OMG Model Driven Architecture (MDA) meets Web Services

SD Web Services World
August 27-31 2001 : Boston, MA

Sridhar Iyengar
Unisys Fellow
Member, OMG Architecture Board
sridhar.iyengar2@unisys.com
Presentation Overview

- Introduction - The integration problem
- Web Services Architecture Overview
- OMG Model Driven Architecture (MDA) Overview
- Mapping OMG MDA to Web Services Architecture
- OMG Web Services and XML Related Efforts
  - XMI 1.x
  - XMI Production of XML Schemas
  - CWM Web Services RFP
  - Web Services SIG Discussions
- Next Steps
Global services and technology company
– 2000 Revenues : $6.9 Billion
– www.unisys.com

Focus on Enterprise Class Servers and E-Business Solutions around the world

Actively influenced and using several OMG Technologies : UML, XMI, MOF, CWM and MDA

Member of W3C, UDDI Consortium, OASIS...

Using MDA principles to integrate J2EE and .Net environments
OMG History

- 800+ Vendors and End User members
- 1989 OMA Vision & Architecture
- 1991 CORBA 1.0
- 1995 CORBA 2.0 IIOP - CORBA Interoperability
- 1997 UML and MOF adopted, Domain specs begin to be adopted
- 1999 XMI and CORBA Components adopted
- 2000 CWM, XML/Value, EAI, EDOC, XMI for XML Schema, Additional domain specs
- 2001 UML for EDOC, EAI, UML 2.0, MOF 2.0
- 2001 OMG reinvents itself using MDA
‘Muddleware’ Architect’s Dilemma

Middleware (Tuxedo, TIP, DCOM, IIOP, RMI, EJB, COM+, SOAP…)
Information Models (MOF, UML, CWM, SPEM, RSM, BODs, ….)

Distributed, Heterogeneous, Client/Server!

Multiple Clients, Servers, Tools, Apps, O/S, File systems, Databases, Repositories, Data Models, Object Models...
### The Middleware Salad Bar

<table>
<thead>
<tr>
<th>Capability</th>
<th>J2EE</th>
<th>COM+</th>
<th>CORBA/OMA</th>
<th>Web Services</th>
<th>OMG MDA*</th>
<th>.Net</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Layer</td>
<td>TCP/IP</td>
<td>TCP/IP</td>
<td>TCP/IP</td>
<td>TCP/IP</td>
<td>TCP/IP</td>
<td>TCP/IP</td>
</tr>
<tr>
<td>Web Protocol</td>
<td>HTTP</td>
<td>HTTP</td>
<td>HTTP</td>
<td>HTTP</td>
<td>HTTP</td>
<td>HTTP</td>
</tr>
<tr>
<td>Interface Definition</td>
<td>Java</td>
<td>Microsoft IDL</td>
<td>CORBA IDL</td>
<td>WSDL</td>
<td>IDL/XMI/WSDL</td>
<td>WSDL</td>
</tr>
<tr>
<td>Meta Language</td>
<td>XML</td>
<td>XML</td>
<td>XML/XMI</td>
<td>XML</td>
<td>MOF/XML</td>
<td>XML</td>
</tr>
<tr>
<td>RPC Mechanism</td>
<td>RMI</td>
<td>DCOM</td>
<td>IIOP</td>
<td>SOAP; XML</td>
<td>SOAP; IIOP</td>
<td>SOAP</td>
</tr>
<tr>
<td>Registry/Repository</td>
<td>JNDI; LDAP</td>
<td>LDAP; ADSI</td>
<td>Repository</td>
<td>UDDI</td>
<td>MOF;UDDI</td>
<td>UDDI</td>
</tr>
<tr>
<td>Process Flow</td>
<td>Proprietary</td>
<td>Proprietary</td>
<td>Proprietary</td>
<td>ebXML; WSFL</td>
<td>UML</td>
<td>Proprietary</td>
</tr>
<tr>
<td>Modeling Language</td>
<td>UML</td>
<td>UML</td>
<td>UML</td>
<td>UML</td>
<td>UML</td>
<td>UML</td>
</tr>
</tbody>
</table>

*With OMG MDA - Multiple platforms are supported*
A short Introduction to Web Services Architecture

Since this topic has been covered well in this conference, I will focus on the highlights so we can define MDA, cover mapping MDA to Web Services
What are Web Services?

