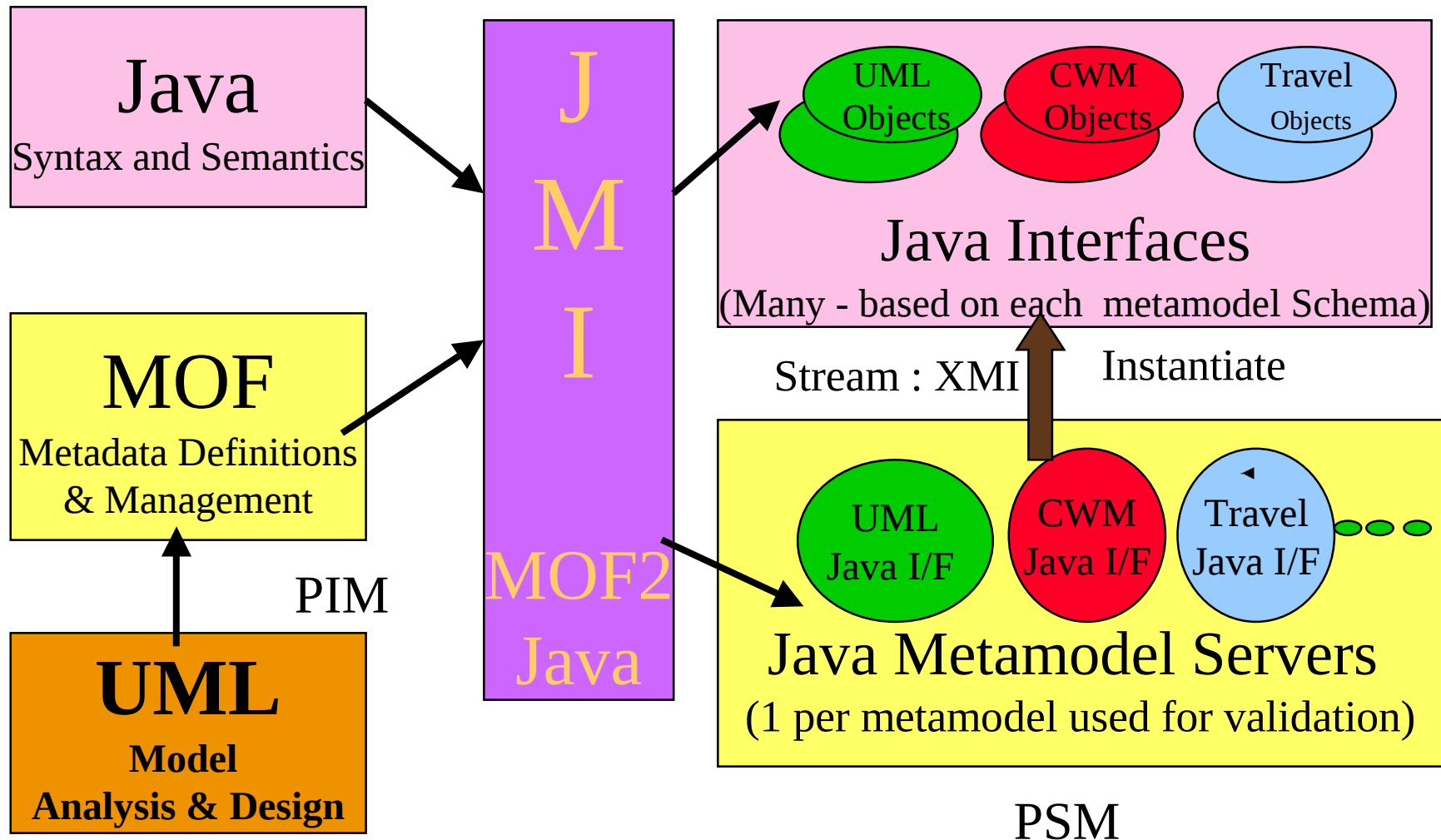
Transform UML Models to Schemas

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



Transform Models to Java

JSR-40 : Java Metadata Interface (JMI) 2002



The 4Ms of MDA

Models, Metadata, Mappings, Middleware

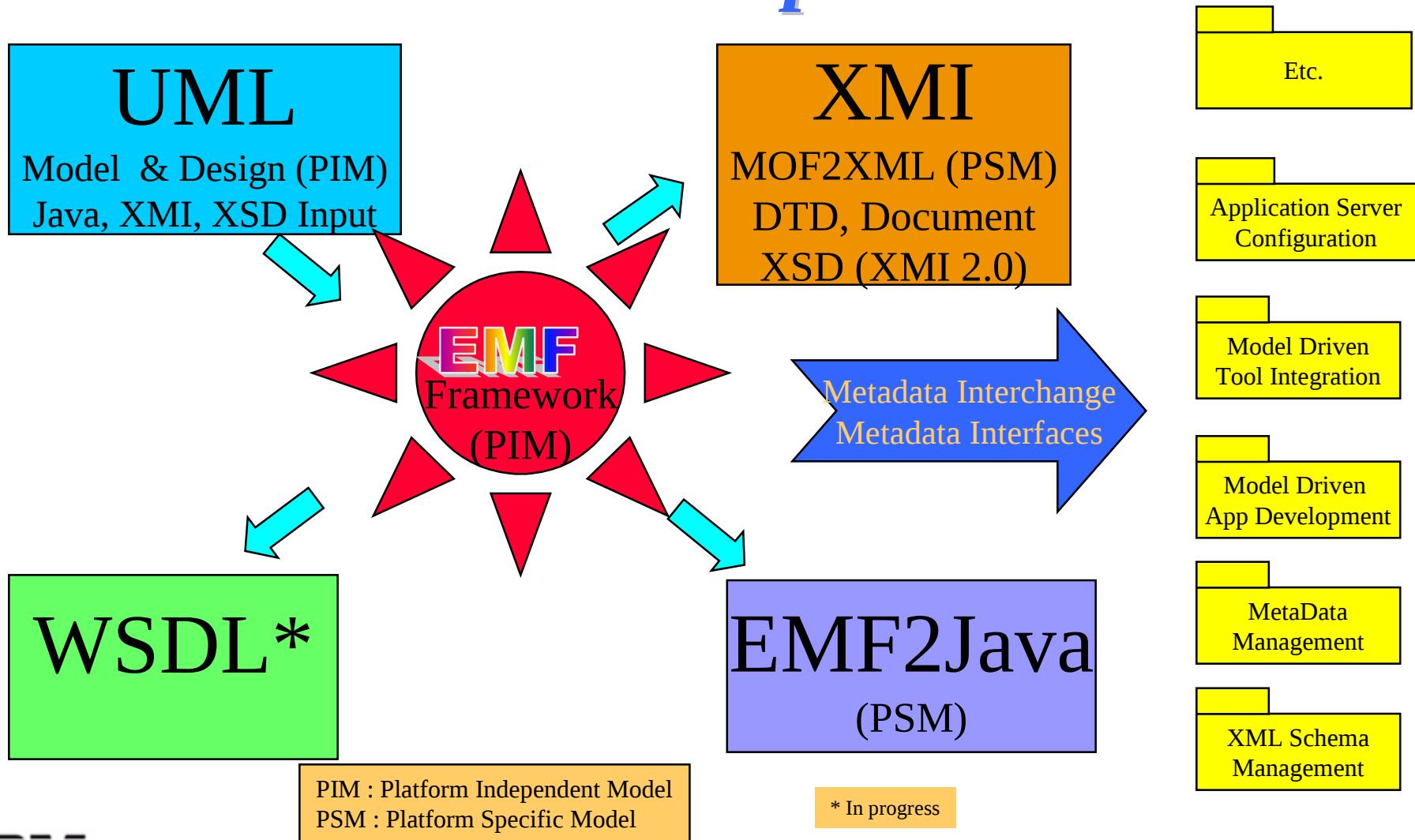
Problem Domain	Source Meta Model (PIM)	Mapping Specification	Target Middleware, Language (PSM)	Interchange Format/API	Comments	Status
Application Development OO Analysis & Design	UML	UML Profile for CORBA	CORBA	XMI	OMG Spec	Adopted 1997
Software Process Engineering	UML	UML Profile for SPEM, SPEM Metamodel	Agnostic	XMI		Adopted 2001
Metadata Management	MOF	MOF to IDL	CORBA	IDL interfaces	PIM for metadata	Adopted 1997
	MOF	MOF to XML DTD (XMI) MOF to XML Docs (XMI)	XML DTD, XML Documents	XMI	Common Interchange Format	Adopted 1999
	MOF	MOF to XML Schema (XMI) XML to MOF (Reverse)	XML Schema, 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 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

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

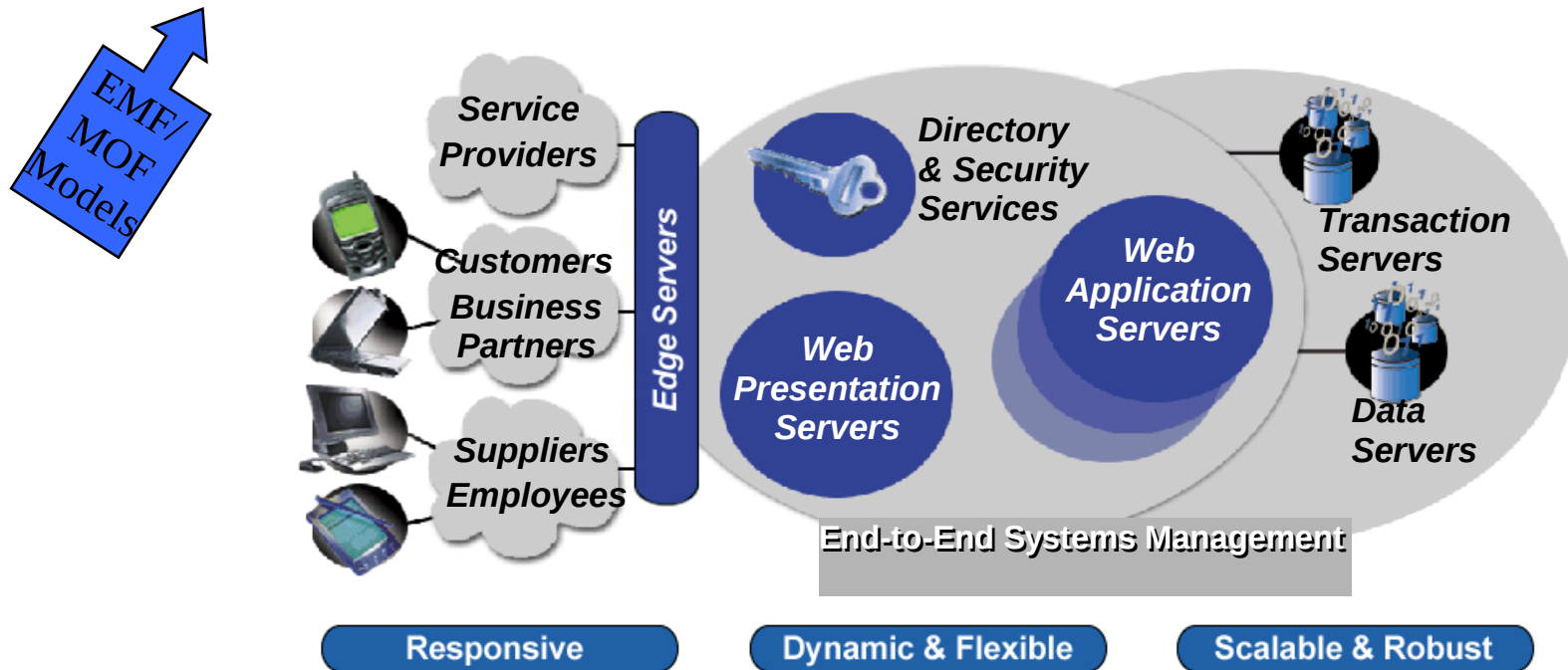
EMF in WebSphere Tools



Sample EMF Metamodels Used in WebSphere

Integration of key software domains

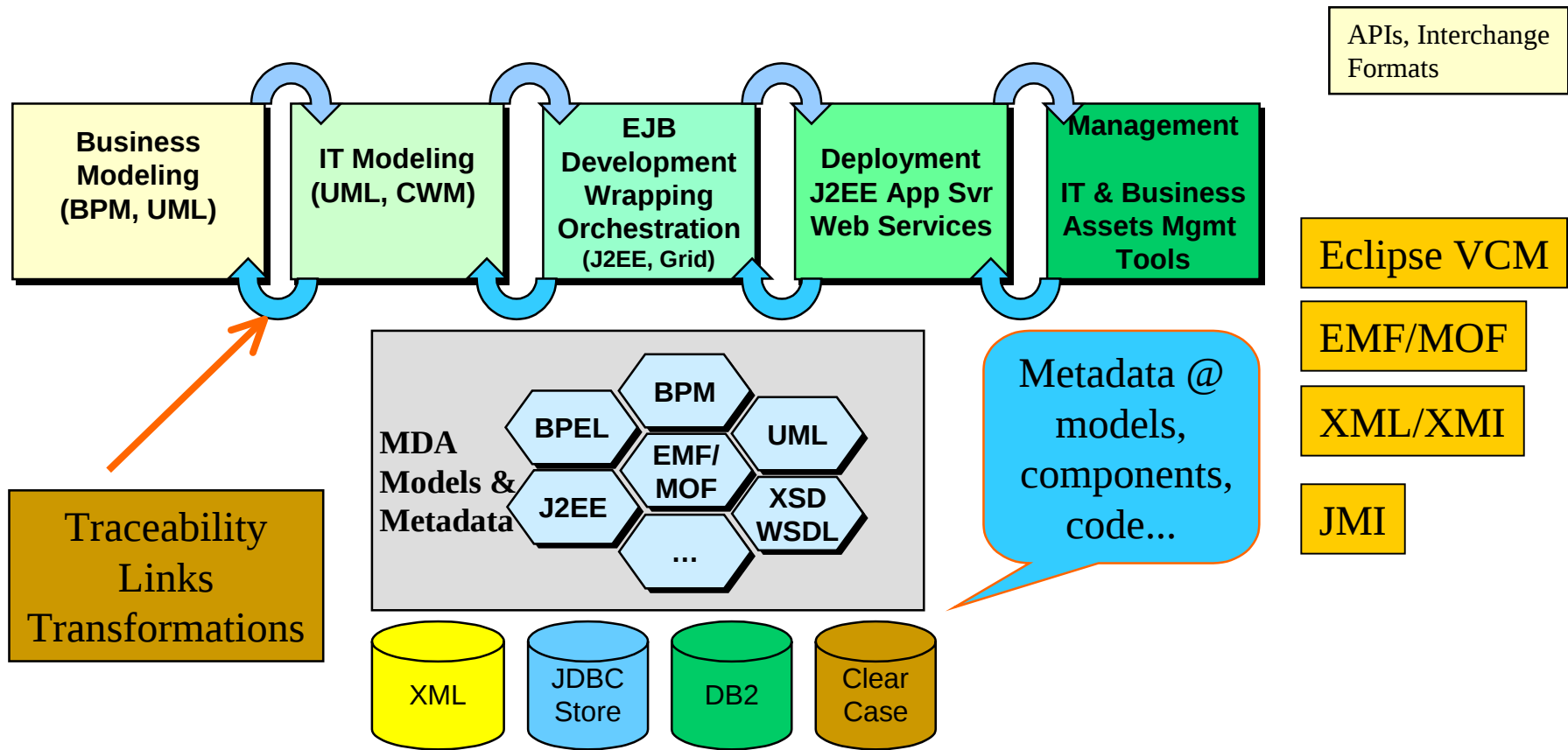
XMI	XML	HTML	JSP	EJB	Java	C/C++	COBOL	SQL	BPM*
MOF	XSD	BPEL*	WSDL	UDDI	FCM	Mapping	WCCM	UML*	



* In R&D

Model Driven Tools Integration

An end to end view - A peek ahead?



