OMG Meeting
Tutorial I: Specification Overview

Written and Presented by
Jon Siegel, Ph.D.
Director, Technology Transfer
Object Management Group
siegel@omg.org
1-781-444-0404
Copyright License Agreement

Information

As a member benefit, OMG will grant permission to use the complete CORBA Tutorial slide set to member organizations that agree to adhere to the terms and restrictions stated in our CORBA tutorial copyright license agreement. Members may use the complete set of materials for educational and training purposes as long as they remain a member in good standing of the OMG consortium.

For information on how to obtain your copy of this CORBA tutorial package, contact the OMG main number at +1-781-444-0404 or send email to siegel@omg.org
Section I: Introduction to OMG’s Specification Suite

• Why Distributed Computing?
• Scope and Applicability of OMG Specifications
Why Distributed Computing?

- Distributed Hardware needs Distributed Software!
Future Networks
Focus on Interoperability

- There will *not* be consensus on hardware platforms;
- There will *not* be consensus on operating systems;
- There will *not* be consensus on network protocols;
- There will *not* be consensus on application formats.

There *must* be consensus on interoperability.
Today’s Architecture

Clients

Internet/Intranet

Phone line

Wireless

Type-Specific Servers

Content Management

Business Logic

Data Layer

Legacy

OTM
Who’s Using CORBA?

• Wells-Fargo Bank
  – Integrate all legacy systems
  – Home Banking, ATMs

• More Banks & Financial Companies:
BankBoston 1998 ROI Study

• CORBA Project ROI - Raw Figures:
  – Quantified Return-on-Investment for EMSTR Analytics was 627%;
  – Payback period was 7.3 months.

• More payoff than could be quantified:
  – More timely and more accurate information to the traders
  – New and deeper analytics.
  – Making the bank’s analytics system available to its customers

• These benefits overwhelm the quantified benefits
More CORBA Users...

- **UK Immigration Department**
  - Suspect Index System

- **CNN Interactive**
  - News feeds from hundreds of sources on multiple machine types and formats are managed with CORBA

- **Pratt & Whitney**
  - Program Planning and Control for jet engine production

- **Matra Datavision**
  - Integration of EUCLID QUANTUM software for CAD/CAM

- **Aircraft Manufacture: Boeing, Airbus**
Still more CORBA Users...

- AWACS Systems Integration
  - Also US Air Force and Navy
- Retail: The Gap; Home Depot
- Transportation: DHL, Fedex, Sabre
  CargoManager, German Railway Company,
  Port of Singapore
Who Makes/Sells ORBs?

- There are over 70 ORBs on the Market
- From different types of companies:
  - System Vendors
  - ORB Vendors
  - Integrated Services Vendors (e.g. ORB-based Transaction Systems)
  - Free ORBs from Universities and Independents
- A Thriving Market, Started by OMG
Testing, Certification

• OMG/Open Group Testing/Certification
  – Announced 5/99: CORBA 2.1 now, 2.3 soon
  – 3 Certified ORBs so far:
    • Fujitsu, AT&T OmniORB, MICO
  – Test Suite Partially funded by ESPRIT
• CORBA.net (www.corba.net)
  – Web-based interoperability demo
• DOPG, Japan, tested Interoperability
  – ORBs and Transaction Systems
  – Fourteen ORBs shown to interoperate
  – Four OTS ImplS shown to interoperate
    • 1-Phase & 2-Phase commit and rollback
From Design to Deployment

Support for All your Business Computing

Modeling Environment:
- Data Model
- Appl Model
- Repository

Applications

Domain Frameworks:
- Mfg: PDM
- Fin: Acctg

Distributed Infrastructure:
- Platforms, Languages, Networks, Protocols, Middlewares, Messaging
From Design to Deployment

Distributed Infrastructure:
Platforms, Languages, Networks, Protocols, Middlewares, Messaging

Support for All your Business Computing
It’s the Infrastructure

- Platform and Language Independence
- Heterogeneous Interoperability
- Leverage Infrastructure & Legacy
- Server Requirements
- Essential Services
- Enterprise Requirements
- Specialized Markets
It’s the Infrastructure

- Platform and Language Independence
- Heterogeneous Interoperability
- Leverage Infrastructure & Legacy
- Server Requirements
- Essential Services
- Enterprise Requirements
- Specialized Markets

