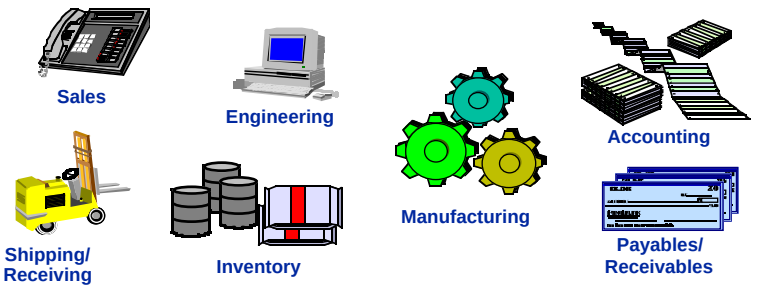
OBJECT MANAGEMENT GROUP

Model Driven Architecture: coping with multiple platforms

Andrew Watson
VP & Technical Director, OMG

Business application integration



Major problem for business & OMG's main focus during 1990s

Roots of the problem

- Approx 1 computer/business in 1960s, hundreds by 1990s
- Multiple programming languages
  - 3 million programmers write COBOL for a living
  - c.f. 1.6 million use Visual Basic, 1.1 million C and C++
- Multiple operating systems
  - Unix, MVS, MacOS, NT, Windows, Windows CE, PalmOS ...
  - Then there's your pager, cell phone, set-top box ...
- Multiple networks
  - Ethernet, ATM, IP, SS7, Appletalk, USB, Firewire ...

Solving the integration problem

- 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 programming languages
- There must be consensus on interfaces and interoperability

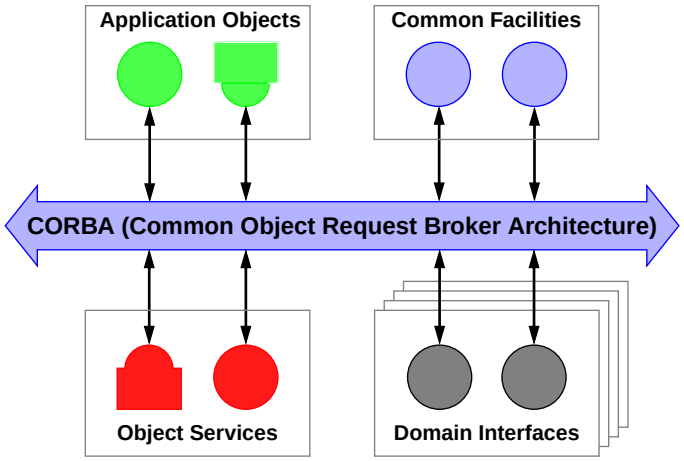
OMG's mission

- Develop a architecture, using object technology, for distributed application integration, guaranteeing:
  - ✓ reusability of components
  - ✓ interoperability & portability
  - ✓ basis in commercially available software
- Specifications freely available to all (OMG members or not)
  - Visit http://www.omg.org
  - Implementation available from at least one OMG member
- Member-controlled and not-for-profit
  - Technical direction determined by Technology Committees

Main platform specifications

- Unified Modelling Language (UML)
  - Uses XML Metadata Interchange (XMI) to exchange designs
- Common Object Request Broker Architecture (CORBA)
  - Platform-neutral Middleware for application integration
- Common Warehouse Metamodel (CWM)
  - Data warehousing
- Meta-Object Facility (MOF)
  - Meta-data repository standard