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

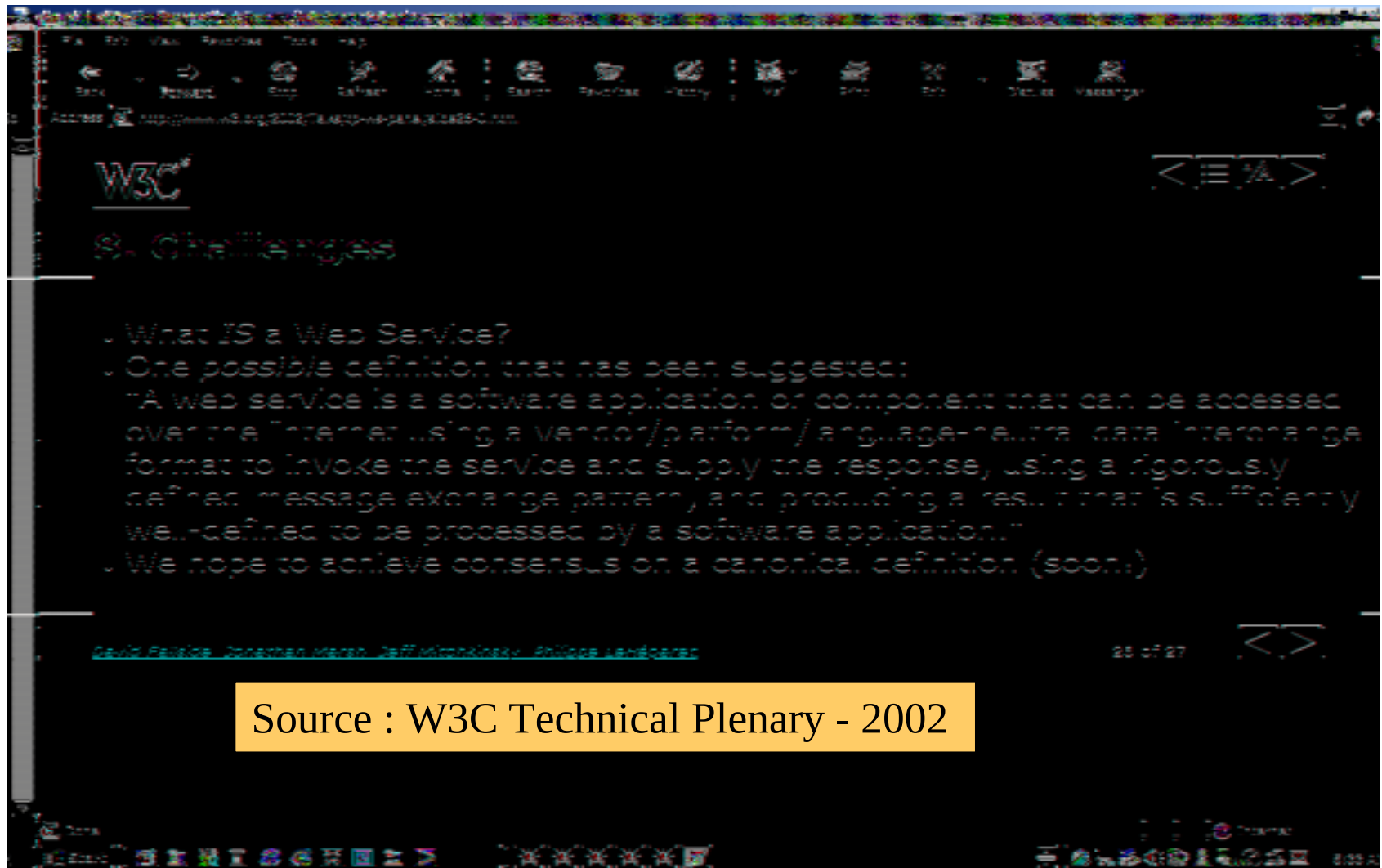
- Introduction - The integration problem
- OMG MDA - Model Driven Architecture Overview
 - What is MDA and why is it important
 - Emerging MDA standards for Model Driven Business Integration
- Web Services Architecture Overview
 - What are Web Services and why are they important
 - Emerging Web Services 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
- Can be simple to very complex; free to expensive

What is a Web Service?

'The challenge of defining it'



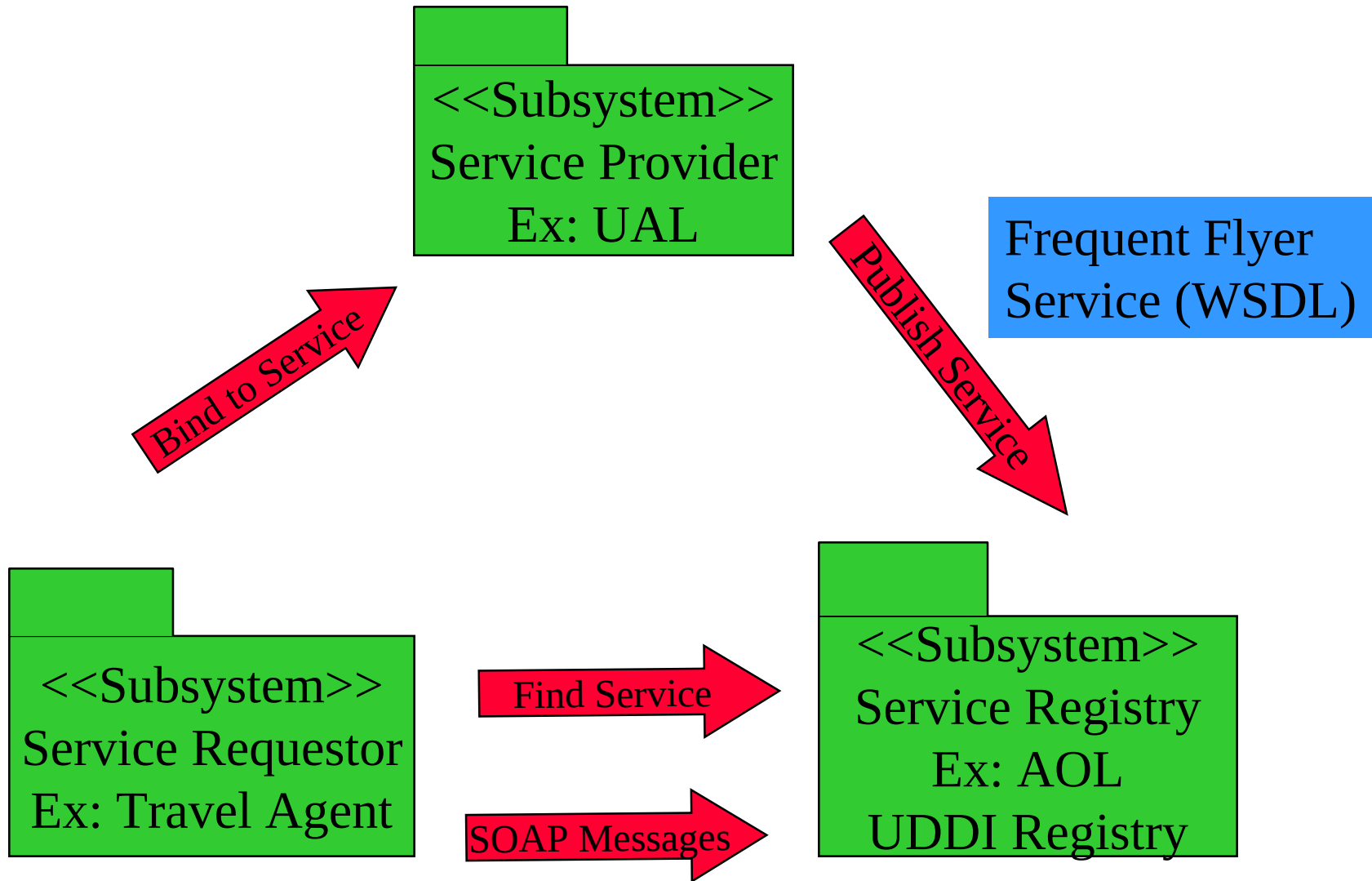
The image is a screenshot of a presentation slide from the W3C Technical Plenary in 2002. The slide is titled 'W3C 8. Challenges' and discusses the difficulty of defining a web service. It includes a list of bullet points and a definition of a web service. The slide is presented in a window with a standard Windows XP interface, including a taskbar at the bottom and a menu bar at the top. The text on the slide is as follows:

- 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!)

At the bottom of the slide, the names of the participants are listed: David Faizi, Jonathan Marsh, Jeff Winkler, and Philippe Verdooren. The slide number '28 of 27' is also visible in the bottom right corner.

Source : W3C Technical Plenary - 2002

Web Services : The basic idea





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

- An 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
- Standards and tools are beginning to mature
 - More work – especially Transactions, Security etc.
- 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-IO 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
 - BPML now builds on BPEL4WS

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 Stack Evolves

Specific Composable Services and Processes
(ebXML, RosettaNet PIP...)

Service Flow (BPEL4WS..)

Service Publication and Discovery (UDDI)

Service Description (WSDL)

XML Messaging (SOAP, XMLP)

Network (HTTP, FTP...)

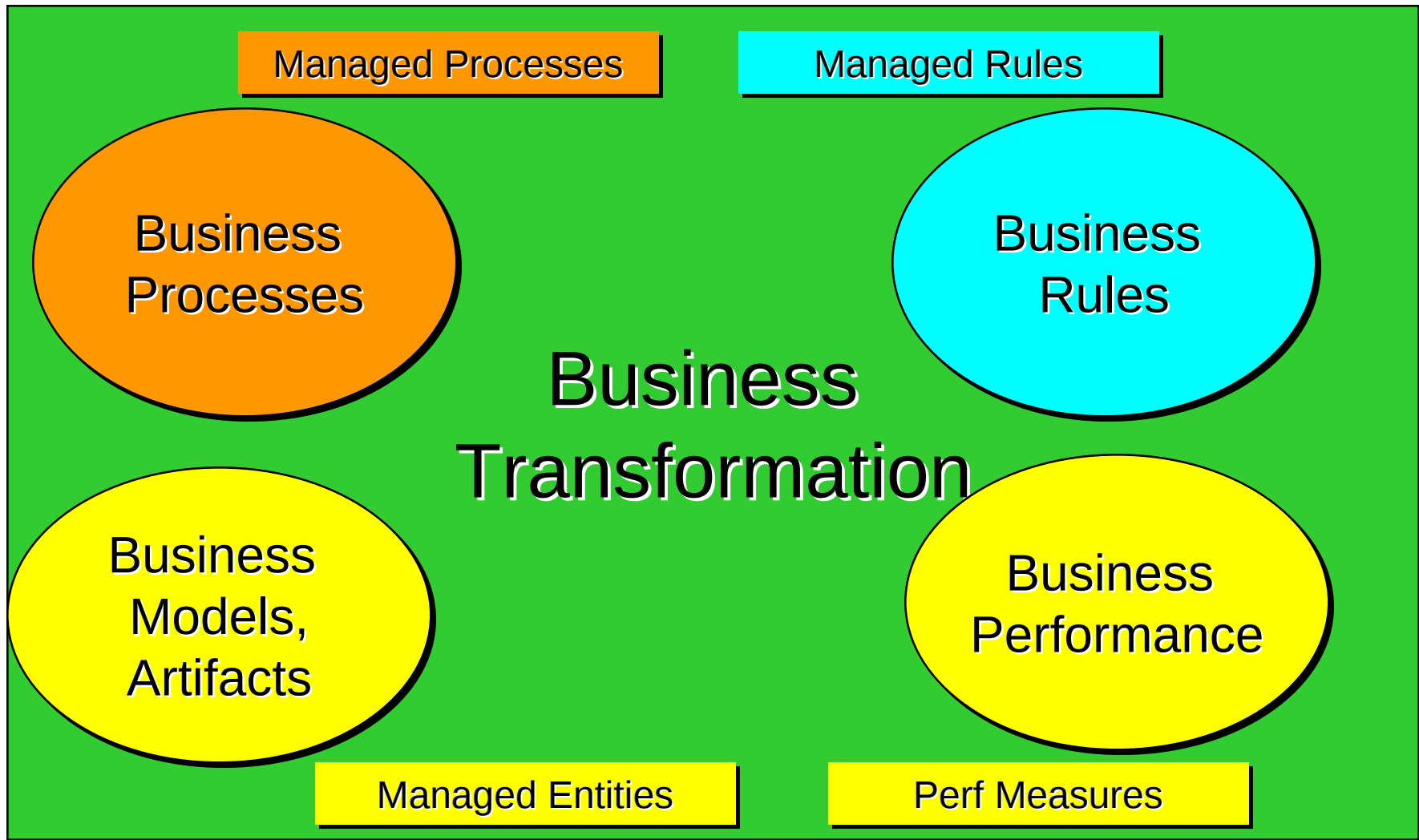
Summary Web Services Standards

- W3C
 - XMLP (SOAP+), WSDL, XAML...
 - W3C Architecture Working Group
- UDDI Consortium → UDDI 3.0 being progressed in OASIS
- 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
 - Web Services for Business Intelligence
 - Modeling and metadata management for Web Services
 - UML2 Activity Graphs, UML2 Infrastructure, MOF2
 - Business Process Definition Metamodel RFP
- OASIS
 - ebXML, UDDI...
- 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