- Object Request Broker Architecture
- ISO-Standard Interface Definition Language
- Mappings to Programming Languages
It’s the Infrastructure

- Platform and Language Independence
- **Heterogeneous Interoperability**
- Leverage Infrastructure & Legacy
- Server Requirements
- Essential Services
- Enterprise Requirements
- Specialized Markets

- Leverages IDL and the ORB Architecture
- Adds GIOP and IIOP ISO-Standard Protocols
- Bridging to Other Networks & Protocols
- Multi-Protocol Object Addressing
It’s the Infrastructure

- Platform and Language Independence
- Heterogeneous Interoperability
- Leverage Infrastructure & Legacy
- Server Requirements
- Essential Services
- Enterprise Requirements
- Specialized Markets

* CORBA 3 Technology

- **Java/EJB**: Forward/Reverse Mapping*, Objects over the Wire*; EJB & CORBA Components*
- **XML/SOAP**: XML/Value Mapping*; Static & Dynamic Representations; Protocol Mappings
- **COM/DCOM**: COM/CORBA Bridge, Both Ways, since 1995
- **Legacy Wrapping**, popular & successful technique
It’s the Infrastructure

- Platform and Language Independence
- Heterogeneous Interoperability
- Leverage Infrastructure & Legacy
- Server Requirements
- Essential Services
- Enterprise Requirements
- Specialized Markets

* CORBA 3 Technology

- **Scalability:** Portable Object Adaptor Infrastructure
- **Fault Tolerant:** FT CORBA* Specification, Standardizes Entity Redundancy
- **Component-Based:** CORBA Component Model*, a Wrapper for EJBs, and More
It’s the Infrastructure

- Platform and Language Independence
- Heterogeneous Interoperability
- Leverage Infrastructure & Legacy
- Server Requirements
- Essential Services
- Enterprise Requirements
- Specialized Markets

* CORBA 3 Technology

- **Directory**: CosNaming, CORBA Trader (an ISO standard)
- **Distributed Events**: Event Service, Notification Service adds Filtering, QoS Control
- **Persistence**: CORBA Persistent State Service*
It’s the Infrastructure

• Platform and Language Independence
• Heterogeneous Interoperability
• Leverage Infrastructure & Legacy
• Server Requirements
• Essential Services
• Enterprise Requirements
• Specialized Markets

* CORBA 3 Technology

- **Transactional**: 2-Phase Commit & Rollback, OMG Standard since 1995
- **Secure**: Mature CORBA Security Architecture, SSL and CSIv2 Protocols, Firewall*
- **Messaging***: Two Asynchronous Modes, Time-Independent Invocations, “Guaranteed” Delivery
- **Quality of Service Control***: Control Timing and Priority of Invocations and Replies
It’s the Infrastructure

- Platform and Language Independence
- Heterogeneous Interoperability
- Leverage Infrastructure & Legacy
- Server Requirements
- Essential Services
- Enterprise Requirements
- Specialized Markets

- Leverage Standard Architecture and Infrastructure
- **Real-Time:** RT CORBA * Specification for End-to-End Predictability
- **Minimum CORBA**: Standard Small-Footprint CORBA for embedded and card-format systems

* CORBA 3 Technology
Who’s Using CORBA?

“A surprising 70% of respondents cited CORBA compliance as *important* or *very important* to integration, outpacing every other factor in the survey, including core functions such as integration of legacy applications with distributed systems and corporate intranets.”

Summary of 547 responses rating middleware selection criteria, reported in
Who’s Developing in CORBA?

In a recent survey of 50 firms:

- 44% develop using Java and CORBA
- 24% develop using COM/DCOM
- 12% develop using both
- 16% use neither
- 4% don’t know

Internet Middleware
Forrester, July 1999
Why Develop with CORBA?

“Standish Research shows that those projects which include the creation of custom middleware have a zero chance of being completed on time and on budget.”

