Integrating Enterprise Information Resources
(using OMG CWM, XMI and MOF)

OMG E-Business Integration Workshop
January 29 2001 : Orlando, Florida

Sridhar Iyengar
Unisys Fellow
Member, OMG Architecture Board
sridhar.iyengar2@unisys.com
(949) 380-5692
Tutorial Overview

- EDOC, E-Business & Data Warehousing
- OMG Modeling and Metadata Architecture
  - Overview of MOF, XMI and UML
- E-Business Intelligence Architecture
- CWM Overview
- CWM Details
- CWM and MDC OIM Status
- Putting it all together
Acknowledgements

- Portions of this presentation have been derived from
  - OMG CWM presentation from the CWM team
  - My earlier tutorials/conference presentations
- Unisys corporation for the support and resources in defining and validating the CWM metamodel specification
- CWM team for their phenomenal work over the last 2 years!
- EDOC 2K for its ever expanding tutorial program
Global services and technology company
  – 1999 Revenues : $7.54 Billion
  – www.unisys.com

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

Actively influenced and using several OMG Technologies : UML, XMI, MOF, CWM etc.
OMG History

- 800+ Vendors and End User members
- 1989 OMA Vision & Architecture
- 1991 CORBA 1.0
- 1995 CORBA 2.0 IIOP - CORBA Interoperability
- 1997 MOF and UML 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, UML 2.0, Better XML and E-Business integration
The Data Integration Problem
(Cannot consistently use the data we have!)

- Heterogeneity
  - Data Sources and Targets, Middleware, OS
  - Tools, Technologies

- Complexity of data
  - Structured (Largely relational)
  - Unstructured
  - Multidimensional

- Federated data sources and targets

- Central Data Warehouse VS Subject Data Marts

- Transformation

- Systems Management etc. etc.
Solving the Integration Problem

What is needed?

- Methodology for building, evolving and integrating software
- 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
Relevance to EDOC

- Historically EDOC has focused on
  - Object and component middleware
  - Enterprise modeling and integration (including metamodeling)

- But
  - Has not paid much attention to data integration and database/warehouse middleware

- In reality
  - Data Integration is an ‘enterprise problem’
  - Data Integration on the Internet is even more painful
  - We can apply to ‘enterprise data’ some of the lessons from ‘enterprise objects’
E-‘Muddleware’ Architect’s Dilemma

Distributed, Heterogeneous, Client/Server!

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

DataMarts  
SQL/Server

Warehouse Builder  
Oracle Apps

Unisys LINC,  
DMSII

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

Warehouse  
NCR  
Teradata

Siebel

SAP  
BI Warehouse

Microsoft SQL Server

BEA  
Weblogic

Multiple Clients, Servers, Tools, Apps, O/S, File systems, Databases, Repositories, Data Models, Object Models...
OMG E-Business Integration

Bird’s eye View

Community & Enterprise Information Portals (KM...)

HealthCare

Financial

Manufacturing

Insurance...

E-Business Application Development
UML, CCM

E-Business Intelligence
CWM

E-Business Application Integration
CCM, EJB, COM, XML

Information Models, Components and Metadata (XMI, MOF...)

Directory, Security, Database, Web, Transaction, Caching, Metadata, Services..

Distributed Runtime Middleware (IIOP, XML/Value, HTTP...)

(C) Iyengar 2000
The E-Business Application Life Cycle and the Process

Architecture Centric, Business driven, Iterative and Incremental

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, Components, Middleware

Rigorous
WebTime!

And do this with quality in a distributed environment
Example Tool Integration Scenario

Visual Development Environments & Legacy Wrapping

EJB/COM+ Generation & Deployment

Web App Srvr or COM+

Component Management

Business, Component and Data Modeling

Traceability Links

Life cycle component Management
Full Traceability, Tool, AppServer independent

WEBDAV*

SCC

MOF/JMI

XMI

XMI/EJB

XMI/UML

* In development
Tutorial Overview

- EDOC, E-Business & Data Warehousing
- OMG Modeling and Metadata Architecture
  - Overview of MOF, XMI and UML
- E-Business Intelligence Architecture
- CWM Overview
- CWM Details
- CWM and MDC OIM Status
- Putting it all together
Metadata Architecture Evolution

Components
Frameworks
E-Business

Y2000, Data
Warehouse

Data
Administration

Hierarchical,
Monolithic
E/R Models
Proprietary Interchange
IBM, Burroughs

Relational
Client/Server
E/R Models
CDIF Interchange
Platinum/CA, ViaSoft, MSP...

OO, O-R
Relational, XML
Distributed Object
Object Models (UML, MOF)
XML Interchange (XMI)
Unisys, IBM, MSFT...