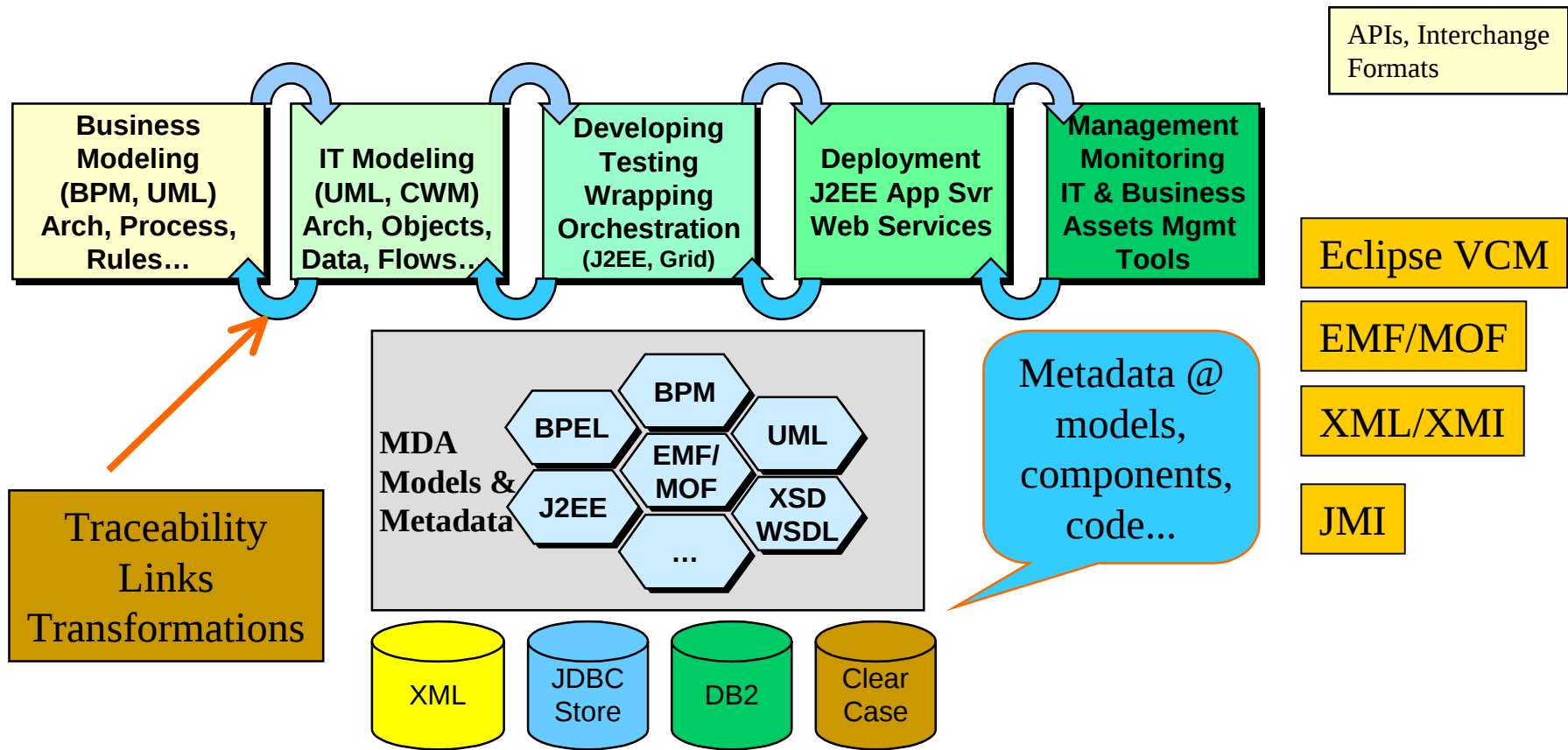
- Introduction - The integration problem
- OMG MDA - Model Driven Architecture Overview
 - What is MDA and why is it important
 - Emerging MDA standards for Model Driven Business Integration
- Web Services Architecture Overview
 - What are Web Services and why are they important
 - Emerging Web Services 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

Business Models, Processes and Rules



Model Driven Tools Integration

An end to end view - A peek ahead?



Business Modeling : MDA Computation Independent Model (BPM)
 IT Modeling : MDA Platform Independent Model (UML, CWM)
 MDA Platform Specific Model (J2EE...)
 Model Transformations across layers

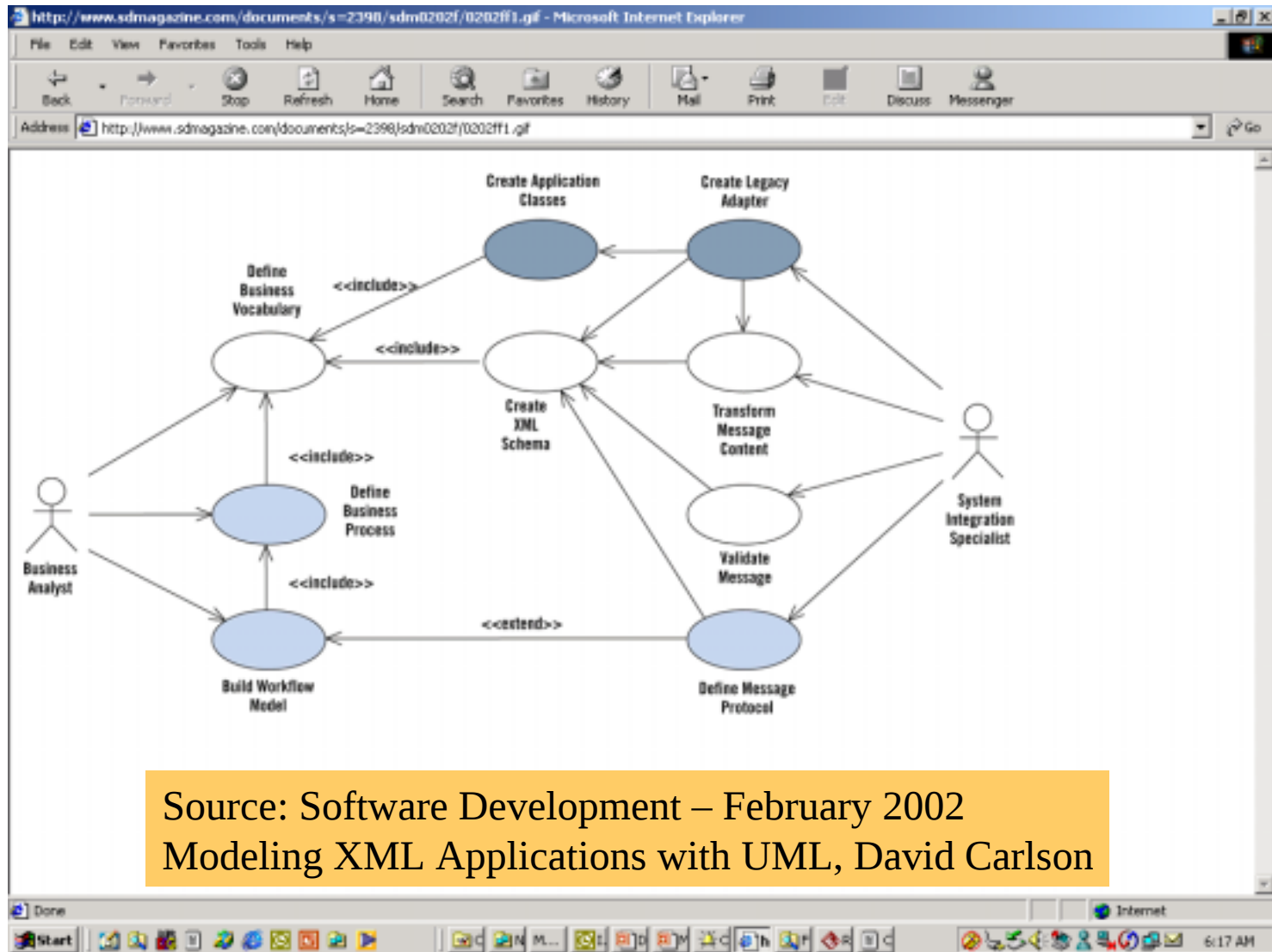
Modeling for Web Services

- 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
- 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

Business Process Definition Metamodel RFP

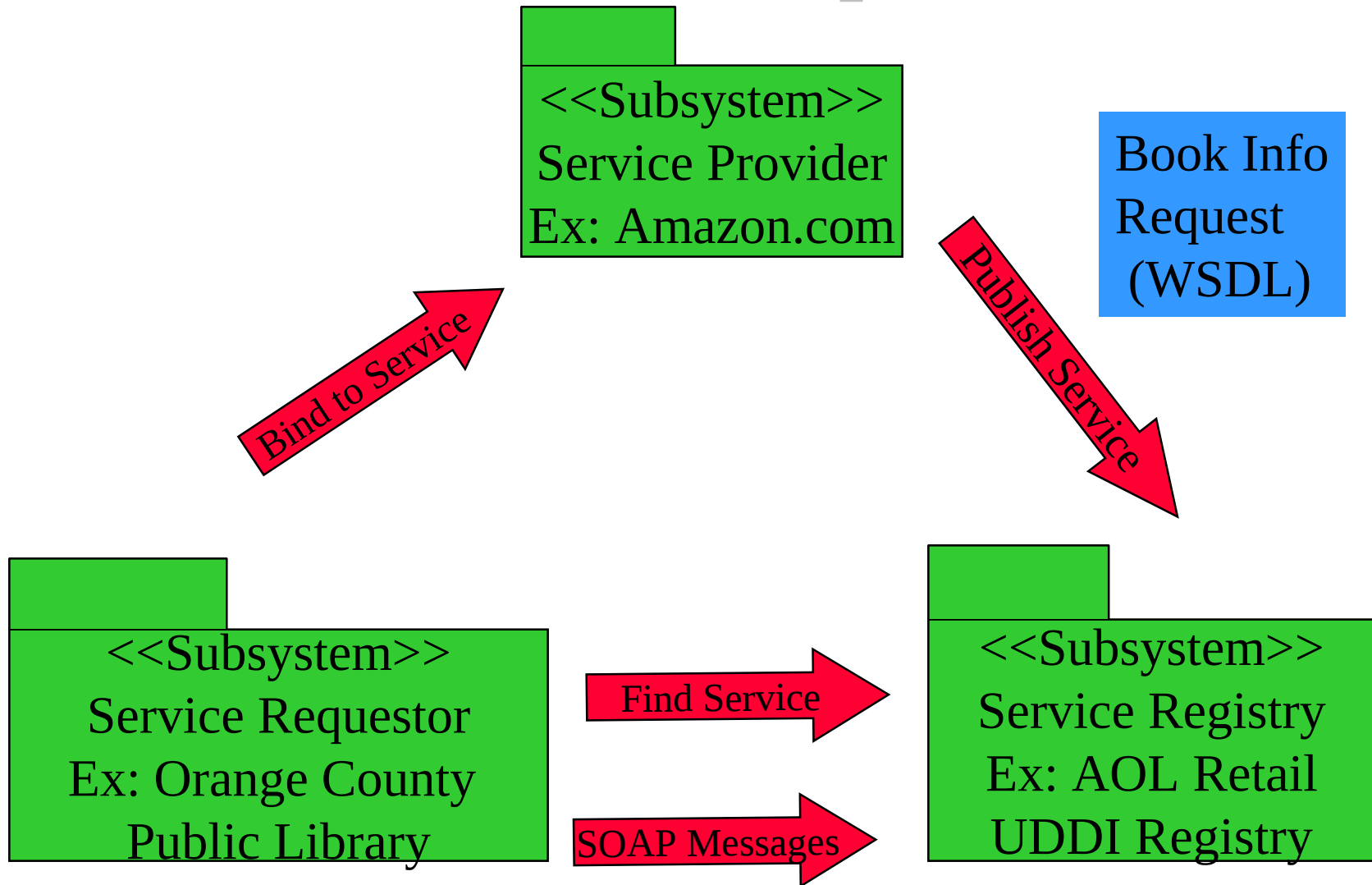
- Issued January 2003
- Goal is to define a platform independent model that unifies the modeling concepts in emerging flow/choreography standards
- Proof of concept mappings to emerging choreography standards (BPEL4WS, XPDL...)
- Basis for providing alternate visualizations as well as textual/machine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

E-Business Integration using XML

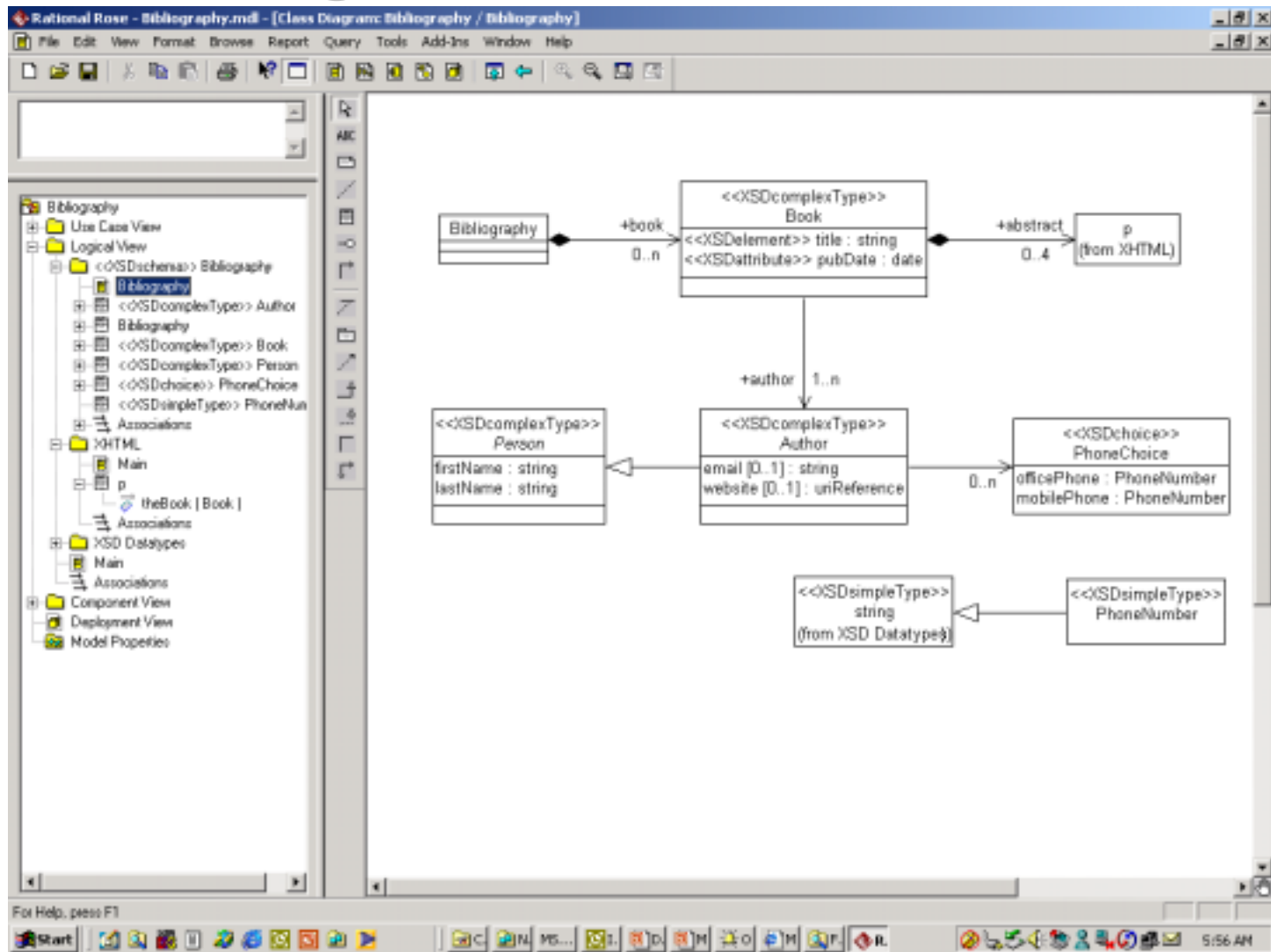


Source: Software Development – February 2002
Modeling XML Applications with UML, David Carlson