OMA Reference Model



(Some) CORBA implementations

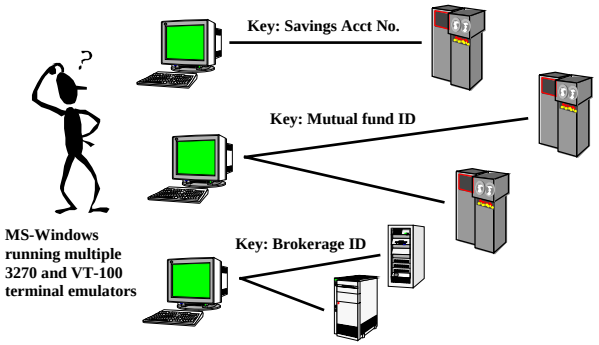
AT&T Labs OmniORB 2	Oracle 8i
Peerlogic LiveContent Broker	Netscape Navigator 4.0
IBM Component Broker	Lotus Notes 5.0
Inprise Visibroker	Novell Netware
BEA Weblogic Enterprise	Sun Java/IDL (in JDK 2)
Deutsche Telecom MICO	Washington U. TAO
Gerald Brose JacORB	Paragon Software Oak
Iona Orbix	Lockheed-Martin HardPack
Red Hat ORBit	Fujitsu ObjectDirector
Hitachi TPBroker	NEC ObjectSpinner
Harvard Arachne	Gemstone GemORB
OIS ORBExpress	DNS SmalltalkBroker

Wells Fargo example: Background

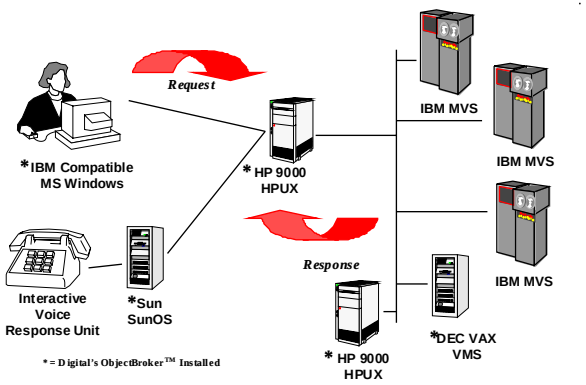
- During 1980s Wells Fargo expanded its range of financial services
  - Introduced a demand for compound statement banking
  - Single unified statement of all accounts and transactions
- But existing systems were account-focussed
  - Hard to associate all accounts with one owner
- Multiple systems for different businesses
  - Deposits -> IBM mainframes
  - Mutual funds -> VAX/VMS
  - Brokerage -> Tandem

Changing requirements

- By early 1990s bank wanted to offer telephone banking
  - But this would require 3 virtual terminals per agent ...