1980 1990 2000

META MODEL EXTENSIBILITY

Copyright 2000, UNISYS Corporation
There is more & more metadata lurking everywhere!

**REPOSITORY**

Person Table

Created-by User: John Doe

Meta Data

Person Table

Space

Archive

Last good Archive 9/28/1998: 2 AM

Employee Record

Name | Char | Meta Data
---|---|---
John | M | 43
Mary | D | 27
Bill | W | 62
George | S | 18

Marital Status | Enumeration (M, D, W, S)
---|---
Age | Integer

**Employee Record**

LastName | String | Meta Data
---|---|---
First Initial | String
Employee Number | Number
E-Business Architectures: 2001

The Component Bus
CORBA/CCM; Java RMI/EJB; COM+; XML/XMI/HTTP

- Application Models/Interfaces
- Domain/Business Models (I/F)
- Technology Models (I/F)

Object Adapter
APIs

Packaged Applications Tools
Manufacturing, Finance...
Business Object Initiative

UML, BO, CWM...

Distributed Component Services
Trading, Transactions, Agents...

Component Repository
Component & Application Management

MOF
XMI
XML
OMG Modeling and Metadata Framework
OMG Modeling and Metadata Framework

UML
Model & Design

XMI
MOF2XML DTD
MOF2XML Doc
MOF2XML Schema*

IDL
MOF2IDL

JMI*
MOF2Java

* Coming

Meta Manage

Vertical Industry Specifications

Data Warehousing

B2B Application Integration

Model Driven App Development

MetaData Management

And So On
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

Most recent minor revision is UML 1.3 (November 1999)
UML 1.3: Overview

Model business processes and behavior

Model business data/structures
**OMG Metamodelling Layers**

- **OMG Terms**
  - User Objects Layer (M0)
  - Model Layer (M1)
  - Metamodel Layer (M2)
  - Meta-Metamodel Layer (M3)

- **Sample Objects**
  - User Objects: <Acme_Software_Share 98789>, 654.56, sell_limit_order, <Stock_Quote_Svr 32123>
  - Model Layer: StockShare, askPrice, sellLimitOrder, StockQuoteServer
  - Metamodel Layer: UML::Class, Attribute, CWM::Table, CWM::Cube
  - Meta-Metamodel Layer: MOF::Class, MOF::Attribute, MOF::Operation

- **XML Terms**
  - User Objects: XML Documents, Business data
  - Model Layer: Business Models, DTDs, App Schemas
  - Metamodel Layer: CWM, EAI, IDL, UML, DTD/Schema
  - Meta-Metamodel Layer: Middleware Schema
Metamodel Architecture

M3 Layer: Specifies meta-metaclasses for the UML metamodel
M2 Layer: Specifies metaclasses for the UML metamodel, such as Class
M1 Layer: Specifies classes for the UML user models, such as Passenger, Ticket, TravelAgency
M0 Layer: User objects that are instances of UML user model classes, such as instances of Passenger, Ticket, TravelAgency
MOF Overview

■ Foundation for OMG Metadata and Modeling architecture
  – Model, design and implement metamodels, (meta)model servers
  – Provides 4 generic meta-object interfaces for introspection (Used by all MOF based Meta-models...)
  – Provides MOF-IDL mapping to automate generation of concrete object interfaces for specific metamodels
  – Provides MOF-XML mappings to automate generation of XML DTDs and Documents (XMI specification)
  – Provides MOF-Java (JMI*) mappings to automate generation of Java classes and interfaces
  – Can also be used with COM using COM/CORBA Interoperability software

■ MOF uses UML for notation and design
  * JMI : Java Metadata Interface in development
Meta Object Facility (MOF)
Architecture

Discover & Manipulate metadata

<<CORBA IDL Module>>
MOF 1.1 Reflective

<<OMG Meta-metamodel>>
MOF Model

Find and Manage Metadata Repositories

Model using UML Class Diagrams precisely

MOF Facility
Reflective module of the MOF

- All meta-models based on the MOF inherit this package.