Web Services for Book Store – An Example

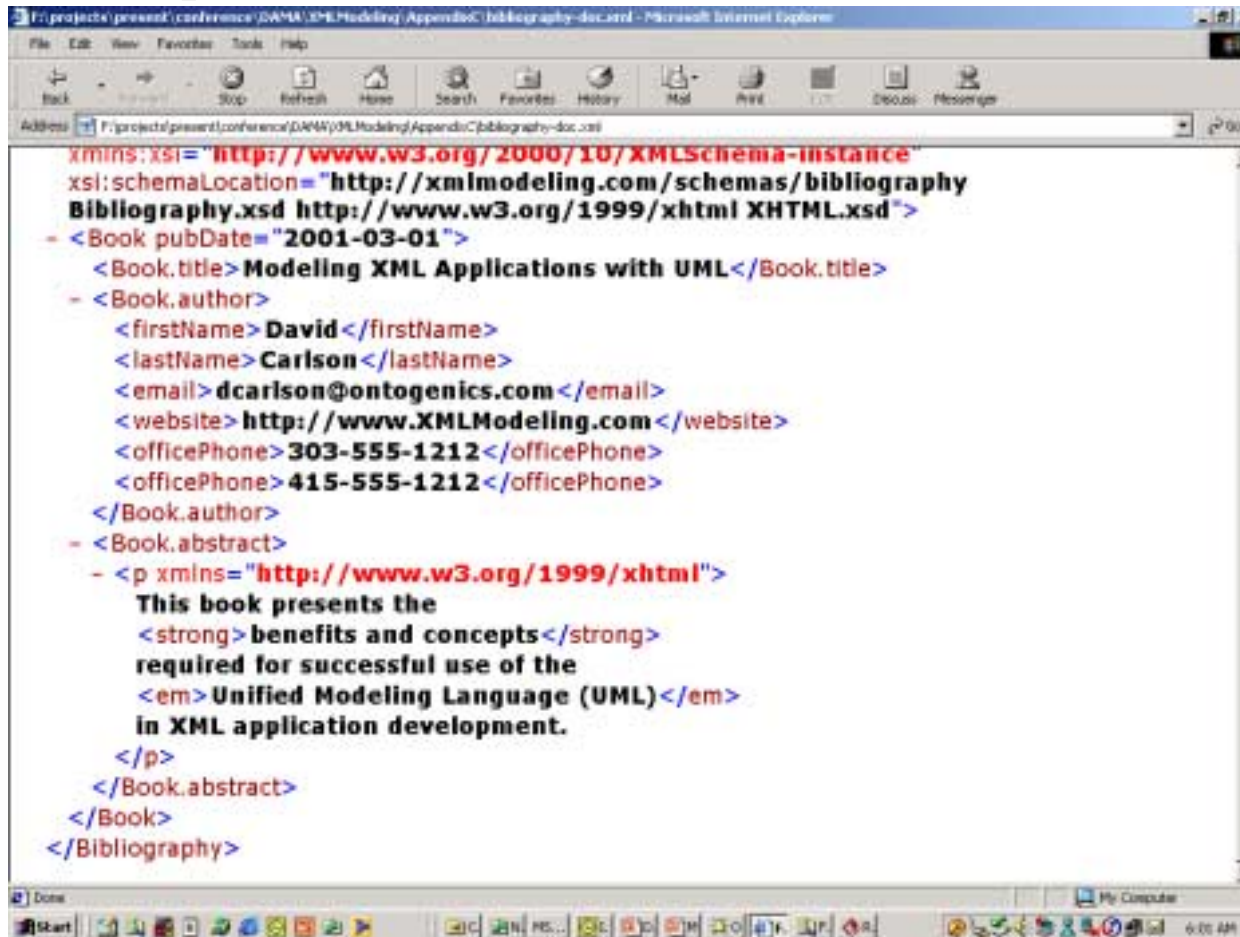


Modeling Web Services and XML



Usage : Model the Application

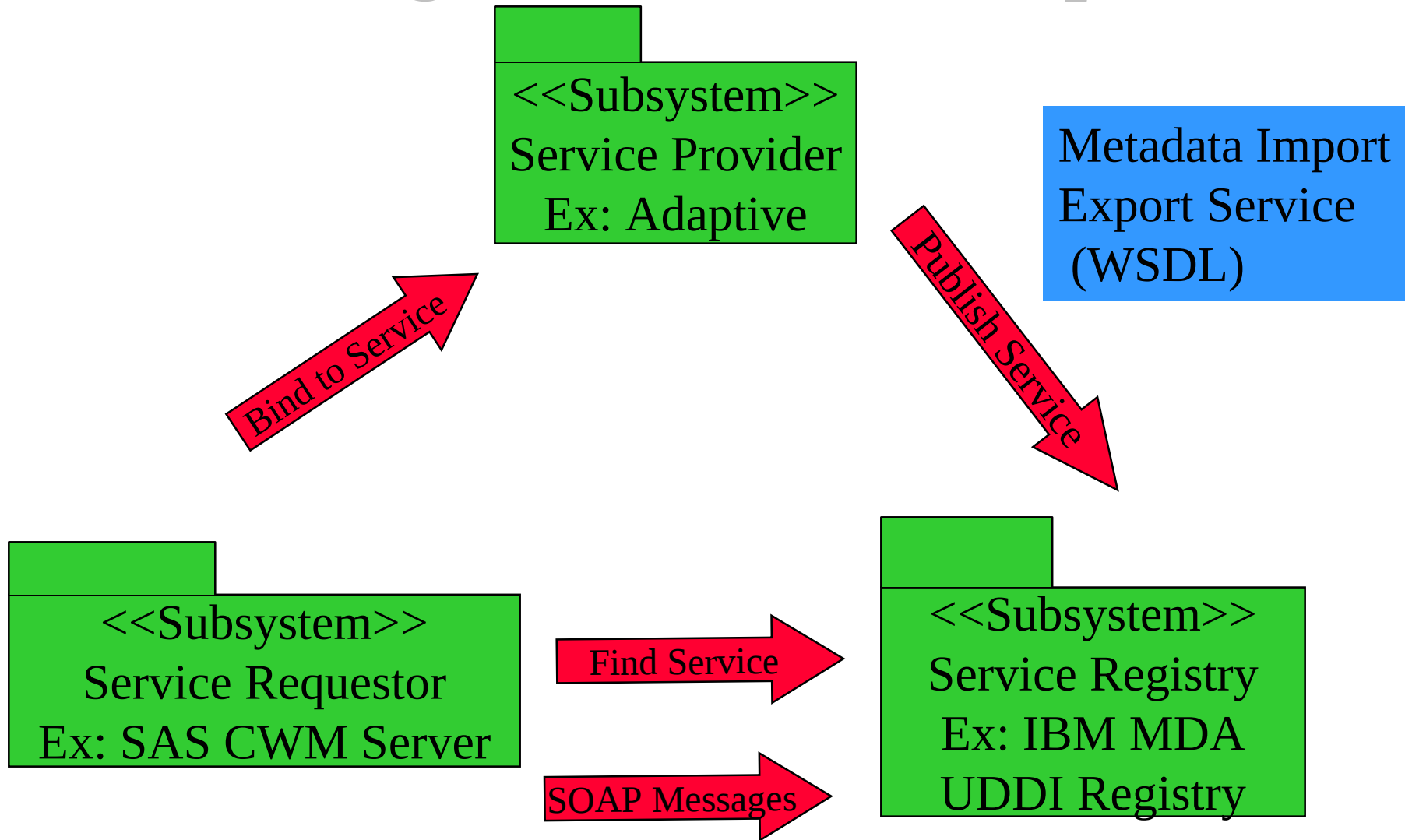
An XML Document that conforms to the UML model



```
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>
```

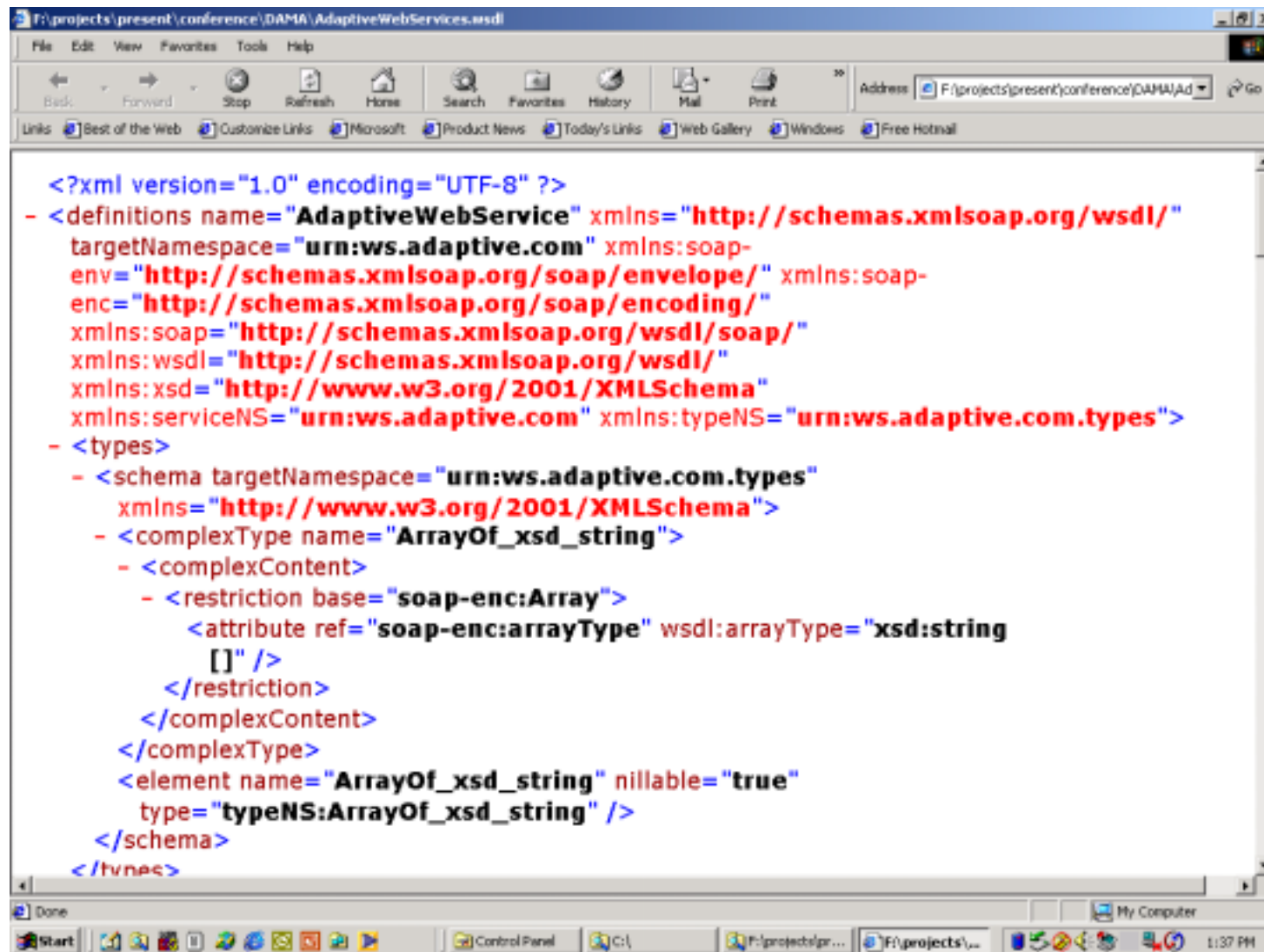
Usage : Generate XML Schemas and Documents