Karen Boucher
The Standish Group
January, 2001
From Design to Deployment

Domain Frameworks:
- Mfg: PDM
- Fin: Acctg

Distributed Infrastructure:
Platforms, Languages, Networks, Protocols, Middlewares, Messaging

Support for All your Business Computing
From Design to Deployment

Domain Frameworks:
- Mfg: PDM
- Fin: Acctg

Support for All your Business Computing
OMG in Vertical Markets (I)

- **Finance/Insurance**: General Ledger, Contract/Agreement, Currency
- **Electronic Commerce**: Negotiation, Resource Registration & Discovery, PKI Management
- **Healthcare**: Person Identification, Clinical Data Access, Record-Based Access Control, more
- **Telecommunications**: TMN, IN, Event QoS, Wireless Protocol, Event Logging
OMG in Vertical Markets (II)

- **Transportation:** Air Traffic Control, Intelligent Highways, Rail Traffic Integration
- **Manufacturing:** PDM, CAD/CAM, Simulation
- **Life Sciences Research:** Genomic Maps, Biomolecular Sequence Analysis
- **Utilities:** Data Access
- **Space & Ground Systems:** Software-Defined Radio
OMG in Vertical Markets (III)

• **C4I:** Military & Emergency Management Logistics

• **HR Management:** Human Resources for Large Organizations

• **Analytical Data Management:** Statistics & Data Analysis

• **Retail Systems**

• **ECIS & Call Center**
From Design to Deployment

Modeling Environment:
- Data Model
- Appl Model
- Repository

Domain Frameworks:
- Mfg: PDM
- Fin: Acctg

Distributed Infrastructure:
Platforms, Languages, Networks, Protocols, Middlewares, Messaging

Support for All your Business Computing
From Design to Deployment

Modeling Environment:

- Data Model
- Appl Model
- Repository

Support for *All* your Business Computing
OMG Modeling Support

- **Unified Modeling Language UML 1.3**
  - World Standard for A&D
  - Representation for Structure, Dynamics, Deployment
- **MOF: Meta-Object Facility**
  - Integrated Repository
  - Standard MetaModel
- **XMI: XML Metadata Interchange**
  - Model & MetaModel Interchange
  - XML-Based Format, including DTDs
- **CWM: Common Warehouse Metamodel**
  - Data Warehousing Integration
  - Record, Table formats; Data Loading & Transformation
Model Driven Architecture

• Proposed OMG Future Direction

• At Its Core, a Middleware – Independent UML Model
Model Driven Architecture

- With This Model –
- Implement in Any Middleware
Model Driven Architecture

- Services Based on Middleware-Independent Model
- Accessible from Any Application
Model Driven Architecture

- Domain Facilities Also Model-Based

- Implement in, or Access From, Any Middleware Environment
Companies gain the ultimate in flexibility: the ability to regenerate applications from a stable, platform-independent model as the underlying infrastructure shifts over time. ROI flows from the reuse of application and domain models across the software lifespan.
MDA in Vertical Markets

- Document Design & Interface Semantics
- In the Common MDA Environment
- That Supports Multiple Platforms, Languages, Networks
From Design to Deployment

Modeling Environment:
- Data Model
- Appl Model
- Repository

Distributed Infrastructure:
- Platforms, Languages, Networks, Protocols, Middlewares, Messaging

Domain Frameworks:
- Mfg: PDM
- Fin: Acctg

Applications

Support for All your Business Computing
Section 2: Supporting Analysis & Design

• UML: The Unified Modeling Language

• The MOF: Meta-Object Facility

• XMI: XML Metadata Interchange
Big Software Projects…

- are like Buildings – they have a structure with many interlocking parts
- You wouldn’t contract to build a skyscraper without seeing plans first:
  - Elevations
  - Interior Views
  - Site Plan
  - Blueprints
  - Floor Plans
  - Structural Analyses
- Large Software Projects deserve the same treatment
- Better Time and Cost Estimates; Less Risk
OMG Metadata before MOF

• An Example: Three places to store Metadata about Objects in your System:
  – Naming Service
  – Trader Service
  – Interface Repository
• But no explicit Metadata Architecture
• MOF defines modeling primitives
  – MOF::Class (MetaClass)
  – MOF::Attribute (MetaAttribute)
4-Layer Metamodel

- Describes Modeling Concepts in a Domain
- Arrow denotes "instance-of" dependency

User Objects Layer (M0)

Model Layer (M1)

Metamodel Layer (M2)

Meta-MetaModel Layer (M3)

<PG Shares> 56 3/8 [User Data]

StockShare AskPrice [Domain Model, CBO, …]

UML::Class Attribute, Operation [UML, CWM, …]

MOF::Class MOF::Attribute [MOF]
What is the MOF?