- This is the foundation for introspection in the OMG metadata architecture.
XML Metadata Interchange (XMI)

- Use W3C Extensible Markup Language (XML) for the transfer syntax and interchange format
  - Specify XML Document Type Definitions (DTD) to enable transfer and verification of
    - UML based models (using UML DTD)
    - Data Warehouse models (using CWM DTD)
    - Java and EJB artifacts (using Java and EJB DTDs)
    - MOF based metamodels and their instances (using MOF DTD)
- Specify a precise MOF to XML mapping
  - Allows interchange of any MOF based metamodel and corresponding models (MOF--> XML Stream)
  - Enables automatic generation of XML documents that can be validated with generated DTDs
- Use UML and MOF for metamodel design and implementation
- Designed by Unisys, IBM, Oracle, DSTC et al
UML 1.1 XML DTD Subset


<!ATTLIST Class XMI.element.att; XMI.link.att;>
<!ELEMENT name (#PCDATA | XMI.reference)*>
<!ELEMENT feature (Feature| StructuralFeature| Attribute| BehavioralFeature| Operation| Method Reception)*>
A Simple Model

Let us pretend this is a trivial (meta)model. MOF and XMI work for models and metamodels.
<Model xmi.id="a1" name = "Business" visibility ="public">
  <ownedElement>
    <Class xmi. id="a7" name = "Customer">
      <feature>
        <Attribute name = "id">
          <multiplicity>
            <Multiplicity lower = "1" upper = "1"/>
          </multiplicity>
        </Attribute>
        <Operation name = "update" scope ="instance"/>
      </feature>
    </Class>
  </ownedElement>
</Model>
XMI/MOF Tools
Interoperability Options

MOF Introspection
write

Model Transfer
DTD Generation
Schema Generation *
(XMI)

Concrete
Model/Metadata Access

Intermediate Stream/File (XML)

Generated Interfaces
for Meta(model)

* Coming in XMI

Copyright 2000, UNISYS Corporation
How to use MOF/XMI

- Define the domain or technology specific model
  - Middleware models are usually called metamodels by OMG
  - Use UML - only knowledge of Class modeling needed to get started on the design

- Export model to XMI processor which generates
  - XML DTDs for the specific (meta) model
  - XML documents that conform to the DTD

- Manage the Models, DTDs and documents in a MOF/XMI compliant distributed repository

- Use XMI toolkits from IBM, Unisys, DSTC and others

- For generating federated (using CORBA) MOF servers use toolkits from DSTC, Unisys etc.
XMI - Automobile Example (simplified)

UML Model

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

XMI Document

\[
\text{<Auto>}
\text{\<Color> Red \</Color>}
\text{\<Door> 4 \</Door>}
\text{\<Engine> 2 \</Engine>}
\text{</Auto>}
\]

IDL, Java...

interface Auto

Class Auto

{public String color;
 public int Door;
 public int Engine;
}

XMI DTD, Schema

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

- EDOC, E-Business & Data Warehousing
- OMG Modeling and Metadata Architecture
  - Overview of MOF, XMI and UML
- E-Business Intelligence Architecture
- CWM Overview
- CWM Details
- CWM Status
- MDC OIM Status
- Putting it all together
**Data Warehousing / Business Intelligence Life Cycle**

1. **Data and Metadata Discovery**
2. **Domain Models, Transforms**
3. **Extract, Cleanse, Transform, Load**
4. **Build Warehouse Components**
5. **Assemble Warehouse Components**
6. **Configure & Deploy Mart/Whse**
7. **Manage Warehouse**
8. **Warehouse Middleware**
9. **Acquired Components Data Sources**

**Database, Warehouse and Metadata Middleware**

**BI : Business Intelligence**
The Federated Business Intelligence Architecture

- Grounded in the following technologies
  - Metamodeling architecture: OMG MOF
  - Metadata interfaces: OMG MOF IDL, JavaSoft JMI*
  - XML Metadata Interchange: OMG XMI, W3C XML
  - Data Warehousing Metamodel: OMG CWM
  - Data Access Interfaces: JDBC, ODBC, OLE/DB, JavaSoft JOLAP*, Java for Data Mining*
  - Distributed Object and Data Access middleware

- Designed by experts in databases, data warehouses, metadata management and distributed computing

* JMI and JOLAP are being defined as part of the Java Community Process
Tutorial Overview

- EDOC, E-Business & Data Warehousing
- OMG Modeling and Metadata Architecture
  - Overview of MOF, XMI and UML
- E-Business Intelligence Architecture
- CWM Overview
- CWM Details
- CWM Status
- MDC OIM Status
- Putting it all together
OMG Common Warehouse MetaModel

- **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 Finalization Task Force in Progress
CWM History - Highlights

- IBM, Oracle, and Unisys propose a Common Warehouse Metadata Interchange (CWMI) RFP to the OMG (6/10/98)
- CWMI RFP issued by the OMG (9/18/98)
- Co-submitting team formed by IBM and Unisys, later joined by Oracle, Hyperion, Genesis, NCR, UBS, and Dimension EDI (1/1/99)
- Liaison between OMG and MDC is formed following Microsoft submission of OIM to Meta Data Coalition
- An initial CWM Specification was jointly submitted by the team (9/17/99)
- CWM XMI Interoperability demo : 11/99
- A final CWM Specification was jointly submitted for evaluation and comments (2/11/00)
The OMG ADTF votes unanimously to recommend adoption of the CWM Specification (3/10/00)