Web Services for Metadata Management – An Example



Web Services for Metadata Management

WSDL Header

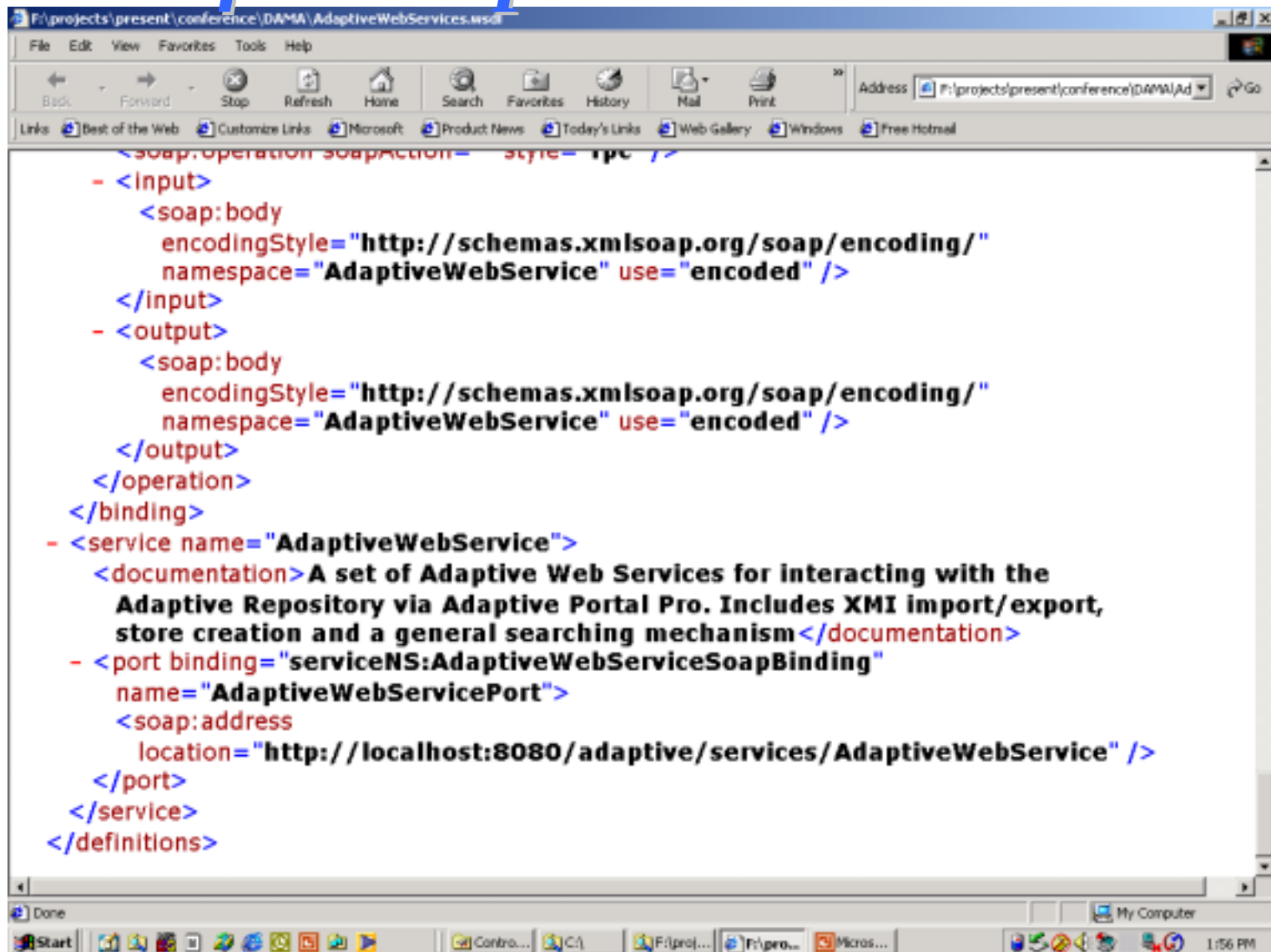


The screenshot shows a web browser window with the address bar displaying the file path: F:\projects\present\conference\ADAMA\AdaptiveWebServices.asdl. The browser's menu bar includes File, Edit, View, Favorites, Tools, and Help. The toolbar contains buttons for Back, Forward, Stop, Refresh, Home, Search, Favorites, History, Mail, and Print. The address bar also includes a Go button. Below the toolbar, there are links to Best of the Web, Customize Links, Microsoft, Product News, Today's Links, Web Gallery, Windows, and Free Hotmail. The main content area displays the WSDL header for AdaptiveWebService.asdl, which is an XML document. The XML starts with a declaration: <?xml version="1.0" encoding="UTF-8" ?>. The root element is <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">. The <definitions> element contains a <types> element, which in turn contains a <schema targetNamespace="urn:ws.adaptive.com.types" xmlns="http://www.w3.org/2001/XMLSchema"> element. The <schema> element contains a <complexType name="ArrayOf_xsd_string"> element, which contains a <complexContent> element. The <complexContent> element contains a <restriction base="soap-enc:Array"> element, which contains an <attribute ref="soap-enc:arrayType" wsdl:arrayType="xsd:string []" /> element. The <restriction> element contains a </restriction> element. The <complexContent> element contains a </complexContent> element. The <complexType> element contains a </complexType> element. The <schema> element contains a <element name="ArrayOf_xsd_string" nillable="true" type="typeNS:ArrayOf_xsd_string" /> element. The <schema> element contains a </schema> element. The <types> element contains a </types> element. The <definitions> element contains a </definitions> element.

```
<?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>
```

Web Services for Metadata Management

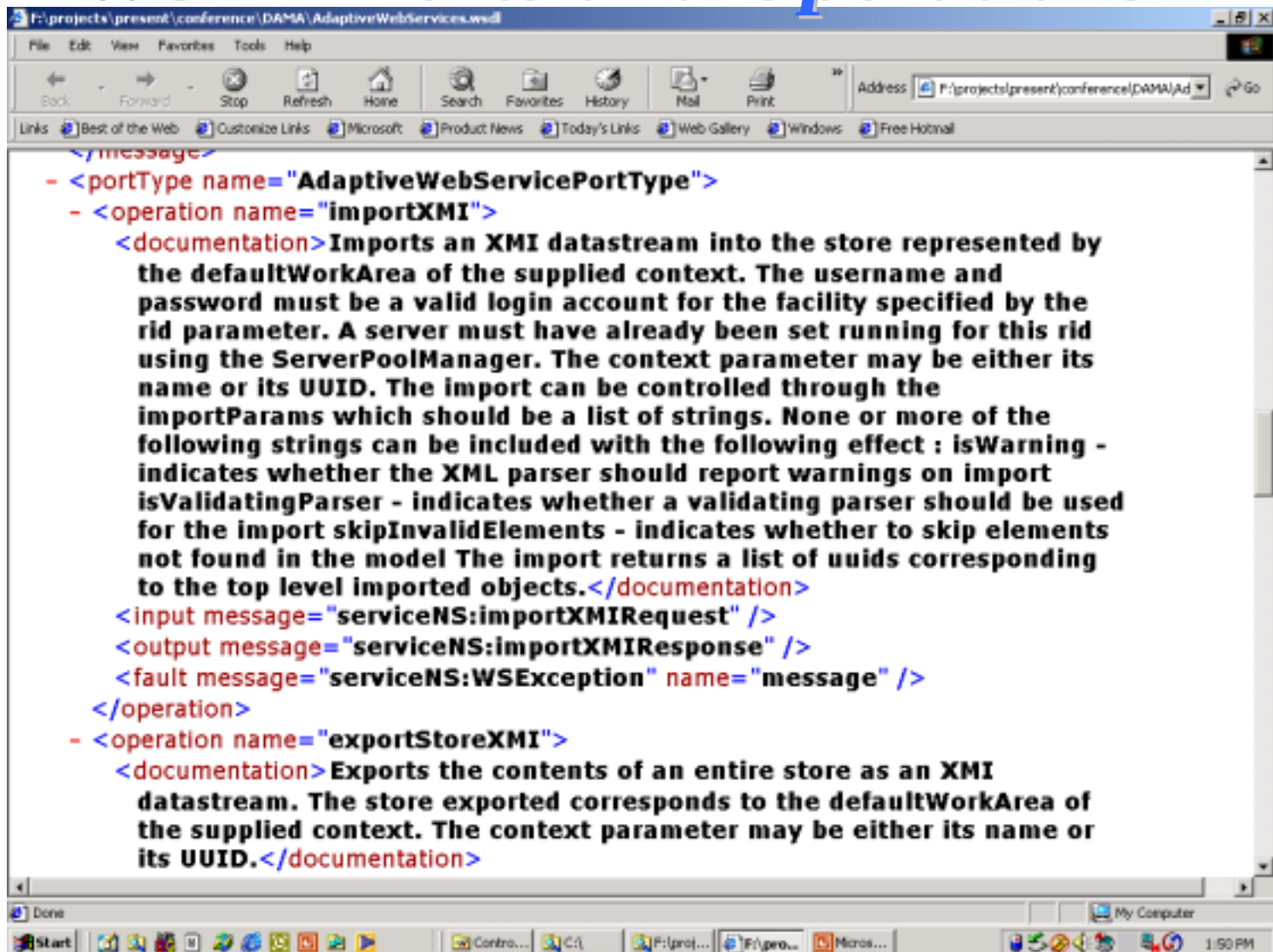
WSDL for Adaptive Web Services Port



```
<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>
```

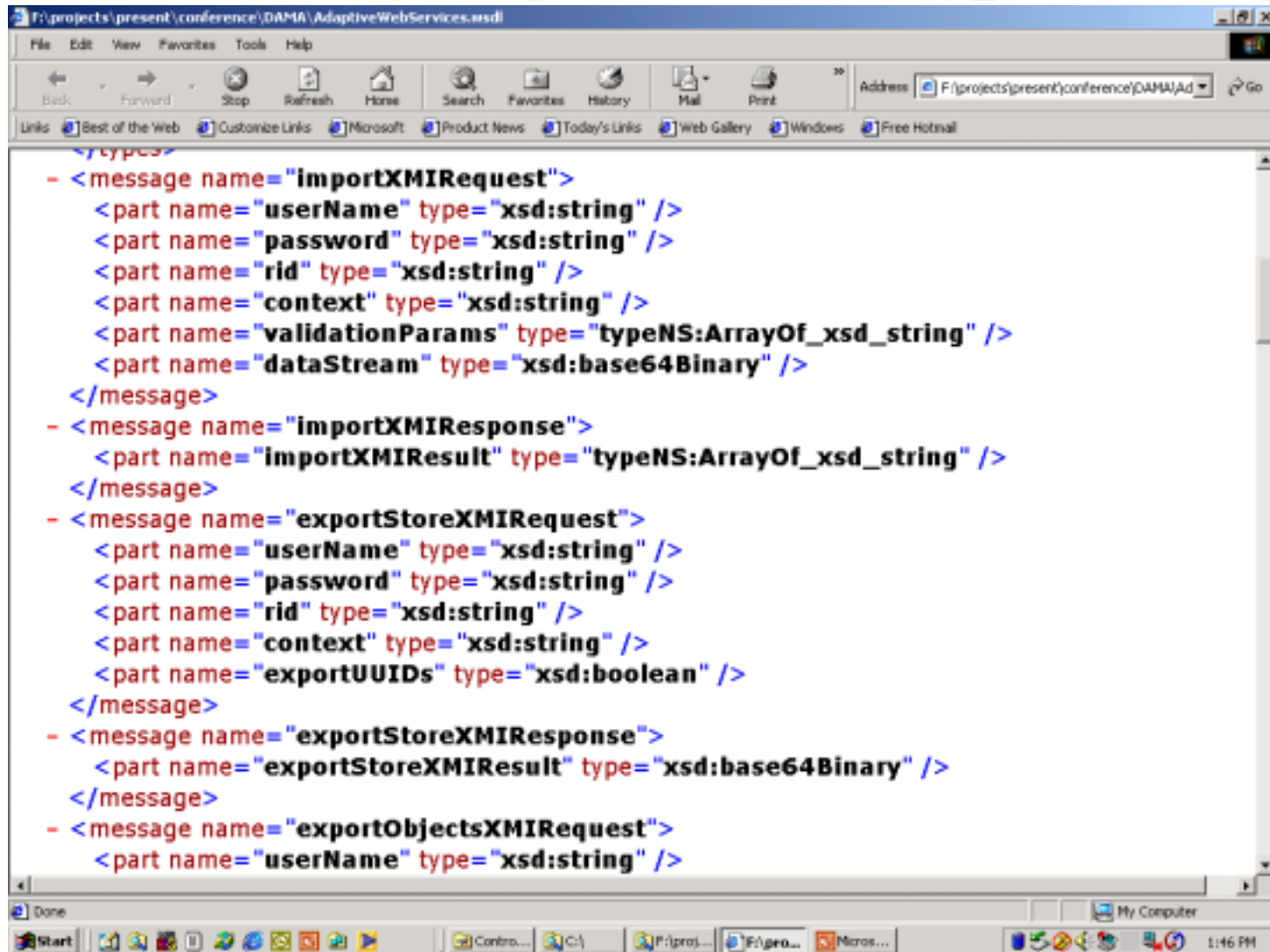
Web Services for Metadata Management

WSDL Ports and Operations



Web Services for Metadata Management

WSDL messages for XMI Import/Export

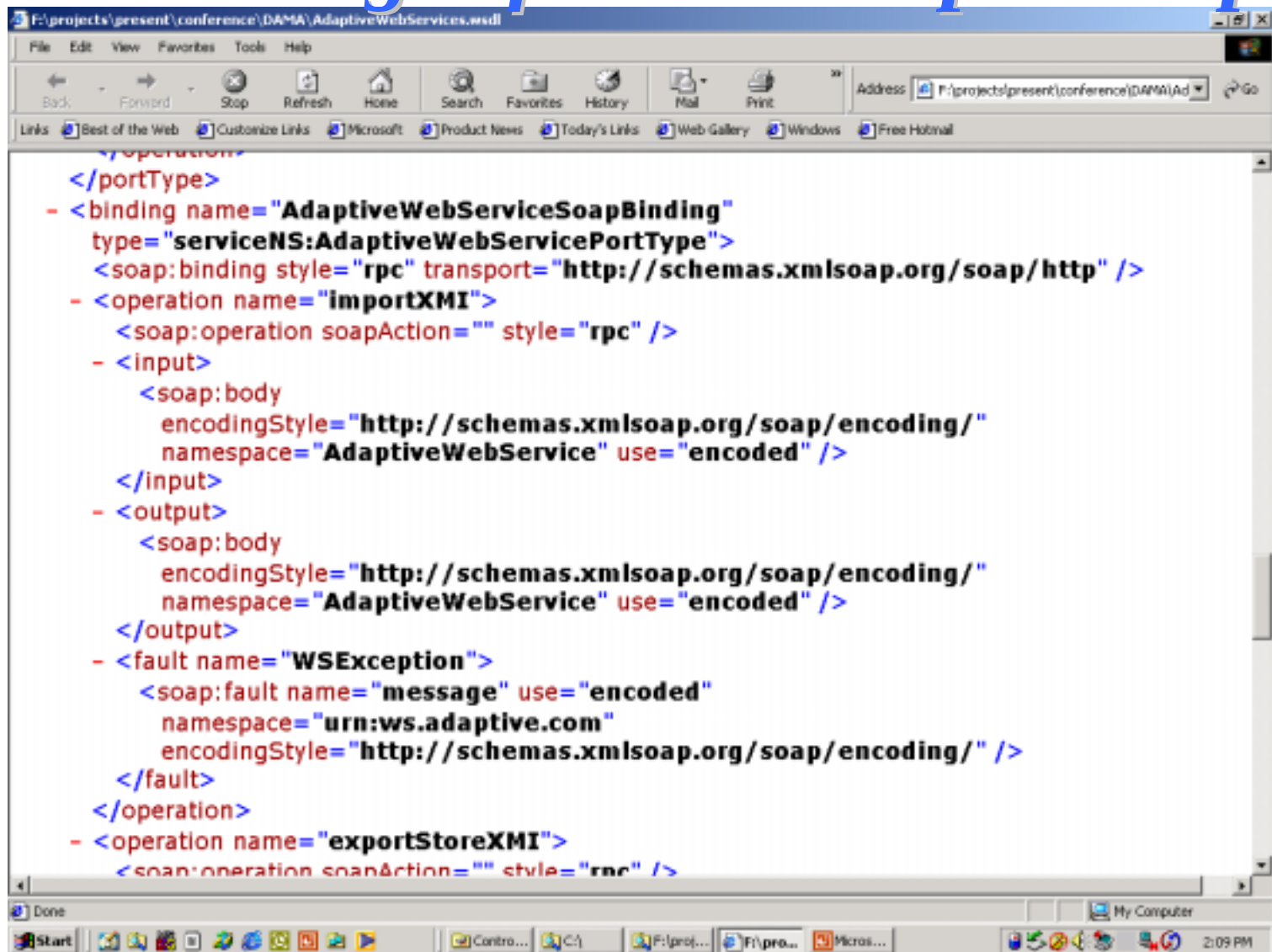


```
<?xml version='1.0'?>
<types>
- <message name="importXMIRequest">
  <part name="userName" type="xsd:string" />
  <part name="password" type="xsd:string" />
  <part name="rid" type="xsd:string" />
  <part name="context" type="xsd:string" />
  <part name="validationParams" type="typeNS:ArrayOf_xsd_string" />
  <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