- ‘Modular and reusable’ applications that can be
  - Invoked over the web (using a protocol like SOAP)
  - Described using a service description language (like WSDL)
  - Published into a registry (like UDDI) so the service can be searched, bound and invoked

- Web Services do not imply component based development.
  - In fact a 20 year old TIP or CICS transaction can be wrapped as a web service

- Web Services need to be composable

- Can be simple to very complex; free to expensive
The Web Services Hype

- Microsoft brilliantly got the wagon rolling with the .Net vision
- The early work on SOAP (Simple Object Access Protocol) by Microsoft and Developmentor gradually attracted all the major players
  - IBM jumping on the bandwagon accelerated the momentum
- 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 these technologies
Web Services: The basic idea

Service Provider
Ex: AOL

Service Requestor
Ex: UAL

Service Registry
Ex: Microsoft
UDDI Registry

Bind to Service
Publish Service
Find Service
Web Services: The Stack Evolves

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

Service Flow (IBM WSFL, MSFT XLANG...)

Service Publication and Discovery (UDDI)

Service Description (WSDL)

XML Messaging (SOAP)

Network (HTTP, FTP...)

P.S: There are many flavors of this, many stop at UDDI layer
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 XP 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 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…
  - Services are bound statically at design/development time

- Service sequencing and flows may extend WSDL
Service Flows
- Individual services can be steps in a workflow
- Flows can be graphically modeled
  - Various notations including UML Activity Diagrams have been used
- Service composition needed for multi party service integration

This area does not have as much consensus as lower level protocols

IBM WSFL, MSFT XLANG are among candidate proposals
Microsoft, IBM, BEA, BowStreet, HP, SUN are providing web services integration into their tool sets

Microsoft has focused on the integration and ease of use/deployment the longest with its .Net initiative

J2EE is rapidly embracing and extending Web Service formats and protocols

- JAXR (Java API for XML Registries is an example)
- IBM, BEA, SUN etc have integrated SOAP/WSDL/UDDI into their J2EE application servers

Methodologies, modeling tools and middleware that support the web services life cycle are immature

Back to the Integration Problem!
Solving the Integration Problem
What is needed?

- Short Answer is XML and HTTP!
- Slightly longer answer is SOAP, WSDL, UDDI
- Even longer answer is SOAP, WSDL, UDDI, WSFL, ebXML, XAML, ACML…
- The real answer is
  - We need to think and build software like engineers
  - All these acronyms are great, but where is the process, methodology and architecture!
  - We need to integrate the systems we have with the newer technologies
  - etc.
An Overview of OMG Model Driven Architecture

A vision that uses and integrates the 4M’s - Models, Metadata, Mapping and Middleware of software. MDA allows developers and users to productively design, build, integrate and manage applications throughout the lifecycle irrespective of hardware or middleware platform.
Solving the Integration Problem
What is needed?

- Methodology for building, evolving and integrating software across the life cycle
- A mechanism for cataloging and indexing and searching enterprise assets (metadata and data, internal and external)
- A flexible service based distributed component architecture that spans the enterprise
- A set of shared standard vocabularies (information models) and notation (meaning and context of data usage)
- A metadata driven approach to automate integration between islands of information
- A set of open modeling, metadata and distributed computing standards
The Enterprise Application Life Cycle

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

Discovery and Transformation

Modeling Architecture, Object, Data...

Add Business Logic

Build /Wrap Components

Assemble & Test Components

Configure & Deploy Components

Component Runtimes (EJB, COM+)

Manage Component

Acquired Components

Models, Metadata, Mappings, Middleware

Business Requirements Modeling

Rigorous

Rapid

And do this with quality in a distributed environment
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.

The unification of the original OMG vision of OMA for distributed object interoperability with this ADTF premise gave birth to MDA.
Solution for Managing Complexity: MDA: Model Driven Architecture

- An eclectic 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 $\iff$ 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
Mapping Platform Independent Models to Specific Platforms

PIM
Software Infrastructure

PIM
HealthCare Business Application (CIAS)

J2EE
PIM to PSM Transformation Model

.Net

PSM
J2EE

PSM
.Net
OMG MDA - Details

Community & Enterprise Knowledge Portals

HealthCare  Financial  Manufacturing  Insurance...