The CWM FTF was formed to finalize the CWM Specification based on public comments and implementation feedback (3/10/00)

The OMG Board adopts CWM (6/19/00)

CWM FTF (Chair Dan Chang, IBM) in process, completion expected 11/00

CWM implementations underway at IBM, Oracle, Unisys, Hyperion, SAS, MITI etc.
CWM Submitters & Supporters

- Submitters
  - IBM
  - Unisys
  - Oracle
  - NCR
  - Hyperion
  - UBS AG
  - Genesis Development
  - Dimension EDI

- Supporters
  - Deere & Co.
  - Sun Microsystems
  - Hewlett-Packard
  - Data Access Technologies
  - InLine Software
  - Aonix
  - Hitachi

- CWM Specs: [http://www.omg.org](http://www.omg.org)
- CWM Info: [http://www.cwmforum.org](http://www.cwmforum.org)
The CWM based Integration

Figure 1. Data warehouse source and target arrangements.

Pairwise (9 connections)  
Hub (6 connections)
A complete specification of the **syntax and semantics** needed to **export/import** shared warehouse metadata and the common warehouse metamodel, including:

- **The CWM Metamodel** (Volume 1)
- Interchange format for shared warehouse metadata (**CWM DTD**, Volume 2)
- Interchange format for the CWM Metamodel (**CWM XML**, Volume 2)
- Access API for shared warehouse metadata (**CWM IDL**, Volume 2)

[http://www.omg.org/technology/technology/cwm/]
Roles of UML in CWM

- The *metamodeling language* (as in the MOF Model)
  - UML Semantics, UML Notation, OCL
- The *foundation metamodel*
  - UML Foundation, Common_Behavior, and Model_Management packages
- The *object (resource) metamodel*
  - Same as above
CWM 1.0 Overview {02/2000}

Common Warehouse Metamodel

Warehouse Management

Warehouse Process

Warehouse Operation

Analysis

Transformation

OLAP

Data Mining

Information Visualization

Business Nomenclature

Resources

Object-Oriented (UML)

Relational

Record-Oriented

Multi Dimensional

XML

Foundation

Business Information

Data Types

Expressions

Keys Index

Type Mapping

Software Deployment

UML 1.3

(Foundation, Behavioral_Elements, Model_Management)
CWM 1.0 Model - Top Level

The major packages in CWM:

- org.omg.uml { UML 1.3 }
- org.omg.cwm { CWM Core }
- org.omg.cwmX { Extensions }
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 Extension Packages**

**Extension Techniques**
- Metamodel specialization
- XMI extensions
- UML tagged values
CWM 1.0 - Overview
CWM 1.0 - Resources

- Relational
- Multidimensional
- XML
- Essbase
- Express
- IMS Database
- DMSII
- COBOL Data
CWM Relational Metamodel (Partial)
CWM 1.0 - Analysis
CWM Transformation

- General mechanism describing data movement and lineage
- Generic transformations from any physical object to any other physical object
- Maps logical structures and concepts in the warehouse onto physical implementation
- Provides for multiple physical implementations of logical structures and concepts
CWM 1.0 - Management
CWM Generation

- **Metamodel**
  - Single logical source
  - Multiple packages
- **Generated for each package**
  - XML document
  - CWM DTD
  - CWM IDL

Used Unisys UREP/CIM for IDL and XML Generation
XMI for E-Business Intelligence
November 1999 OMG Demo
CWM Case Study at UBS: The Business Problem

- UBS: One of the largest banks in the world
- Legacy system needs to be migrated to new CBD-based system
- Legacy system uses
  - COBOL record data models
  - Entity-Relationship models
- CBD system uses
  - UML object models
  - Different Entity-Relationship models
**UBS: A CWM-Based Prototype Solution**

- Use CWM to place legacy models (COBOL record models and ER models) in a repository
- Use CWM to place the new system’s object models and ER models in the repository
- Use CWM to define the transformations, and place models of the transformations in the repository
  - CWM’s transformation capabilities are key
  - A CWM transformation engine can execute the transformation without additional programming by reading CWM-based transformation models
- UBS validated CWM by building repositories for these models
UBS Migration Repository: Overview

(Courtesy Hans-Peter Hoidn UBS)
Standardization Benefits for UBS

- No lock-in to specific vendors
- Models can be exchanged among different vendors’ tools
- Leverages OMG’s Meta Object Facility (MOF)
  - Allows CWM data models and UML object models to be handled in an integrated fashion (UML is also based on the MOF)
- MOF options for exchanging models among tools
  - XMI Leverages W3C XML standard
  - CORBA/Java
Tutorial Overview