```
</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="WSEException">
  <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" />
```

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)

A peek at integrating Business Processes using UML and BPEL4WS

This is early work and is intended to spur
Discussion!

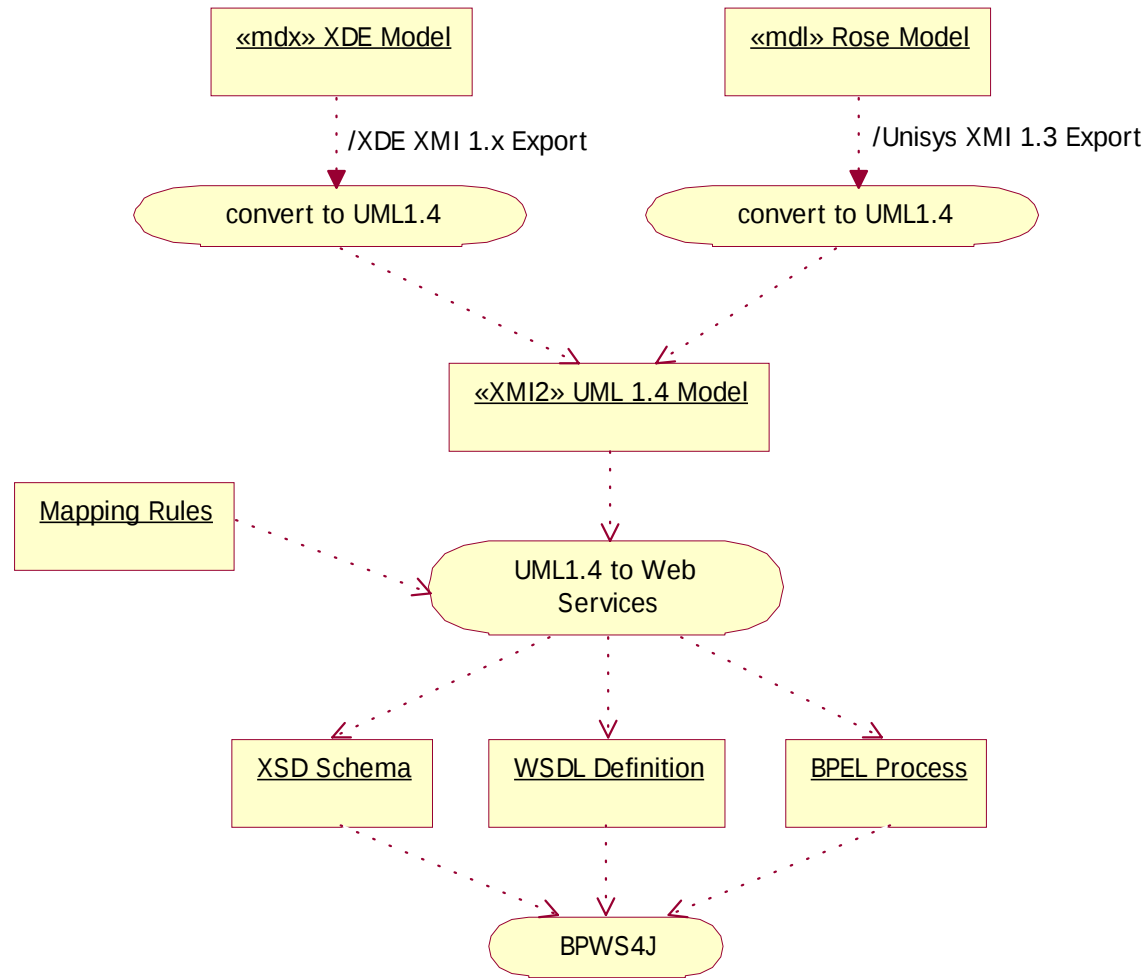
See the BPEL4WS spec for the examples, BPEL4WS syntax

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

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

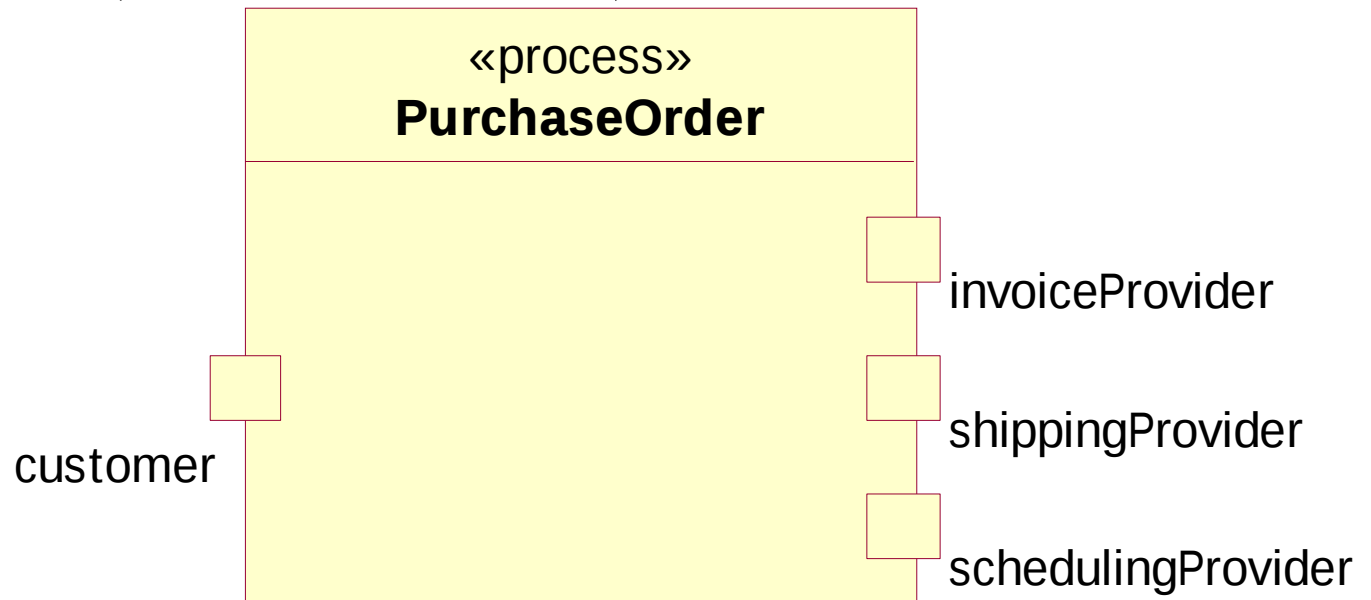
- The profile should cover broadly the same set of concepts as BPEL
- Support the concepts of XSD and WSDL that are required to support BPEL, but don't cover the whole of service oriented architecture in this profile
- Standard UML terminology for concepts is used where available, e.g. Interface rather 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

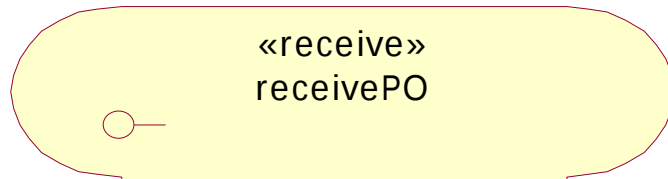
BPEL partners as ports

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

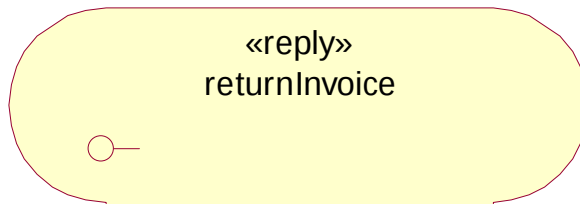


Behavior Basics

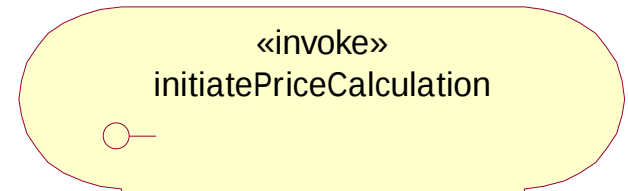
Receive a synchronous operation call:



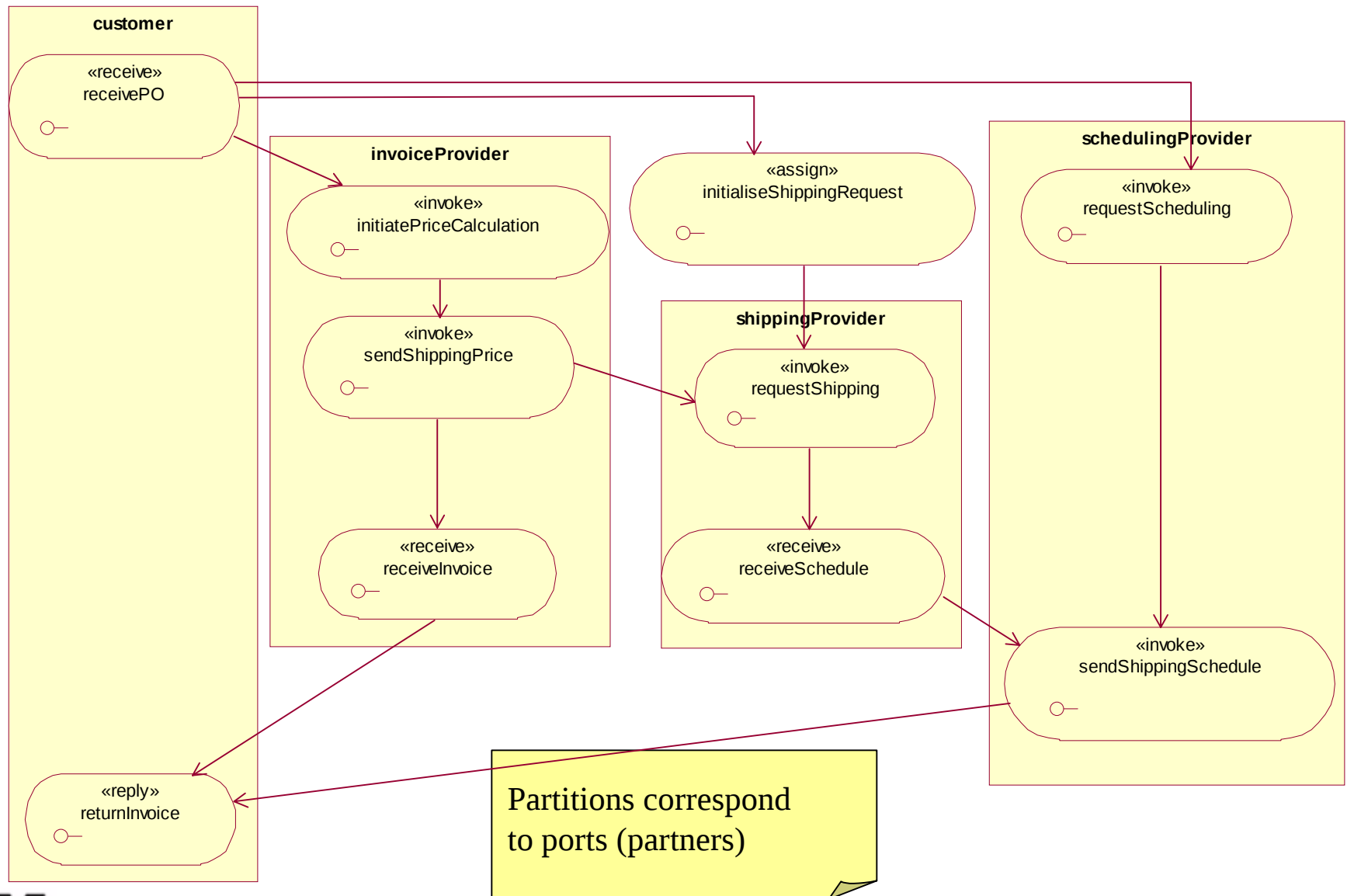
Reply to a previous synchronous request:



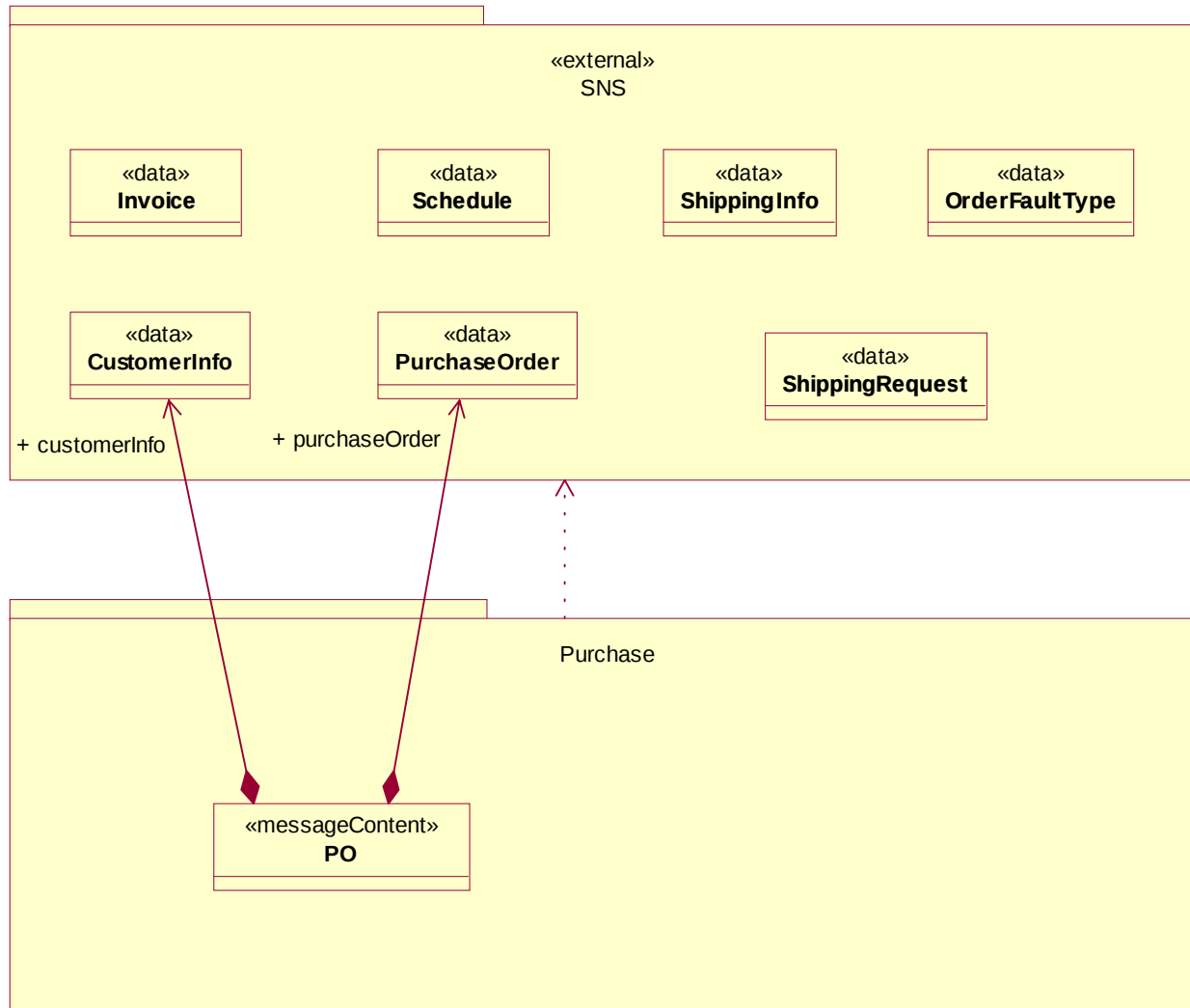
Invoke an operation synchronously or asynchronously on a partner:



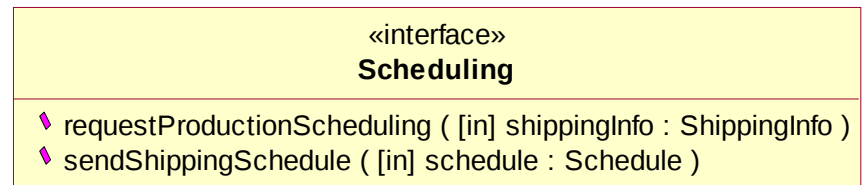
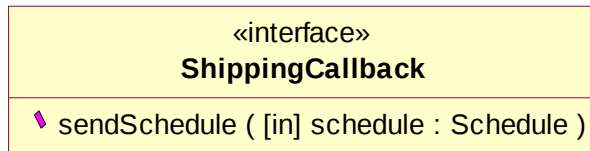
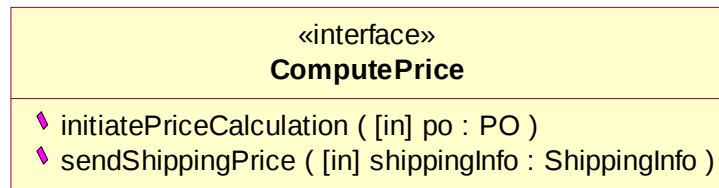
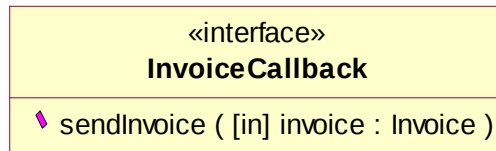
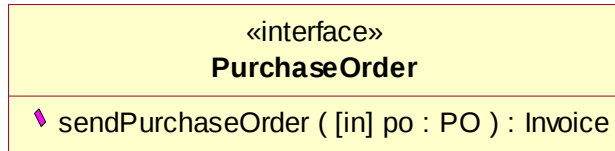
Purchase Order Process



Data and Message Types

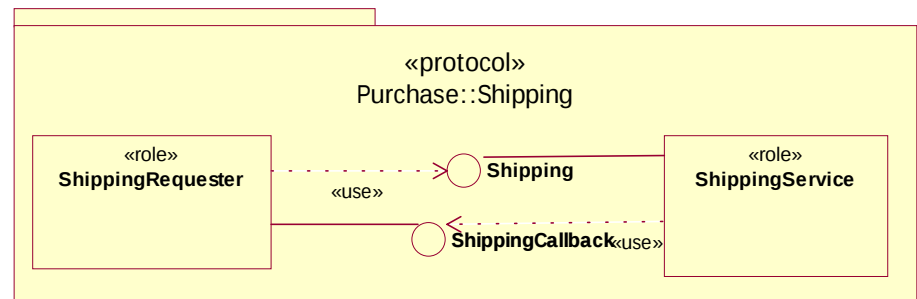
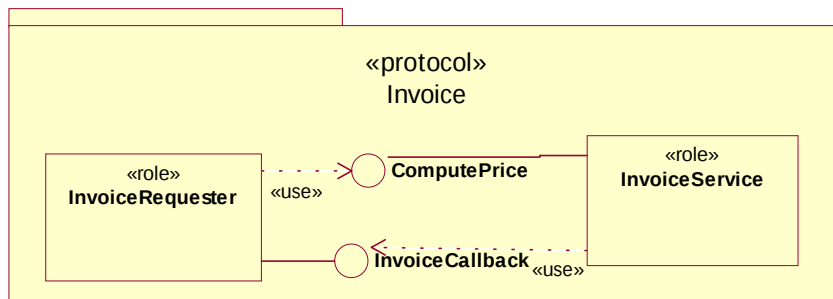
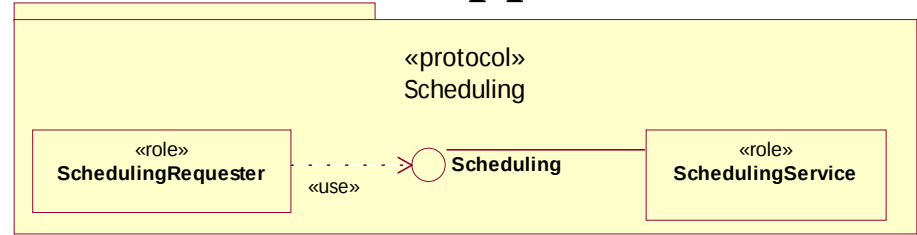
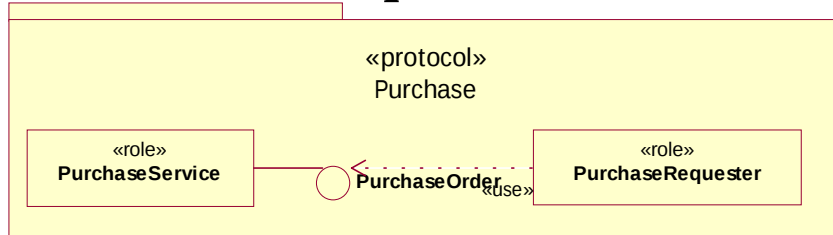


Interfaces

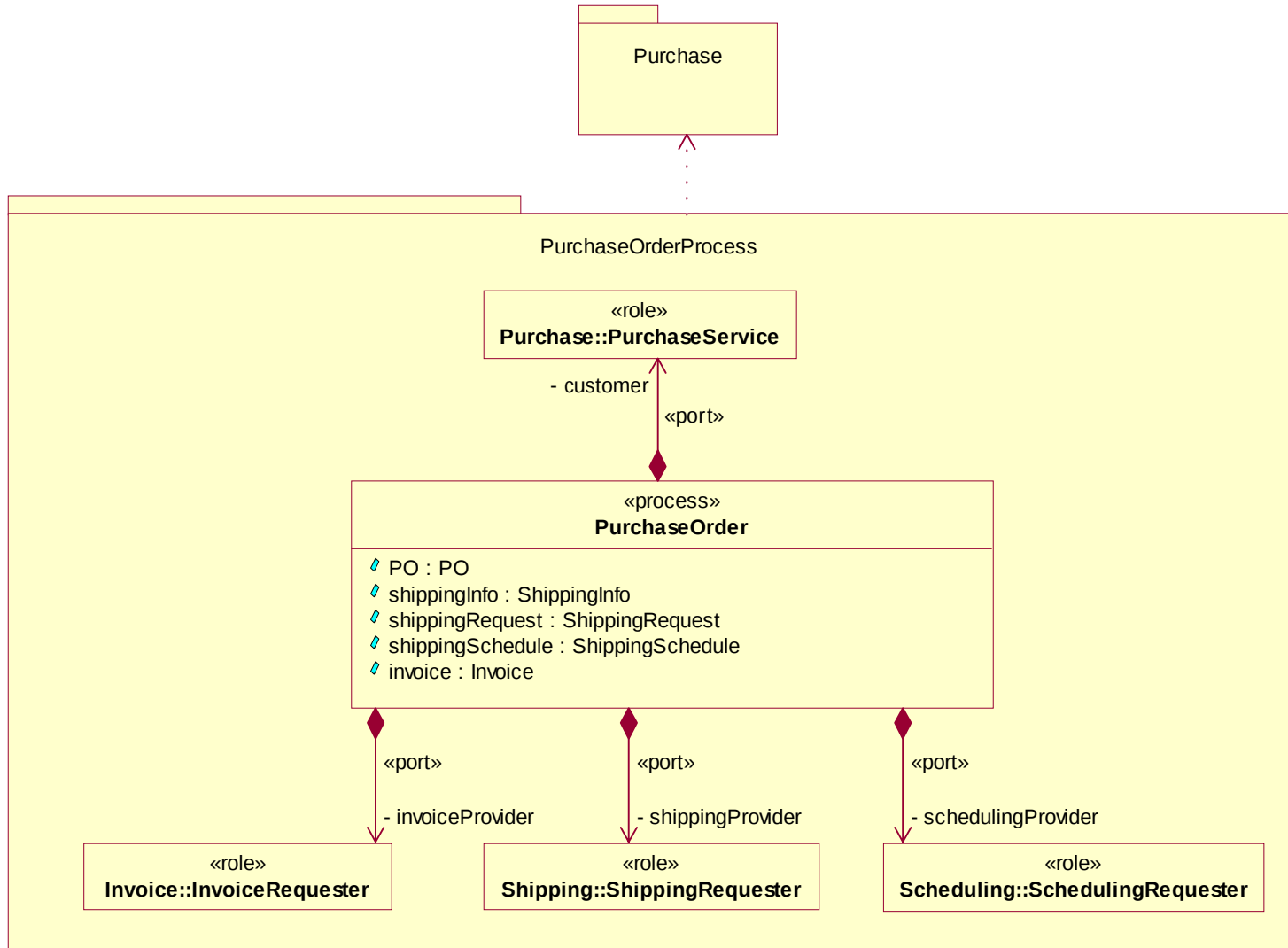


Protocols

- 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



BPEL Mapping Overview

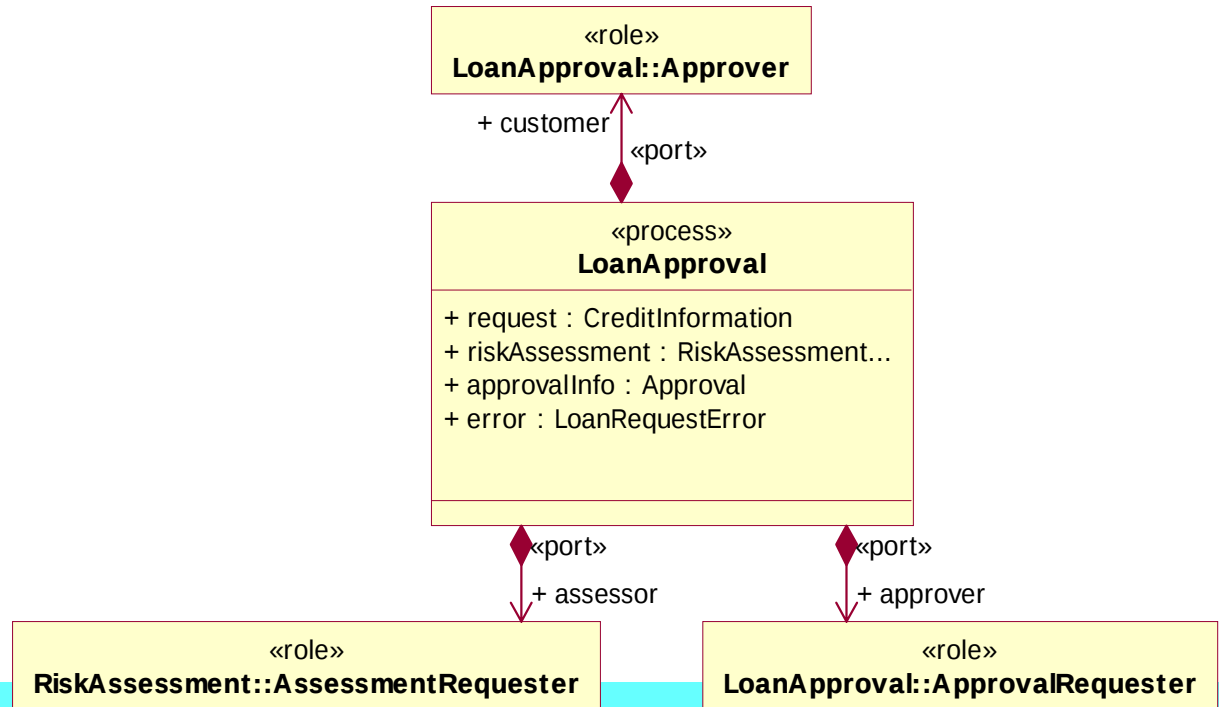
<<process>> class	BPEL process definition
Activity graph on a <<process>> class	BPEL activity hierarchy
<<port>> associations	BPEL partner declarations
<<process>> class	BPEL containers
attributes Hierarchical structure and control flow	BPEL sequence and flow activities
Decision nodes	BPEL switch activities 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
  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>
```