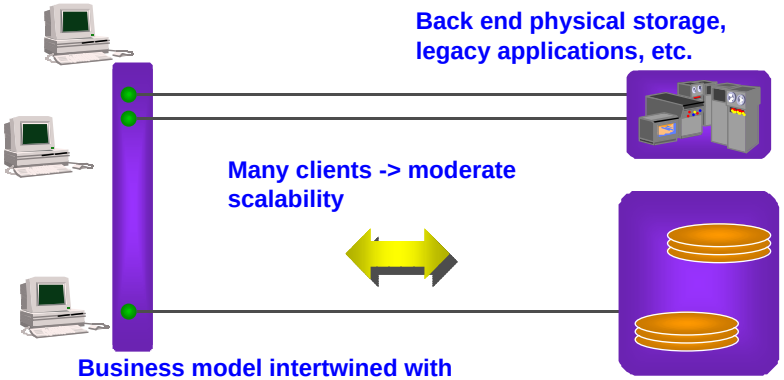
Integration using third tier



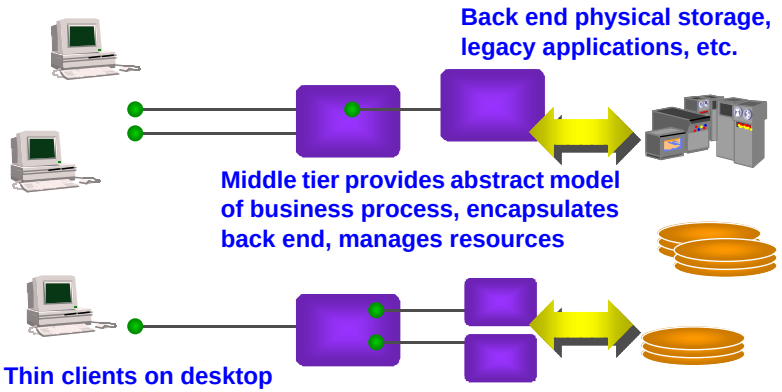
Further enhancements

- An Interactive Voice Response Unit
  - Running on a SunOS box
- Automated Teller Machine access to brokerage information
  - Via mainframe-based Prodigy service
- Access to real-time stock-market data
  - Via purchased external network and Quotron application, running on AIX on IBM RS/6000
- Access via the Web
  - Wells Fargo the first Web-accessible bank?

Enterprise Systems 80s style - 2 tiers



Enterprise Systems 90s style - N tiers



Creating multi-tier applications

- Component Models extend basic middleware to support common patterns of 3 tier development
  - EJB does Java components over CORBA
  - CCM does multi-language components over CORBA
- Manage sharing critical resources in application server
  - Processes, threads, memory, database connections, etc.
  - Replace previous hand-coded layer over middleware
- Provides access to enterprise services with minimal coding
  - Security, transactions, naming etc.
- Support re-use of interchangeable application components (?)

Enterprise Java Beans based on CORBA

- **Invocation** — RMI over IIOP between J2EE instances
- **Security** — OMG GIOP 1.2 & OMG Common Secure Interop V2
- **Naming** — JNDI uses OMG Interoperable Naming/IIOP
- **Transactions** — OMG OTS over IIOP recommended, and will be required in J2EE 1.4
- **Persistence** — EJB 2.x is same as CCM/PSS

Example middleware domain: Telecoms

- In 1998 AT&T working on approx. 30 systems using CORBA
  - FASTAR is “the application that keeps AT&T off the front page of the Wall Street Journal”
- Telefónica has about a dozen deployed CORBA apps
  - Largest services 1,600 operators, handles 1m calls daily
- Swiss Telecom handles subscriber Admin from 17 regional offices with CORBA app with Windows/COM front ends

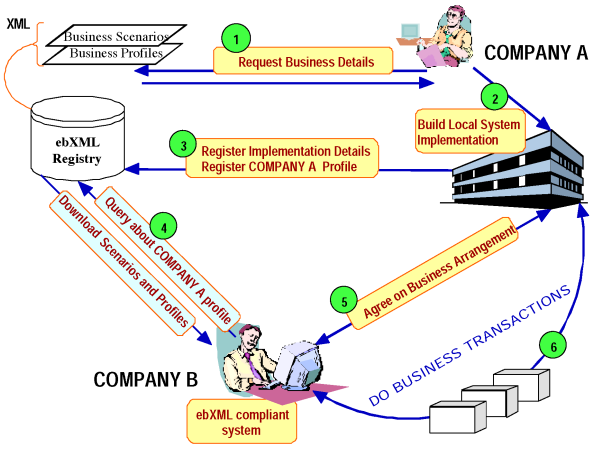
OMG’s Domains

- Task Forces adopt Domain technology:
  - ✓ Business Objects
  - ✓ Electronic Commerce
  - ✓ Finance
  - ✓ Transportation
  - ✓ Utilities
  - ✓ Manufacturing
  - ✓ Telecoms
  - ✓ Healthcare
  - ✓ Lifesciences
  - ✓ Space & satellite
- Special Interest Groups share information:
  - ✓ C4I
  - ✓ Testing
  - ✓ Benchmarking
  - ✓ Human Resources
  - ✓ Simulation
  - ✓ EAI (etc., etc. ...)

Where next?