- The MOF defines an abstract model called a meta-metamodel
- The MOF specification defines a standard distributed repository:
  - That is, a set of modeling constructs and IDL interfaces to define and manipulate a set of interoperable metamodels
- With UML and XMI, an integral part of a complete suite of modeling tools
What is this good for?

• Every development environment is built on a meta-model:
  – Languages like C++, Java, Smalltalk, etc.
  – Environments like CORBA, COM, CICS, etc.
• You need to consider this when you pick a modeling tool
  – Specialized tools have only limited use
  – Generalized tools may not constrain to models implementable in your development environment
• You may already have, or need, multiple modeling tools
  – Use XMI to transfer models among them, mapping from one meta-model to another
  – Store your models in the standard MOF repository regardless of their tool of origin, or meta-model
• Interfaces for *reflective* and *introspective* functions let objects or applications examine their meta-data
  – Take advantage of modeling to design and implement more flexible, powerful applications
UML – a Graphic Language for

- Visualizing
  - Using the standardized graphic UML displays
- Specifying
  - Semantics to define
    - static structure
    - dynamic behavior
    - model organization
- Constructing
  - Map UML to Programming Environment and Generate some code Automatically
- Documenting
  - Every phase of lifecycle from analysis and design through deployment and maintenance
The UML Specification defines

- **UML Semantics**
  - Defined using a metamodel

- **UML Notation Guide**
  - Defines a graphic syntax for UML semantics

- **UML Standard Profiles**
  - Extensions for SW development and business modeling

- **UML CORBA facility Definition**
  - A standard repository for UML models
  - Supports XMI

- **Object Constraint Language**
  - A standardized constraint language
UML Semantics Metamodel

• Foundation: Structural Diagrams – static structure
  – Class Diagram
  – Object Diagram
  – Component Diagram
  – Deployment Diagram

• Behavior: Behavioral Diagrams – dynamic behavior
  – Use Case Diagram
  – Sequence Diagram
  – Collaboration Diagram
  – Statechart Diagram
  – Activity Diagram

• Model Management Diagrams – organization
  – Packages
  – Subsystems
  – Models
Class Diagram

- Each block denotes a class
- A class has
  - Attributes – characterizing objects of the class
  - Operations – to manipulate the attributes, or perform other actions
- Classes may be associated in various ways:
  - Associations
  - Generalizations
  - Dependencies
  - Refinements

<table>
<thead>
<tr>
<th>Store</th>
</tr>
</thead>
<tbody>
<tr>
<td>Totals: long</td>
</tr>
<tr>
<td>POSlist: List</td>
</tr>
<tr>
<td>Login()</td>
</tr>
<tr>
<td>getPOStotals(Totals:long)</td>
</tr>
<tr>
<td>updateStore(Totals:long)</td>
</tr>
</tbody>
</table>
XMI: XML Metadata Interchange

- **XMI Integrates 3 key industry standards:**
  - XML
  - MOF
  - UML

- The XMI Specification consists of:
  - Set of XML DTD production rules for transforming MOF-based metamodels into XML DTDs
  - Set of XML document production rules for encoding and decoding MOF-based metadata
  - Design principles for generating XMI-compliant DTDs
  - Design principles for generating XMI-compliant XML documents
  - Concrete XML DTDs for MOF (to exchange metamodels) and UML (to exchange models)
The Metadata Problem

CWM Addresses a Problem Facing Every Enterprise:

• Many Databases
• Many Repositories
• Many Schemas Describing the “Same” Data
• Moving Data Requires Manual Schema Transformation
CWM Integrates your Data

- Integrates Existing Data Models
- Maps to Existing Schemas
- Supports Automated Schema Generation
- Supports Automated Database Loading
- The Basis for Data Mining and OLAP

Across the Enterprise
CWM Defines Metamodels for:

- CWM Foundation
- Relational Data
- Record Data
- Multidimensional Data
- XML Data
- Data Transformations
- OLAP
- Data Mining
- Info Visualization
- Business Nomenclature
- Warehouse Process
- Warehouse Operation
# The CWM Metamodel