«process»
LoanApprovalProcess::LoanApproval
+ request : CreditInformation
+ riskAssessment : RiskAssessmentDetails
+ approvalInfo : Approval
+ error : LoanRequestError

Mapping: Partners



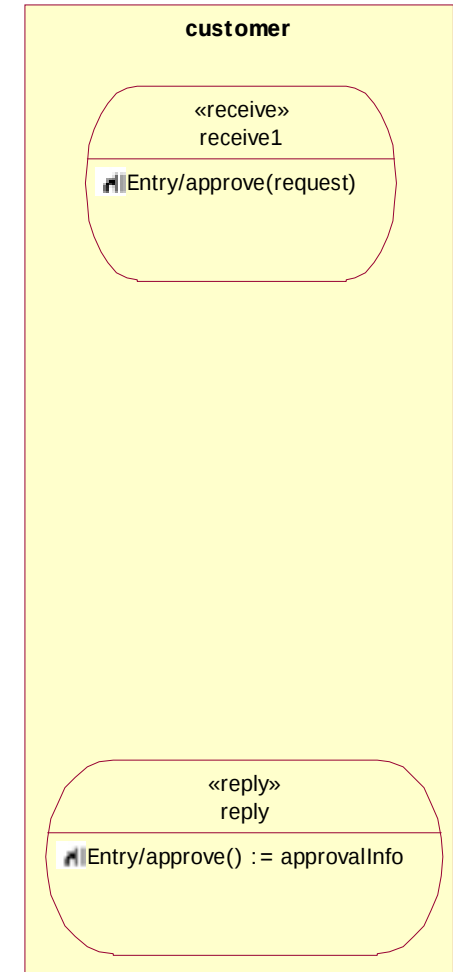
```

<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>
  
```

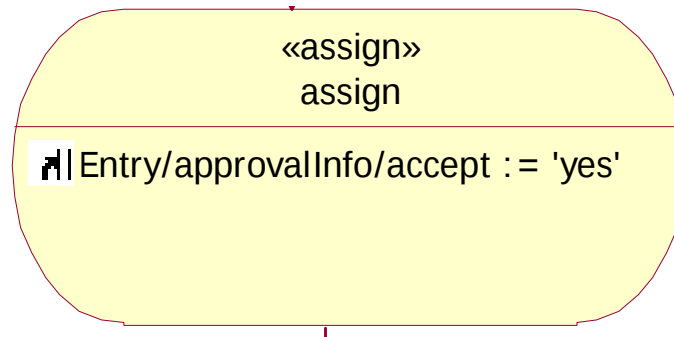
Mapping: Receive/Reply Activities

```
<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') >= 10000"/>
  <source linkName="receive1_to_invokeAssessor"
    transitionCondition="bpws:getContainerData
('request','amount') < 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>
```