- EDOC, E-Business & Data Warehousing
- OMG Modeling and Metadata Architecture
  - Overview of MOF, XMI and UML
- E-Business Intelligence Architecture
- CWM Overview
- CWM Details
- CWM and MDC OIM Status
- Putting it all together
Roles of UML in CWM

- The *metamodeling language* (as in the MOF Model)
  - UML Semantics, UML Notation, OCL

- The *foundation metamodel*
  - UML Foundation, Common_Behavior, and Model_Management packages

- The *object (resource) metamodel*
  - Same as above
CWM Metamodel Details : Part 1

Foundation
and
Data Resources
Package Architecture

Modular Design

- Minimum dependencies
  - Cross package services provided by links to UML
- Avoid subpackages
- Reduced complexity, improved understanding
- Use only the packages you need
The modeling environment

- UML notation used as diagramming technique
- UML metamodel extended to support warehouse concepts
**Foundation**

Metamodels shared by other packages

- **Foundation**
  - Business Information
  - Data Types
  - Expressions
  - Keys & Indexes
  - Software Deployment
  - Type Mapping

<table>
<thead>
<tr>
<th>Warehouse Process</th>
<th>Warehouse Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformation</td>
<td>OLAP</td>
</tr>
<tr>
<td>OLAP</td>
<td>Data Mining</td>
</tr>
<tr>
<td>Data Mining</td>
<td>Information Visualization</td>
</tr>
<tr>
<td>Information Visualization</td>
<td>Business Nomenclature</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Management</th>
<th>Analysis</th>
<th>Resource</th>
<th>Foundation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| UML 1.3 (Foundation, Behavioral_Elements, Model_Management) |
Business Information

- Responsible parties & their contact information
- Documentation and general commentary
- Hierarchies of business types can be constructed
Data Types

- Supporting classes for modeling of data type systems
- Extent class represents collections of instances
**Keys & Indexes**
- Shared by several data resource models
- Promotes similar representation across models
Expressions

- Tree-structured, functional model of expressions
- Full access to all CWM objects
- Supports both “black box” and “white box” expressions
- Use “white box” expressions for interchange and lineage
Software Deployment

- **SoftwareSystem**
  - A software package on a CD

- **A Deployed Software System is a set of DeployedComponents**
  - An installed SoftwareSystem

- **Each Deployed Component is on a specific Machine**
  - An installed program
Software Deployment

- DataManagers provide access to data
  - A deployed DBMS
- DataProviders wrap other DataManagers as specified by ProviderConnections
  - ODBC and JDBC drivers
Foundation

• **Type Mapping**
  – Map types to corresponding types in other systems
  – Designed for simple data type exchanges
  – Use Transformations for more complex mappings

**Type Mapping**
- Map types to corresponding types in other systems
- Designed for simple data type exchanges
- Use Transformations for more complex mappings
Data Resources

Describe logical and physical data containers

- Operational sources
- Warehouse targets
- Logical models
# Data Resource Matrix

<table>
<thead>
<tr>
<th>Resource</th>
<th>Intent</th>
<th>Extent</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Object Oriented</strong></td>
<td>Package</td>
<td>Extent</td>
</tr>
<tr>
<td></td>
<td>Class</td>
<td>Object</td>
</tr>
<tr>
<td></td>
<td>Attribute</td>
<td>Data Value</td>
</tr>
<tr>
<td><strong>Relational</strong></td>
<td>Catalog/Schema</td>
<td>RowSet</td>
</tr>
<tr>
<td></td>
<td>Table</td>
<td>Record</td>
</tr>
<tr>
<td></td>
<td>Column</td>
<td>Field Value</td>
</tr>
<tr>
<td><strong>Record</strong></td>
<td>RecordFile</td>
<td>RecordSet</td>
</tr>
<tr>
<td></td>
<td>RecordDef</td>
<td>Record</td>
</tr>
<tr>
<td></td>
<td>Field</td>
<td>Field Value</td>
</tr>
<tr>
<td><strong>Multi-dimensional</strong></td>
<td>Schema</td>
<td>MemberSet</td>
</tr>
<tr>
<td></td>
<td>Dimension</td>
<td>Member</td>
</tr>
<tr>
<td></td>
<td>Dimensioned Object</td>
<td>Member Value</td>
</tr>
<tr>
<td><strong>XML</strong></td>
<td>Schema</td>
<td>Document</td>
</tr>
<tr>
<td></td>
<td>ElementType</td>
<td>Element</td>
</tr>
<tr>
<td></td>
<td>Attribute</td>
<td>Data Value</td>
</tr>
</tbody>
</table>