- “Middleware” entered the IT lexicon about ten years ago
  - One big computer per company –> several smaller ones
- Today we aim for similar integration between enterprises
  - Why should I type your computerised bill into my computer?
  - Conditions are right: most businesses computerised, public networks cheap and ubiquitous (compare EDI in the 80s)
- ebXML, Microsoft .NET, Sun ONE and other ‘Web services” initiatives aim to be “the” middleware to link enterprises
  - Probably no single one will completely succeed

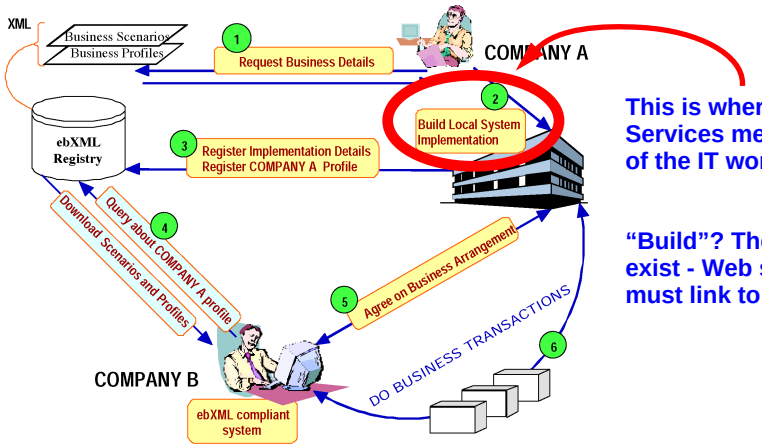
The view from the top (1)



Typical Web Services use-case

(borrowed from ebXML Technical Architecture Specification v1.0.4)

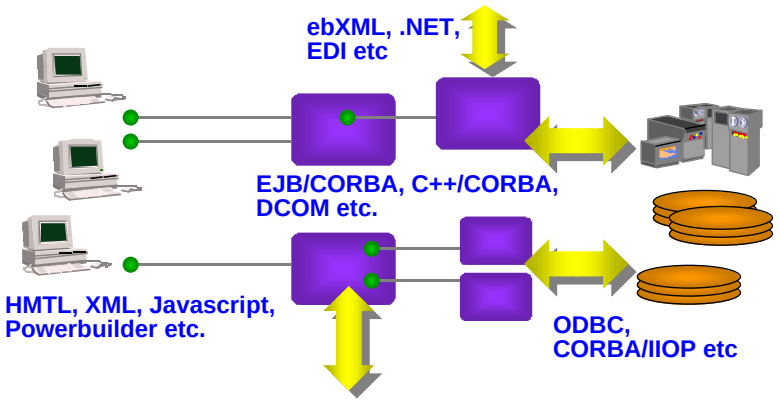
The view from the top (2)



This is where Web Services meet the rest of the IT world

"Build"? They already exist - Web services must link to them.

The view from the trenches



Integrating multiple middlewares

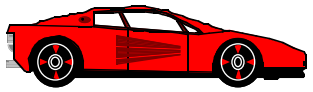
- Web services and other, established middleware must coexist
- OMG has designed Model-Driven Architecture (MDA) to help
- Maintain common design base in a neutral, widely-used format
  - Unified Modelling Language (UML)
- Derive platform-specific UML models for platforms in use
- Use tools to transform platform-specific models into interface descriptions for all cooperating platforms
  - Where necessary, transform design data for existing sub-systems back into UML PSMs

Unified Modelling Language

- The successor to multiplicity of OO A&D notations of early 90s
- Result of OMG RFP begun in 1994, completed in late 1997
  - Complemented by XML Metadata Interchange (XMI) spec
- Standard released flood of support
  - 60+ books
  - A dozen tools
  - Training widely available
  - Used in 70% of IS shops?
- UML 2 design now underway

UNIFIED MODELING LANGUAGE™

Simple example: creating XML DTDs

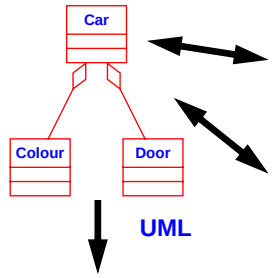


<Car>
 <doors> 2</ doors>
 <colour> red</ colour>
</Car>

<auto doors="2" colour="red"/>

?