Mapping: Assign Activities

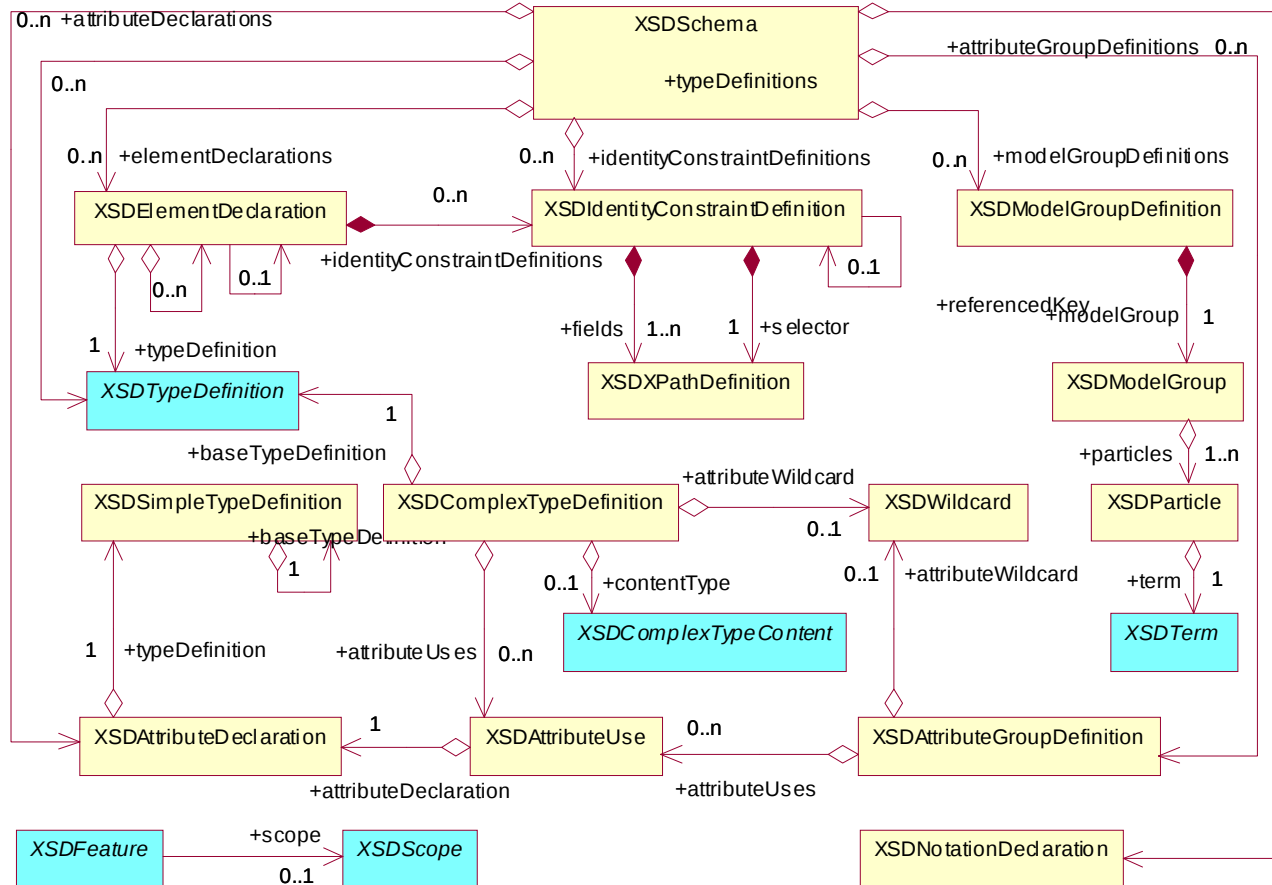


```
<assign name="assign">  
  <copy>  
    <from expression="'yes'"/>  
    <to container="approvalInfo" part="accept"/>  
  </copy>  
</assign>
```

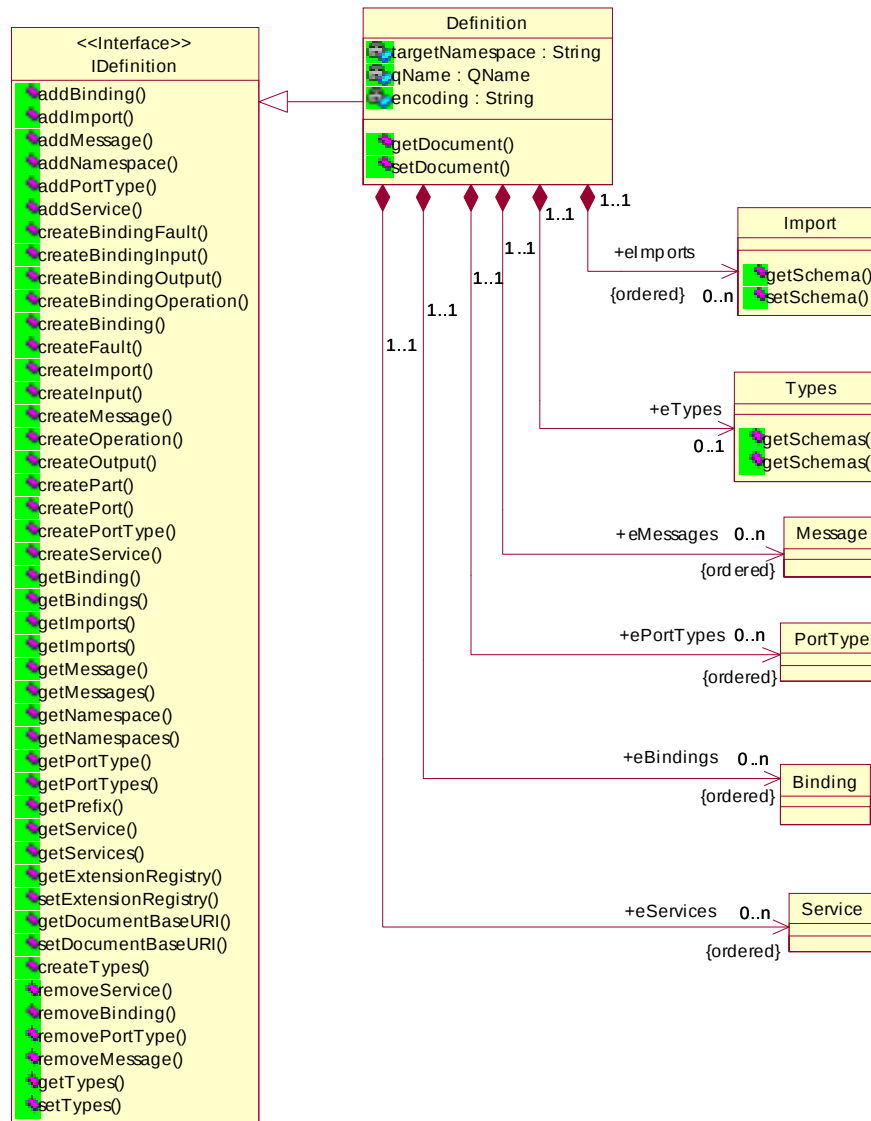
Using MDA to execute the Mapping

- 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

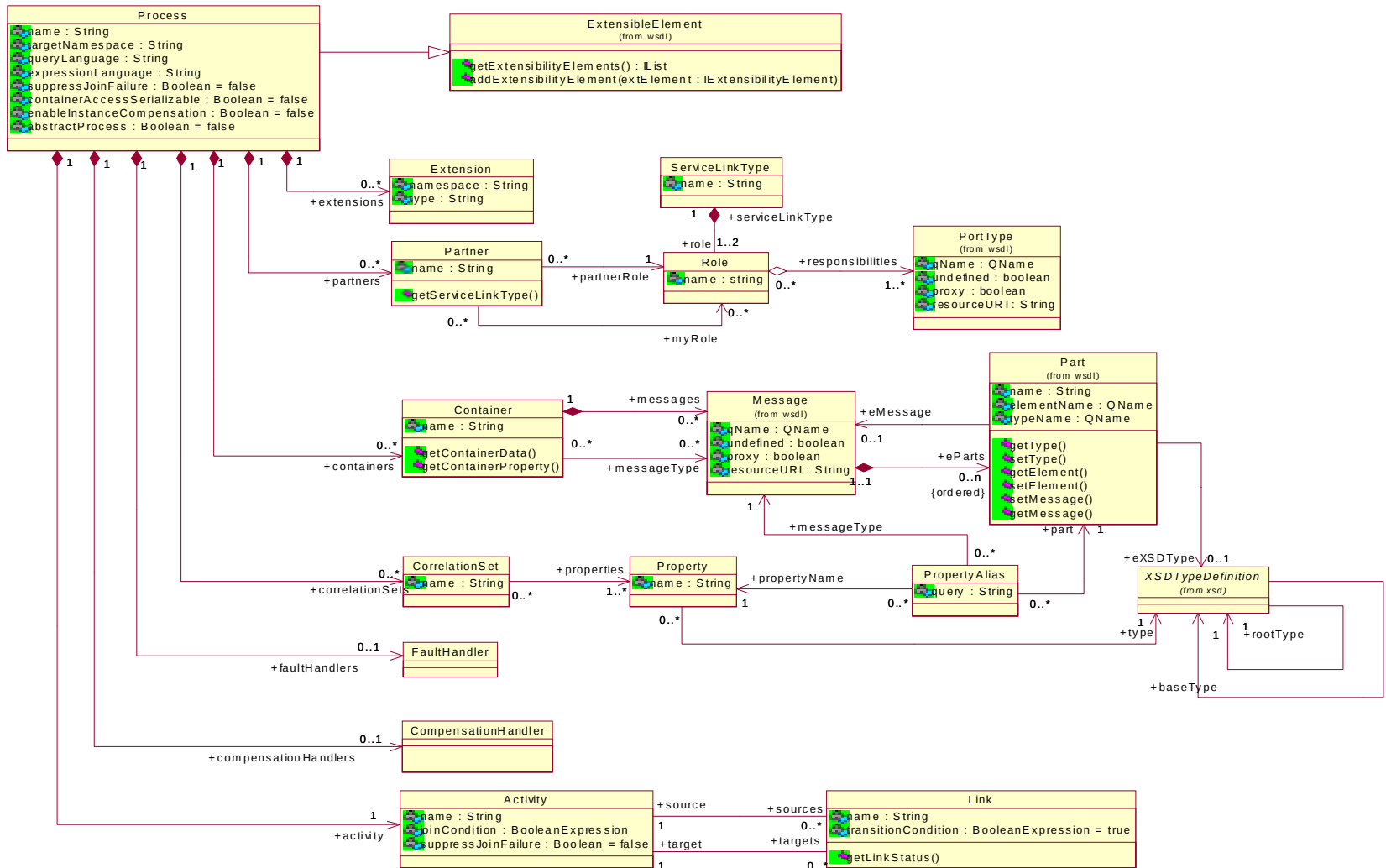
```
+notationDeclarations 0..n
```



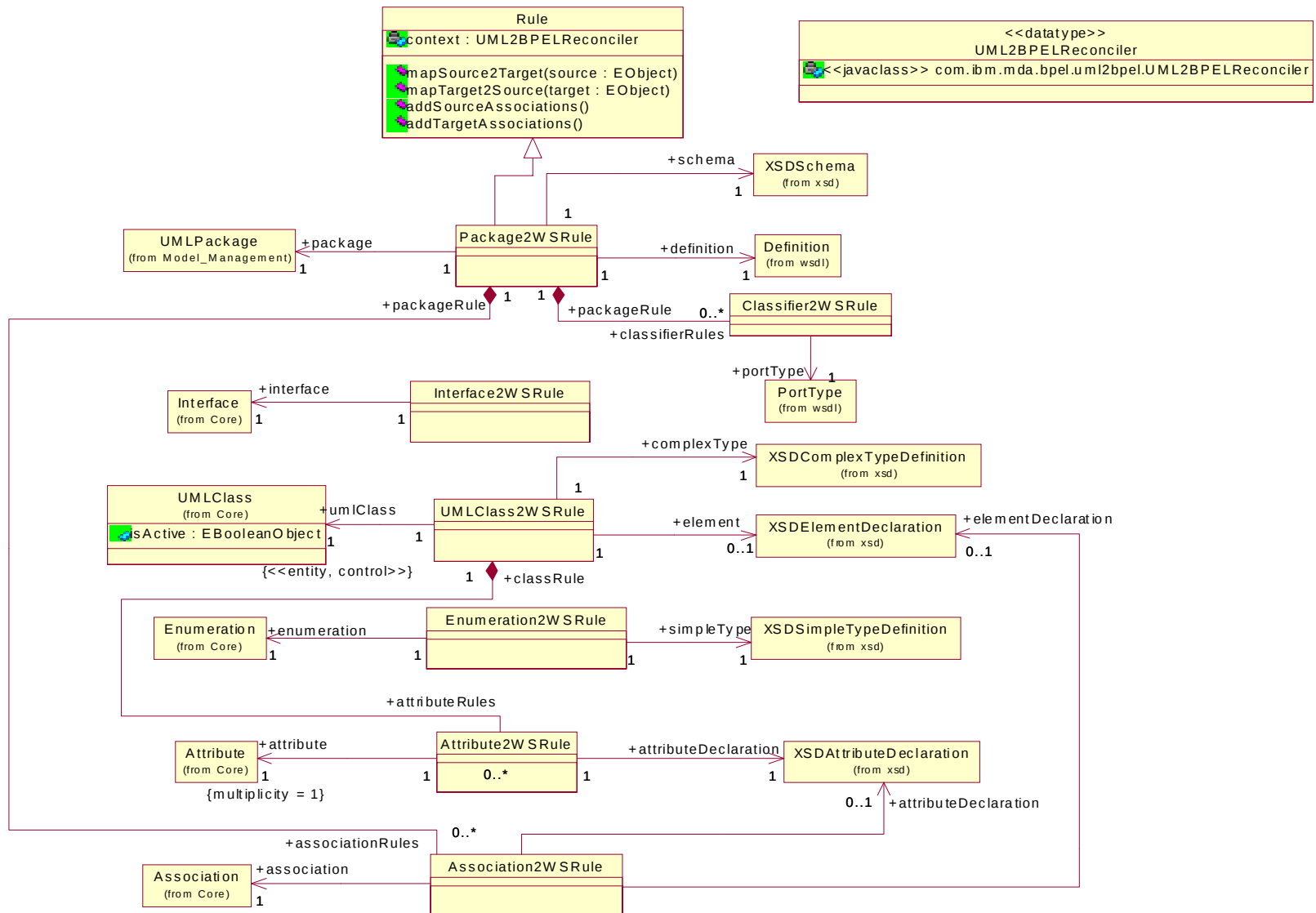
WSDL Model – From WSAD



Draft BPEL Model



UML to BPEL Mapping model



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 architectural context

For More Information

- OMG : www.omg.org
- MDA : www.omg.org/mda
- IBM : www.ibm.com
- IBM Alphaworks : www.ibm.com/alphaworks (UDDI4J)
- 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)