Copyright 2000, UNISYS Corporation
Data Resource Packages

- Relational
  - RDBMS catalogs & ODBC/JDBC client catalog views
  - SQL-99 compliant
Data Resource Packages

- Record
  - Basis for traditional databases & files
  - Self-describing, delimited, & fixed-offset supported
Data Resource Packages

- Multidimensional
  - Physical representation of multidimensional databases
Data Resource Packages

• XML
  – Supports XML 1.0
  – Basis for XML documents
  – Allows use as sources and as targets
CWM Metamodel, Part 2

Analysis
# Data Analysis

Describes production & analysis of warehouses

- Describe analytical & deployment structures
- Design data movement & transformations
- Deployable on a number of data resources

<table>
<thead>
<tr>
<th>Management</th>
<th>Warehouse Process</th>
<th>Warehouse Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis</td>
<td>Transformation</td>
<td>OLAP</td>
</tr>
<tr>
<td>Resource</td>
<td>Data Mining</td>
<td>Information</td>
</tr>
<tr>
<td>Foundation</td>
<td></td>
<td>Visualization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Business</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nomenclature</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Object (UML)</th>
<th>Relational</th>
<th>Record</th>
<th>Multi-Dimensional</th>
<th>XML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Information</td>
<td>Data Types</td>
<td>Expressions</td>
<td>Keys</td>
<td>Mapping</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

UML 1.3
(Foundation, Behavioral_Elements, Model_Management)
Transformation

- General mechanism describing data movement and lineage
- Generic transformations from any physical object to any other physical object
- Maps logical structures and concepts in the warehouse onto physical implementation
- Provides for multiple physical implementations of logical structures and concepts
Transformation

- Support for “White-Box” transformation mappings

- Leverages the UML hierarchy: Mapping of Classifier-to-Classifier and Feature-to-Feature

- Mapping of Classifier-to-Feature
Transformation

- Transformations can be specified for arbitrary model elements
- Reification of the transformation “process”
- Relates to Warehouse Process and Warehouse Operation metamodels
OLAP

- Analytical model: Cubes, measures, dimensions, attributes, levels, and hierarchies
- Cubes contain multiple measures and are implemented via cube regions
OLAP

- Dimensions: Multiple levels, attributes, and hierarchies
- Levels are used in multiple hierarchies; support a subset of the dimension attributes
Data Mining

- Models the fundamental meta data necessary for constructing and managing Data Mining models
- Three conceptual areas: Model, Settings and Attributes
Data Mining

- Models major settings types: Statistical, Clustering, Association Rules, Supervised

- Relates settings to specification and attributes
Information Visualization

- CWM core metamodel for information visualization and publishing
- Separation of “logical” rendered object from rendering “transformation”
- Recursive/composite structuring
Business Nomenclature

- CWM metamodel for “Business Metadata”
- Intended for Data Warehousing and Business Intelligence domains
- Common business terms and concepts
- Used in conjunction with analysis and reporting tools
Business Nomenclature

- Taxonomies consist of concepts
- Glossaries consist of terms
- Taxonomies and Glossaries can be associated
- All are relevant to some “business domain”
Warehouse Management

Orchestrates warehouse activities

- Warehouse Process relates
  - Transformations
  - Triggering events

- Warehouse Operation logs
  - Transformation activity
  - Metrics

<table>
<thead>
<tr>
<th>Management</th>
<th>Warehouse Process</th>
<th>Warehouse Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis</td>
<td>Transformation</td>
<td>OLAP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data Mining</td>
</tr>
<tr>
<td></td>
<td>Information</td>
<td>Visualization</td>
</tr>
<tr>
<td></td>
<td>Nomenclature</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource</th>
<th>Object (UML)</th>
<th>Relational</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Record</td>
<td>Multi-Dimensional</td>
</tr>
<tr>
<td></td>
<td>XML</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Foundation</th>
<th>Business Information</th>
<th>Data Types</th>
<th>Expressions</th>
<th>Keys Index</th>
<th>Type Mapping</th>
<th>Software Deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UML 1.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(Foundation, Behavioral Elements, Model Management)</td>
</tr>
</tbody>
</table>

UML 1.3

Copyright 2000, UNISYS Corporation
A Warehouse Process identifies warehouse tasks and the events that trigger them (“what gets done”)

Relates Transformation Activity to Event types
Warehouse Operation

- Activity and Step Execution track details of executions of Transformations (tracks “what got done when”)
- Step Execution may be related to a UML CallAction model element
Warehouse Operation