<table>
<thead>
<tr>
<th>Warehouse Process</th>
<th>Warehouse Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformation</td>
<td>OLAP</td>
</tr>
<tr>
<td>Data Mining</td>
<td>Information Visualizatin</td>
</tr>
<tr>
<td>Business Nomenclature</td>
<td></td>
</tr>
<tr>
<td>Multidimensional</td>
<td>XML</td>
</tr>
<tr>
<td>XML</td>
<td></td>
</tr>
<tr>
<td>Business Information</td>
<td>Data Types</td>
</tr>
<tr>
<td>Expression</td>
<td>Keys, Indexes</td>
</tr>
<tr>
<td>Type Mapping</td>
<td>Software Deployment</td>
</tr>
</tbody>
</table>

| UML 1.3 (Core, Common_Behavior, Model_Management) |
Section 3: Object Basics

- Introduction to Objects
- OMG Interface Definition Language (IDL)
Object Technology

- OMG’s membership believes that an approach based on object technology simplifies the problem:
  - Offers a single view of a distributed, heterogeneous system.
  - Four keys to object orientation help integration of distributed systems: Encapsulation, Polymorphism, Inheritance and Instantiation.
What is an Object?

• An Object - -
  – Combines Functionality and Data
  – Typically represents a real-world object
  – Has a well-defined interface
  – and an “object reference” or address
  – Follows basic OO principles:
    • Encapsulation
    • Polymorphism
      • Inheritance
      • Instantiation
CORBA Features

- Transparencies:
  - Programming language
  - Platform/vendor
  - Operating System
  - Location
  - Network HW/SW
- Dynamic binding *and* typing
- in an Open Specification with multivendor support

- Object Orientation
  - Encapsulation
  - Polymorphism
  - Inheritance
  - Instantiation
- Extended services
  - Naming/trader
  - Events/notification
  - Transactions
  - Security, domains
IDL ISOLATES INTERFACE FROM IMPLEMENTATION

Client Side

Object Implementation Side

Object Request Broker
Define the Object's Interface in OMG IDL:

Operations Parameters, Types Exceptions

This Defines a language-independent API for the Object.
IDL ISOLATES INTERFACE FROM IMPLEMENTATION

Objects may be written in programming languages -- C, C++, Java, Smalltalk, Ada, COBOL, Visual Basic, or . . .
IDL ISOLATES INTERFACE FROM IMPLEMENTATION

might be Wrapped Legacy Applications, Tool-Generated Objects, or Objects purchased from Vendors.
IDL ISOLATES INTERFACE FROM IMPLEMENTATION

The same IDL defines the Client Side API.
Just like object implementations, Clients may be programmed by hand, generated by tools, or purchased from vendors.
Role of OMG IDL

Client Side

Object Implementation Side

Standard Protocol

ORB

COBOL

C

C++

Ada

Smalltalk

More

IDL
OMG/ISO IDL

OMG IDL (Interface Definition Language) Separates the Interface from the Implementation:

- multiple-inheritance, strongly typed, public interface specification language;
- independent of any particular language/compiler;
- mappings will be provided for many languages/compilers;
- not a programming language.

Enables Interoperability

Supports the Dynamic Request Mechanism
Roles of Infrastructure

- Provides a Local, Well-Known Point of Contact for *All* Object Invocations a Client may make
- Passes invocation to Local or Remote target Object Implementation
- Understands IDL; Maintains Repository of available Object Interfaces
- Also Maintains Repository of Available Implementations
- Federates this information across System

A WEB OF INTERCONNECTED ORBs
Section 4: CORBA Interoperability Basics

- Interoperability via ORB-to-ORB communication
- IIOP: OMG’s Standard Protocol
- CORBA and non-standard protocols
- CORBA, OLE, and COM/DCOM
- CORBA, Java, and the Web
- Scalable CORBA Servers
ORB to ORB Interoperability
ORB to ORB Interoperability
CORBA Interoperability

CORBA 2.0 Interoperability Comprises:
- An overall architecture for CORBA-CORBA communications;
- An API for adding bridges;
- A general multi-transport message format (General Inter-ORB Protocol or GIOP);
- An API for gateways using ESIOPs -- (Environment-Specific Inter-ORB Protocols)

**UNIVERSAL, OUT-OF-THE-BOX INTEROPERABILITY:**

- IIOP - that is, GIOP over TCP/IP - is mandatory for compliance, either internally or via a bridge;
- Specialized protocols are optional and well-supported by the specification.
Asynchronous/Messaging Spec

• Four extensions to the Architecture.
  CORBA 3 Clients can:
  – make requests which do not block;
  – make requests which do not complete during the lifetime of the client;
  – control QoS associated with a request;
  – control ordering of multiple requests.