E-Business Application Development
E-Business Intelligence, E.I. Portals
E-Business Application Integration

Mappings to Platforms
Information Models, Metadata Formats
Middleware Platforms

Copyright 2001, UNISYS Corporation
OMG MDA - Specific Models

Community & Enterprise Knowledge Portals (KM)

- HealthCare
- Financial
- Manufacturing
- Insurance...

E-Business Application Development (UML, SPEM UML4EDOC)

E-Business Intelligence, E.I. Portals (CWM)

E-Business Application Integration (UML4EAI)

Mappings to Platforms (UML4EDOC, UML4EJB, CWM2XML)

Information Models, Metadata (UML, MOF, XMI, XML...)

Middleware Platforms (J2EE, CORBA, .Net, SOAP...)

Copyright 2001, UNISYS Corporation
OMG Model Driven Architecture
MDA for Enterprise Integration

UML
Model & Design (PIM)

EDOC
UML4EDOC*(PSM)
UML4EAI*(PSM)

Integration for
Vertical Industry Applications
Business Components
Model Driven App Integration
Model Driven App Development
MetaData Management
Business Process Integration

CORBA
UML4CORBA(PSM)

EJB
UML4EJB* (PSM)

*Coming

PIM : Platform Independent Model
PSM : Platform Specific Model

Copyright 2001, UNISYS Corporation
Early examples of MDA

- OMG has been using MDA principles for about 3 years in several specifications

- 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

- Application Development
  - UML for modeling, XMI and IDL for tool integration and interchange
  - UML profile CORBA: UML for modeling, IDL for Integration

- Data Warehousing
  - UML and CWM for Modeling, XMI and IDL for Warehouse integration

- Application Integration
  - UML profile for Enterprise Application Integration (UML4EAI), Distributed Object Computing (EDOC)
OMG Model Driven Architecture
MDA for Metadata Integration

- UML Model & Design (PIM)
- IDL MOF2IDL (PSM)
- XMI MOF2XML (PSM)
- MOF2Java* (PSM)
- JMI MOF2Java* (PSM)
- Vertical Industry Specifications
- Data Warehousing
- B2B Application Integration
- Model Driven App Development
- MetaData Management
- And So On

PIM: Platform Independent Model
PSM: Platform Specific Model

*Coming
The UML is a graphical language for
  - specifying
  - visualizing
  - constructing
  - documenting
the artifacts of software systems

Added to the list of OMG adopted technologies in November 1997 as UML 1.1

Latest revision is UML 1.4
  - UML 1.5 Revision in progress
  - UML 2.0 major enhancement in progress
UML 1.4: Overview

Model business processes and behavior

Model business data/structures
Meta Object Facility (MOF) Overview

- Discover & Manipulate metadata
- Find and Manage Metadata Repositories
- MOF Model
- MOF Facility
- MOF 1.3 Reflective
- Model using UML Class Diagrams precisely

<<OMG Metamodel>>
<<Interfaces>>

Copyright 2001, UNISYS Corporation
OMG MOF Model (UML Subset)
Mapping from PIM to PSM - Simple Example using MOF/XMI

UML Model (PIM)

Auto
- Color : String
- Door : Integer
- Engine : Integer

XMI Document (PSM)

<Auto>
  <Color> Red </Color>
  <Door> 4 </Door>
  <Engine> 2 </Engine>
</Auto>

XMI DTD, Schema (PSM)

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

IDL, Java… (PSM)

interface Auto
Class Auto
  {public String color;
   public int Door;
   public int Engine;
  }
Transform UML models to DTD

XMI 1.1 : ad/99-10-02, ad/99-10-03

XML Syntax and Encoding

MOF Metadata Definitions & Management

UML Model Analysis & Design

XMI MOF TO DTD, XML

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

Validate

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

PSM

UML Models
CWM Models
Travel Documents

UML DTD
CWM DTD
Travel DTD

Transform UML models to DTD
Design-driven XMI for a car

Objects and Designs

Model in XMI

XMI DTD, Schema

XMI Document

Model Interchange

Instance Interchange

<Class>
  <Name>Auto</Name>
  </Class>

<element name="Auto" />
<!ELEMENT Auto (Color*, Door*)>

<Auto>
  <Color>red</Color>
  <Door>2</Door>
  </Auto>
Transform UML Models to Schemas