- ChangeRequest objects represent a proposed change or one that has been implemented or rejected.
- Measurement objects can hold values for any object (such as volumetric details)
CWM Generation, Validation, and Extension
CWM Generation

- Metamodel
  - Single logical source
  - Multiple packages
- Generated for each package
  - XML document
  - CWM DTD
  - CWM IDL

The Definitive CWM MetaModel

CWM Packages

<<metamodel>>

Package XML Files (MOF Model DTD)

Apply UML to MOF Transformation Rules

CWM Team

CWM Repository Server

CWM CORBA IDL Files

CWM Packages

CWM DTD Files

IDL In/Out

In/Out XML

Document Type

XMI Interfaces
**CWM Development**

- **Metamodel Proposal**
  - An information metamodel is proposed to be included in the submission
  - Group analysis to determine if the metamodel should become part of CWM

- **Model Team**
  - A team is formed to develop a metamodel as part of the CWM submission

---

**Information Metamodel for the CWM team to consider**

- **Metamodel Accepted**
- **Metamodel Rejected**

---

**Information Covered**

- Information
- Outside Scope

---

Copyright 2000, UNISYS Corporation
CWM Development

- Metamodel Development
  - Metamodels constructed by domain knowledgeable modelers
  - A CWM Metamodel is constructed by analyzing common portions of existing Warehouse tool models as well as public reference models
  - Tool logical models can be reconstructed using CWM as common starting point
Use Case Scenarios

- Develop use cases of the individual models
- Develop representative use cases that involve several packages used in combination
- Develop use cases that represent entire target tools
CWM Validation

Validation

- Each Use Case defines a “slice” through the model being validated
- Use Cases represent “typical” problems that the model must solve

![Diagram of CWM Validation process]

- OLAP Model
- Deployment Model

Semantic Mapping

Abstraction Refinement
CWM Extensions

- CWM provides interchange of the common portions of warehouse tool meta models
- CWM should be used as the new foundation of a warehouse tool model
- Volume 3 of the Specification contains examples of CWM Extensions
CWM Extensions

- Extensions
  - Published tool specific information for the purpose of interchange
  - Physical characteristics of databases not common in general but interchangeable in a heterogeneous environment
CWM Extensions

- Extensions
  - Common ancestry in base metamodel
  - Tool-specific default structures (e.g., Database)
Extending CWM

- Extending the model to redefine a working tool metamodel without generating a new DTD.
  - Categorized into three generic types:
    - Proprietary Attributes: Tool specific definitions not intended for interchange
    - Proprietary Associations and or Classes: Tool specific areas not common and not intended for interchange
    - Sharable Extensions: Tool specific definitions proposed for interchange via tagged value pairs
Extending CWM

- Extending the model to redefine a working tool metamodel without generating a new DTD.
  - Proprietary Attributes: Tool specific definitions not intended for interchange
  - Proprietary Associations and or Classes: Tool specific areas not common and not intended for interchange
  - Sharable Extensions: Tool specific definitions proposed for interchange via tagged value pairs
Extending CWM

- Extending the model to redefine a working tool metamodel without generating a new DTD.
  - Proprietary Attributes: Tool specific definitions not intended for interchange
  - *Proprietary Associations and or Classes: Tool specific areas not common and not intended for interchange*
  - Sharable Extensions: Tool specific definitions proposed for interchange via tagged value pairs
Extending CWM

- Extending the model to redefine a working tool metamodel without generating a new DTD.
  - Proprietary Attributes: Tool specific definitions not intended for interchange
  - Proprietary Associations and or Classes: Tool specific areas not common and not intended for interchange
  - **Sharable Extensions: Tool specific definitions proposed for interchange via tagged value pairs**

![Diagram of CWM and ToolX relational models with columns and attributes]
CWM
Summary and Actions
A common specification that defines, in UML, the structure and semantics of shared metadata in data warehousing and business intelligence

- **Resource**: Object, Relational, Record, Multidimensional, XML
- **Analysis**: Transformation, OLAP, Data Mining, Information Visualization, Business Nomenclature

A common specification that defines, in XML, the interchange format and, in IDL, the access API for such shared metadata
**CWM Specification:**

*CWM XML, CWM DTD, CWM IDL*

- CWM Metamodel (in UML Notation)
- MOF DTD
- CWM XML
- CWM DTD
- CWM IDL
- CWM Metadata Interchange (in XML)
- CWM Metadata Access
CWM Extensions (CWMX)

- Published vendor specific metamodel for the purpose of metadata interchange (Volume 3 & Volume 4, non-normative)
  - Common ancestry in the CWM metamodel
  - Demonstrates the validity of the CWM metamodel
  - Demonstrates the extensibility of the CWM metamodel
Tutorial Overview