• For details, come to the CORBA 3 Tutorial
ORB RFP5, in 2 parts, standardized COM/CORBA Interworking during 1996-7
Strategy -- Leveraging Java

Java Orblet

Java Enabled WEB Browser

WEB Server

HTTP
-- HTML
-- GIF, JPEG
-- AV, WAV

CGI

Programs
IIOP (CORBA) TCL

IIOP

Sea of Objects (CORBA) IIOP

Object Management Group

Copyright © 2001 Object Management Group
Server-Side Scalability

- CORBA is great for huge, heavily loaded applications
  - and, with other specializations, for real-time, embedded, and fault-tolerant systems too
- CORBA Client-Side Model is Simple
- Scalability is implemented on the Server
- Several different Resource and Memory Allocation Models adjust for load
- CORBA Server Mechanisms:
  - Portable Object Adapter (POA, CORBA 2.3)
  - CORBA Component Model (CCM, CORBA 3)
CORBA 3.0 Adds --

• Improved Java and Internet Integration
  – Java-to-IDL (reverse) Mapping
  – Firewall Specification
  – CORBA Object URLs

• Quality of Service Control
  – Asynchronous Invocation/Messaging
  – Invocation QoS Control
  – Realtime, Minimum, Fault Tolerant CORBA

• CORBA Component Model
  – Objects Pass-by-Value
  – Component container
    • Transactional, Persistent, Secure
  – Distribution Format
  – Scripting Language Specification
CORBA Component Model (CCM)

- CORBA is great for building Enterprise and Internet applications
- But, of the thousands of CORBA usage patterns, a few stand out
- CCM packages up these successful patterns, including
  - POA servant management
  - Transactions and Persistence
  - Security
  - Event Handling
  - Configuration
  - Interface Connection and Assembly
- This speeds and simplifies application building, and ensures success
EJB Integration

• An EJB can look like a CCM Component to another CCM Component
• A CCM Component can look like an EJB to another EJB
• This Allows an Application to use a Combination of EJBs and Components
Section 5: What’s an ORB

- Components of an ORB:
  - Client Side
  - Server Side
- Client Stubs and Server Skeletons
- The BOA and the POA
- DII, DSI, and Interface Repository
All objects are defined in IDL by specifying their interfaces.
Object definitions (interfaces) are manifest as objects in the Interface Repository, as client stubs, and as implementation skeletons.
Descriptions of object implementations are maintained as objects in the Implementation Repository.
ORB Components

Client

- Dynamic Invocation
- Client Stubs
- ORB Interface

Object Implementation

- Impl Skeletons
- DSI
- POA

ORB Core

One interface
One interface per object adaptor
One interface per object operation

DSI: Dynamic Skeleton Interface
POA: Portable Object Adapter

Proprietary interface
Normal call interface
Up call interface
Client Side

Clients perform requests using object references.

Clients may issue requests through object interface stubs (static) or dynamic invocation interface.

Clients may access general ORB services:
- Interface Repository.
- Context Management.
- List Management.
- Request Management.
Implementation Side

Implementations receive requests through skeletons (without knowledge of client-side invocation approach).

The Portable Object Adapter supports a wide range of implementation schemes in a portable way.

The POA supports both the static and dynamic skeleton interfaces.
What’s a **Servant**? and Why?

- Sometimes the simple One-Object-Reference for One-Running-Implementation model isn’t enough:

- For example -
  - When we’re getting millions of hits per hour and we need multiple implementations for a single object reference, or
  - When we have 4 million purchase orders but a single running implementation, or just a few, could serve them all.
Enter the Servant

- We can change the model on the server side without breaking anything on the client side.
- A Servant is a running implementation which provides the functionality for one or more or less Object References.
- Policies specify how Servants map to Object References.
- Object Ids (OIDs) identify Servants to the POA.
- The POA includes components to activate, deactivate, and manage Servants.
The POA Supports -