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

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

XMI 2.0 OMG Document: ad/01-06-12,13
XMI Production of XML Schemas: OMG Vote next week

- Bi-directional mappings from UML/MOF to XML Documents and XML Schema
- Formal metamodels for XML Schema and XML
- Reverse engineering of XML DTDs to assist transformation to schemas and UML
- Integrate the Analysis and Design of XML applications to be better integrated with the web services platform
- Accelerates tool integration and new methods for modeling web services
Transform UML/MOF to Java

**JSR-40 : Java Metadata Interface (JMI)**

- **Java**
  - Syntax and Semantics

- **MOF**
  - Metadata Definitions & Management

- **UML**
  - Model Analysis & Design

- **MOF2 Java**

- **JMI (Many - based on each metamodel Schema)**

- **Java Interfaces**
  - (1 per metamodel used for validation)

- **Java Metamodel Servers**

- **PSM**

**Transform UML/MOF to Java Diagram**

- **Stream : XMI**
- **Instantiate**
- **UML Objects**
- **CWM Objects**
- **Travel Objects**

- **PSM**
- **UML Java I/F**
- **CWM Java I/F**
- **Travel Java I/F**

**UNISYS**

Copyright 2001, UNISYS Corporation
package jmi.eg.simpleRdb;
public interface BaseObjectClass
    extends javax.jmi.reflect.RefClass
{
    public BaseObject createBaseObject()
            throws
            javax.jmi.reflect.JmiException;
    public Table createTable(
            java.lang.String name ) throws
            javax.jmi.reflect.JmiException;
    public column createColumn(
            java.lang.String name ) throws
            javax.jmi.reflect.JmiException;
}
Quick Lessons Learned

- Platforms and protocols come and go
  - Some become ubiquitous over time; most don’t
- Just in the pace of 2 years DTD mania has become XML Schema mania! { YML and ZML are next!}
- We used MDA to map UML/MOF to IDL and XML DTDs
- We used MDA to map UML/MOF to XML Schema
- We used MDA to map UML/MOF to EJB/Java respectively
- We even reverse mapped XML DTD and Documents to UML
- So design your models in UML and transform to DTD, XML Schema, Java, C#, IDL.. Etc.
- Use well defined, standard design patterns to save time
OMG MDA Technologies

- Submissions
- EJB
- Java
- SPE
- UML Profile for EDOC...

- UML
- MOF
- CWM
- CORBA Med
- Life Sciences
- Electronic Commerce

- Evaluating
- Manufacturing

- CIAS
- Enterprise App Integration
- Document Management
- Etc.

- Standards
- MOF
- UML
- CCM
- IDL UML4CORBA
- CWM
We finally apply the MDA principles - Modeling the Web Services Architecture (and its components) so we can formally define web service models with rich semantics (remember the ‘semantic web’) so we can manage the full application life cycle.
How to apply MDA principles to Web Services

- Of course we start by modeling the various parts of the web services architecture
- The key components of the architecture to model are
  - XML itself
  - UDDI
  - WSDL
  - Transformations from UML/MOF to these technologies
- These models of models are called ‘Metamodels’ in OMG parlance
- Having these metamodels and transformation models assists semantic integration of applications and data
Mapping Platform Independent Models to Specific Platforms

PIM Service Registry Model

PIM Service Definition Model

OMA

WSA: Web Services Architecture
OMA: Object Management Architecture
IR: Interface Repository

PIM to PSM Transformation Model

PSM OMA - IR

PSM WSA - WSDL

PSM WSA - UDDI

Copyright 2001, UNISYS Corporation
**Metamodels of XML and XML Schema itself**

- XML and XML Schema Model
  - Defined in the OMG XML Production for XML Schemas proposal
  - OMG Document #: ad/01-06-12,13
  - A complete MOF model using UML notation of XML and XML Schema is available
  - Forward and reverse mappings from UML/MOF to XML and XML Schema have been specified

- Goal is to over time unify multiple schema models such as XML Schema, XMI, RDF etc using MOF

- See also UML profile for XML at xmlmodeling.com
Figure 8-1 XML Schema top level declarations
The top level XML Schema declarations consist of the description of the schema itself (namespace prefix, target namespace, etc.) and the declarations within the schema. These declarations include global scope Attributes, global scope Elements, attribute
Metamodel of UDDI

- Formally defining a metamodel for UDDI will make it more easy to integrate the structure and semantics of other directories and registries.