Creating XML DTDs




```
<Class>
 <Name> Car</Name>
</Class>
```
Model in XMI

Model interchange

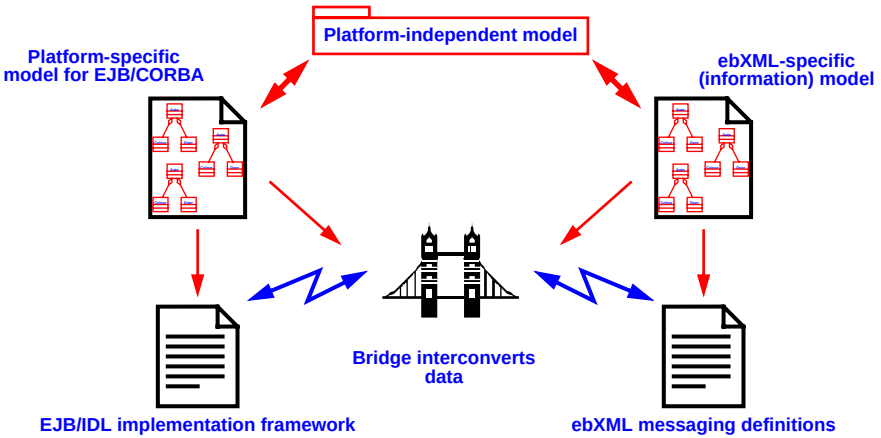
```
<element name="Car"/>

<!ELEMENT Car(Colour*, Door*)>
```
XMI Schema & DTD

```
Class Car
  { Colour colour
    Door   door
  }
```
Java, IDL

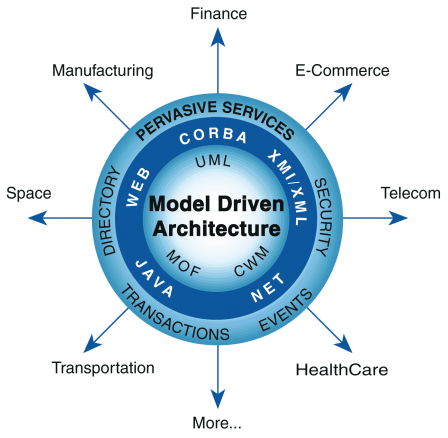
```
<Car>
 <Colour>Red</Colour>
 <Door>2</Door>
</Car>
```
XMI doc.

Instance interchange

Model-Driven Architecture

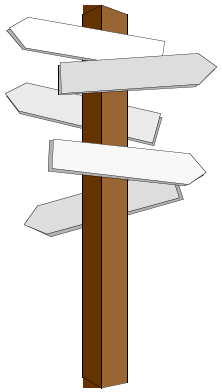


Model-Driven Architecture



MDA: Work in progress

- Some interesting industry exemplars:
  - Wells Fargo (again!)
  - GCHQ
- Multiple platform services (e.g. JTS/OTS)
- Specification tools
  - UML profile for EDOC (for PIMs)
  - XMI maps UML –> XML DTDs
  - UML profile for CORBA
  - UML for Java
  - UML for .NET ... ??

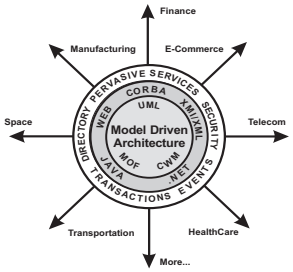


Summary

- In the 90s, single middleware standards integrated a confusing mixture of platforms inside the firewall
- In the 00s, we must integrate a confusing mixture middlewares across the firewall
  - Web services, WAP, the Web, standard and proprietary enterprise middleware
- As before, semi-rigorous, machine-readable component descriptions provide the foundation
  - From IDLs to UML

For further Information

- General: http://www.omg.org
- UML & XMI: http://www.omg.org/uml/
- MDA: http://www.omg.org/mda/
- Email: andrew@omg.org



Thank You