- Object Implementation Portability between ORBs
- Object Instances with Persistent Identities
- Transparent Activation
- Single Servant Implementing Multiple Instances
- Transient Objects with minimal programming
- Fine or Coarse Control of Behavior and Persistence by an Implementation
- Multiple Policies for Key Object Behaviors
- Implementations Inheriting from Static Skeleton Classes
Section 6: Introduction to the OMA

- The CORBA services, CORBA facilities, and CORBA domains
- Using Standard Services in design, implementation, and at runtime
- Key Basics: Naming/Trader; Event Services
- Business necessities: Transaction and Security Services
OMA Overview

Application Objects

Vertical CORBA Facilities

Horizontal CORBA Facilities

Object Request Broker

CORBA Services
OMA Overview

Application Objects
Vertical CORBA Facilities
Horizontal CORBA Facilities

Object Request Broker

CORBA Services

Lifecycle
Events
Naming
Persistence
Transactions
Concurrency

Externalization
Security
Time
Properties
Query
Licensing
Naming, Events/Notification

Two Keys to Coordinating your Distributed CORBA Application

– use Naming Service to pass Object References around your network
– use Events and Notification Services to inform clients and other objects of a change in state or other significant event
Transactions, Security

Two Keys to Running your Business on CORBA -

- Transaction Service: Executing with the reliability and speed of electronic business
- Security: Interoperate with your customers and suppliers while keeping your private business private
OMA Overview

Not standardized by OMG; Scope is Single application or vendor
OMA Overview

Not standardized by OMG; Scope is Single application or vendor

Business Objects
Healthcare
Finance
Telecommunication

Internationalization
Secure Time
Mobile Agents
Printing Facility

Application Objects

Vertical CORBA Facilities

Horizontal CORBA Facilities

Object Request Broker

Lifecyle
Events
Naming
Persistence
Transactions
Concurrency

Externalization
Security
Time
Properties
Query
Licensing

CORBA Services
Vertical CORBAfacilities

- Official OMG specifications in vertical market domains:
  - Business Objects
  - Finance/Insurance
  - Electronic Commerce
  - Healthcare
  - Telecommunications
  - Transportation
  - Manufacturing
  - Life Sciences Research
  - Utilities
  - Space
  - Coming: Statistics, Call Center

- Bring benefits of CORBA and OMA to Domains.
More Domain DSIGs

• Distributed Simulation DSIG
  – Supported by US DMSO
  – But not just military simulations

• C4I DSIG
  – Command, Control, Communications, Computers, and Information
  – Military orientation

• GIS DSIG
  – Geographic Information Systems
  – Liaison with OGC

• ADSS DSIG
  – Autonomous Decentralized Systems
  – Applications to Dist Sim, Mfg, other areas
Domain Liaisons/Support

- Telecomms: TINA-C, DAVIC, NMF, ISO SC21, SC29, ITU-T
- Finance, Insurance: FSTC, ACORD, SIMC
- Workflow: WfMC
- Electronic Commerce: CommerceNet, EPF
- Pharmacy: NCPDP, NACDS
- Healthcare: HISB, HL7, DICOM, MRI
- Transportation: FAA, European ATCs
- Geographic IS: OGC
- Oil E&P: POSC
Domains in the OMA

CCM, future BOF, CBOs, & Framework

Horizontal CORBAfacilities

CORBAservices

Remember - Access to components is object-oriented; NOT hierarchical!
Business Object Progress

- Three Business Objects RFPs
- Issued in PA in April 99 by A&DTF
- To complete mid-2000
  - UML Profile for EDOC
  - UML Mapping to CORBA
  - Textual Notation for UML
- To follow:
  - CORBA Mapping for UML EDOC Profile
- a BOF Architecture all of OMG can support
BODTF Specifications

Business Objects Domain Task Force

• The first set of CBOs: Task, Session
  – Support Computer-based Cooperative Work

• Workflow Specification
  – Supported by Workflow Management Coalition and others
Telecom Specifications

Telecommunications Domain Task Force:

- Control and Management of A/V Streams
- Notification Service
- CORBA/TMN Interworking
- CORBA/IN Interworking
- Telecom Log Service Facility
Manufacturing Specifications

Manufacturing Domain Task Force:

- Product Data Management Enablers Spec
- Distributed Simulation HLA Specification
  - Mfg DTF working with Distributed Simulation SIG
  - A High Level Architecture for Distributed Simulation
  - Supported by US DMSO
Healthcare Specifications