- An alternative approach is to define a generic interface (JAXR - Java API for XML Registries) - but this is not likely to capture the richer semantics in models.

- Models of UDDI are being developed at IBM, Unisys and Microsoft (among other vendors).
Metamodel of UDDI - Subset

This is only a subset of the UDDI metamodel. A complete model will be in an upcoming OMG Proposal.
See xmlmodeling.com for...
Formally defining a metamodel for WSDL will make it more easy to integrate the structure and semantics of other interface definition languages (IDL, Java…)

A formal model (MOF metamodel) of WSDL is being defined by IBM and Unisys for an OMG submission

– The CWM Web Services RFP - so that it becomes easier to integrate analytical web services with the rest of the database/data warehouse integration infrastructure

This model can be integrated with additional metamodels of Java and IDL that have already proposed in OMG specifications
WSDL Metamodel Fragment

Metamodel: Courtesy IBM
How do we use these metamodels of WSDL, UDDI and XML!

- With in the OMG suite of metamodels, UML is the most well known and used to integrate application development and deployment tools.
- CWM - Common Warehouse Metamodel is more recent, but is a landmark model that addresses the integration requirement of various types of databases (network, relational, XML, multidimensional etc.,)
- The OMG is now in the process of integrating these suite of metamodels and the information they represent into the world of web services.
- A quick peek at CWM will show us the way!
OMG Common Warehouse Metamodel (CWM)

- Scope
  - Data Warehouse lifecycle metadata interchange and management

- Initial submission: IBM, Unisys, NCR, Hyperion, Oracle, Genesis, UBS, Dimension EDI...
  - Metamodel - Single logical & physical!
  - Generated XML DTDs
  - Generated MOF - IDL mappings
  - Generated XML document

- Approved by OMG Board in June 2000
- CWM 1.1 Revision is being now worked on
CWM 1.0 Overview {02/2001}
Common Warehouse Metamodel

Warehouse Management
- Warehouse Process
- Warehouse Operation

Analysis
- Transformation
- OLAP
- Data Mining
- Information Visualization
- Business Nomenclature

Resources
- Object-Oriented (ObjectModel)
- Relational
- Record-Oriented
- Multi Dimensional
- XML

Foundation
- Business Information
- Data Types
- Expressions
- Keys Index
- Type Mapping
- Software Deployment

ObjectModel
(Core, Behavioral, Relationships, Instance)
### CWM Data Resource Matrix

<table>
<thead>
<tr>
<th>Resource</th>
<th>Intent</th>
<th>Extent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Oriented</td>
<td>Package</td>
<td>Extent</td>
</tr>
<tr>
<td>Relational</td>
<td>Class</td>
<td>Object</td>
</tr>
<tr>
<td>Record</td>
<td>Attribute</td>
<td>Data Value</td>
</tr>
<tr>
<td>Multi-dimensional</td>
<td>Column</td>
<td>Data Value</td>
</tr>
<tr>
<td>XML</td>
<td>RecordSet</td>
<td>Field Value</td>
</tr>
<tr>
<td></td>
<td>Record</td>
<td>Member</td>
</tr>
<tr>
<td></td>
<td>Dimension</td>
<td>Member Value</td>
</tr>
<tr>
<td></td>
<td>Dimensioned</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ObjectOriented</td>
<td></td>
</tr>
</tbody>
</table>

This matrix illustrates the relationship between different types of data resources and their corresponding intents and extents. The columns represent the categories of resource, intent, and extent, while the rows categorize the specific elements within each category.
Modular Design
- Minimum dependencies
  - Cross package services provided by links to UML
- Avoid subpackages
- Reduced complexity, improved understanding
- Use only the packages you need
CWM 1.0 - Resources