- EDOC, E-Business & Data Warehousing
- OMG Modeling and Metadata Architecture
  - Overview of MOF, XMI and UML
- E-Business Intelligence Architecture
- CWM Overview
- CWM Details
- CWM and MDC OIM Status
- Putting it all together
CWM Status

- CWM FTF in progress since 3/2000
- Weekly/bi-weekly teleconferences
- Mail : cwm-ftf@omg.org
- Next revision 2/2001
- Interoperability showcase scenario defined
- CWM implementation in process
- Preparations for incorporating MDC OIM requirements underway
  - Following decision by MDC to continue metadata standardization at OMG
MDC votes to build on OMG CWM

- Microsoft joins MDC and transfers OIM (12/1998)
- OMG and MDC have established liaisons with each other
  - OMG/MDC joint meetings in 1999, 2000
- Some progress, efforts move in parallel
  - MDC already using OMG UML as a foundation for its Open Information Model
  - OMG CWM spec uses OIM as design reference and incorporates some features
  - OIM 1.x to incorporate some CWM enhancements
  - OMG XMI based DTDs for MDC OIM available
- MDC votes to continue metadata standardization at OMG (9/8/2000) instead of proceeding in parallel!
**JSR-40 : Java Metadata Interface (JMI)**

- A new Java Community Process standardization efforts
  - Design well underway
  - Progress report at Java One in June 2000
  - Public Review planned for 4Q2000

- Builds on OMG MOF, XMI and UML specifications

- Supports JSR-26 : UML-EJB Mapping specification being developed in parallel

- A pure Java metadata specification

- Will be used in
  - JSR-69 : Java for OLAP
  - JSR-73 : Java for Data Mining
  - JSR-26 : UML profile for EJB

- CWM to be used in the Java Business Intelligence platform
Tutorial Overview

- EDOC, E-Business & Data Warehousing
- OMG Modeling and Metadata Architecture
  - Overview of MOF, XMI and UML
- E-Business Intelligence Architecture
- CWM Overview
- CWM Details
- CWM and MDC OIM Status
- Putting it all together
MOF, XMI for Integration

- Use UML for analysis and design of metamodels, Information Model etc. (these are models after all!)
- Define and manage metamodels/profiles using the MOF
  - Use UML based modeling tools, or MOF interfaces (normative) or XMI
  - Enables life cycle meta data interoperability and design reuse across metamodels
  - Relationships and Subtyping across metamodels supported by the MOF
- Use XML for exchanging metadata via OMG XMI
- Use MOF-IDL mappings for concrete IDL interfaces to metamodels {JSR-40 Java Metadata Interface in future}
- Inherit MOF Reflective interfaces for interoperable meta objects across meta-models
Importance of (Meta)modeling using MOF, XMI and UML

- Provides a regular and rigorous infrastructure at a higher level of abstraction
- Furnishes an architectural basis for extensions and evolution of software
- Facilitates alignment with other standards that use a metamodel architecture
  - Potentially eliminate or evolve redundant standards (e.g., CDIF now endorses XMI)
- Supports interoperability and integration across domains at the semantic level
- Use UML to design metamodels and models, MOF to implement and manage them and XMI to interchange them over the Internet
Importance of CWM

- The industry standard for data warehouse interchange and interoperability
- Allows integration of enterprise information so that we can better leverage the data we have
- Already supports the most widely used data base and file systems
- Extensible to vendor/customer specific needs
- Builds on UML, XMI and MOF
- Accelerates data warehousing, portal and business intelligence market
Who is Implementing XMI, MOF, UML and CWM?

- IBM VisualAge for Java, WebSphere, Rose tool kit
- IBM VisualWarehouse, Hyperion, Oracle, SAS
- Rational Software (Unisys XMI for Rose)
- DSTC, OMEX...
- Oracle Designer, Meta Integration, Together/J, Objecteering, ObjectsByDesign, Aonix…
- 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*
OMG UML/XMI/MOF Usage

- Submissions
- EJB
- Java
- SPE
- UML Profile for EDOC...
- Evaluating
- MDC OIM
- CIAS
- Enterprise App Integration
- Document Management
- Etc.
- Standards
- MOF
- UML
- CCM
- IDL UML4CORBA
- CWM
- CORBA Med
- Life Sciences
- Electronic Commerce
Concluding Thoughts

- Ensuring a **unified model driven distributed meta object architecture** is key to solving the heterogeneous integration problem.

- 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…).

- Modeling and Metadata matters - Master it: Use it in for your middleware, use it for your domains.

- CWM is the key to integrating Enterprise Information.
For More Information

- Unisys : www.unisys.com/marketplace/urep
- 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