Healthcare Domain Task Force

• Master Patient Identifier Specification
• Lexicon Query Service Specification
• Clinical Observations RFP
• Healthcare Resource Access Control
Finance Specification

Financial Domain Task Force:

• Currency Specification
• Party Management Facility Spec
• General Ledger Spec
Transportation Specification

Transportation Domain Task Force

- Air Traffic Control Display Manager Interface Specification
  - Support from Air Traffic Control agencies in Europe and the US
  - This technology is running in prototype in Germany now
E-Commerce Specification

Electronic Commerce Domain Task Force:

- Negotiation Framework Specification
  - Part of the E-Commerce Architecture
  - Wide support among OMG members
Life Science Research Spec

Life Sciences Research Domain Task Force

- OMG’s newest Task Force

• Biomolecular Sequence Analysis RFP
  - DNA, RNA, Proteins are all Sequences
  - Interfaces for Representation, Manipulation, Analysis of sequences
    • Does *not* standardize analytical methods!
CORBA/OMA Environment

• Starts with the Basics:
  – IDL Interfaces & Mappings
  – ORB-based Architecture
  – Static & Dynamic Invocation Modes
  – GIOP/IIOP Interoperability
  – Optional Asynch Modes
  – Naming Service
  – Event Service

• Add Services and Facilities:
  – Security Service
  – Transaction Service
  – Object Trader Service
  – COM/CORBA Mapping

• Then add Domain Components...
Section 7: Overview of OMG

• OMG Organization

• Who belongs to OMG?

• Creating a new OMG specification
OMG: Background

- About 800 member companies, world’s largest software consortium.
- Founded April 1989 - Twelve Years Old
- Small staff (27 full time); no internal development. Offices in U.S.A., Germany, Japan, U.K, Australia, India.
- Dedicated to creating and popularizing object-oriented standards for application integration based on existing technology.
Worldwide Scope

3M
Alcatel
ACORD
AOL/T-W
BEA Systems
BellSouth
Boeing Aircraft
Borland
Bristol-Myers
Compaq
Computer Assocs
Concept 5
Daimler-Benz AG
Deere & Co.
DMSO
EDS
Ericsson
Ford Motor
Fujitsu
GCHQ
GMD Fokus
HP
Hitachi
IBM
IONA
Lucent
MedContras
Metaphase
Mitre
Motorola
NEC
Nippon T&T
NIST
Nokia
Nortel
NSA
PrismTech
Rational SW
Raytheon
SAP AG
Shell Services
Siemens
Sprint
Sun
Telelogic
TIBCO
Unisys
Vertel
W3C
Workflow Mgmt

7/18/01 Copyright © 2001 Object Management Group 114
Meetings, Meetings!

- OMG Specifications are adopted at our meetings.
- Held Five times a year, at member companies’ sites around the world.
- Lasts a week and attracts over 500 people.
- Every subgroup meets; up to 30 simultaneous sessions on some days.
- You’re invited to come as an observer! Just let me know (email: info@omg.org).
Adoption Process

- RFI (Request for Information) to establish range of commercially available software.
- RFP (Request for Proposals) to gather explicit descriptions of available software.
- Letters of Intent to establish corporate direction.
- Task Force and End User evaluation & recommendation; simultaneous Business Committee examination.
- Board decision based on TC, End User, and BC recommendations.
Availability

Innovative approach for selection of standard interfaces to adopt:

1. OMG adopts & publishes interface specifications.

2. Interface Implementations must be available commercially from OMG Corporate member.

3. Interface Specifications are freely available to members and non-members alike.

4. Interface Specifications chosen from existing products in competitive selection process.
Review: CORBA Features

- **Transparencies:**
  - Platform/vendor
  - Operating System
  - Network HW/SW
  - Location
  - Programming language

- **Dynamic binding** and **typing**

- **Object Orientation**
  - Encapsulation
  - Polymorphism
  - Inheritance
  - Instantiation

- **Extended services**
  - Naming/trader
  - Events/notification
  - Security, domains

in an Open Specification with multivendor support
Contact OMG:

• Web:
  – Home page: http://www.omg.org
• Email: siegel@omg.org or info@omg.org
• Telephone: 781-444-0404