- Relational
- Multidimensional
- XML
- Essbase
- IMSDatabase
- DMSII
- Express
- COBOLDat
MDA for Data Warehouse Integration using CWM
What next for CWM: CWM Web Services

- CWM is now being used by IBM, Oracle, Unisys, Hyperion and many others for integrating enterprise data, metadata and content

- However this interchange even though it uses XML (CWM is XMI compliant and uses XML for its interchange) has the following problems
  - The interchange patterns are too fine grained
  - The interchange patterns need to be formalized between the integration partners
  - The interchange protocol does not support web services

- To address these needs OMG is standarding ‘CWM for Web Services’
CWM Web Services RFP

- Requires Integration of
  - CWM Metamodel
  - XMI - The XML DTDs and Schemas that conform to CWM
  - WSDL
  - SOAP
  - UDDI

- A key principle of OMG MDA is define formal models (platform independent and platform specific) of these technologies and their transformations

- The suite of metamodels and transformations being proposed will integrate MDA and Web Services
CWM Web Services RFP -
A Potential Solution

- The specific interchange patterns model will be specified using UML and MOF and this metadata interchanged using XMI

- Mappings from UML/MOF to WSDL and UDDI will make it possible to
  - Design web services applications and components using UML
  - Manage all related metadata using MOF/XMI and XML
  - Automatically publish the interfaces/service definitions in CORBA IR, UDDI registry and JNDI/LDAP servers!
  - Maintain traceability across the life cycle (try this today!)

Please note that the final adopted proposal may be different - Too early to tell!
MDA meets Web Services bringing some order to the web services Chaos!

WSA: Web Services Architecture
OMA: Object Management Architecture
IR: Interface Repository

PSM OMA - IR
PSM WSA - WSDL
PSM WSA - UDDI

PIM Service Registry Model
PIM Service Definition Model
UML 1.4 App Design
OMA to PSM Transformation Model
UML4CORBA
UML4WS
UML4UDDI

Copyright 2001, UNISYS Corporation
The Enterprise Application Life Cycle

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

Models, Metadata, Mappings, Middleware

Rigorous

Rapid

Needs to work for Web Services too - MDA Enables this!
Who is Implementing key MDA standards?

- IBM VisualAge for Java, WebSphere, Rose tool kit
- IBM VisualWarehouse, Hyperion, Oracle, SAS
- Rational Software (Unisys XMI for Rose); SUN Forte
- Microsoft VisualStudio.Net
- DSTC, OMEX, Hyperion
- Oracle Designer, Meta Integration, Together/J, Objecteering, ObjectsByDesign, Unisys, Aonix, Webgain
- OMG standard metamodels and DTDs (MOF, UML, CCM, CWM*, Java*, EJB*…) 
- Metadata Coalition voted overwhelmingly to build on OMG specifications for metadata (September 2000)
- Java Community Process JSR-40 : Java Metadata Interface*
- Java Community Process JSR-69 : Java for OLAP*
OMG UML/XMI/MOF Usage

- UML
- MOF
- XMI
- XML

- Submissions
- EJB
- Java
- SPEM
- UML Profile for EDOC...

- Evaluating
- MDC OIM
- CORBA Med
- Life Sciences
- Electronic Commerce

- CIAS
- Enterprise App Integration
- Document Management
- Etc.

- Standards
- MOF
- UML
- CCM
- IDL UML4CORBA
- CWM

Copyright 2001, UNISYS Corporation
Concluding Thoughts

- Ensuring a *unified model and metadata driven architecture* is key to solving the heterogeneous integration problem
  - Use models and metadata to enable transformation
  - Use MDA principles throughout the life cycle
  - OMG MDA is the realization of this vision

- Most customers have and will continue to have components and information/data from multiple sources and formats that need integration

- XML/HTTP, CORBA/EJB, DCOM/ActiveX, MOF/XMI (Metadata), UML (Modeling) and CWM (Data Warehousing) need to work together with the content (business models, BODS, HL7 RIM…)

- As new platforms arrive, MDA can map to them, so your investment is protected. MDA is being mapped to Web Services
For More Information

- MDA : www.omg.org/mda
- Unisys : www.unisys.com
- W3C : www.w3c.org
- DSTC : www.dstc.edu.au
- Sridhar : sridhar.iyengar2@unisys.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