Application Servers
G22.3033-011

Session 12 - Main Theme
Web Services Platforms (Part I)

Dr. Jean-Claude Franchitti

New York University
Computer Science Department
Courant Institute of Mathematical Sciences

---

**Agenda**

- Continued from Session 11:
  - EAI and B2Bi Environments
- MDA Technology
  - UML-Based Component Modeling and Related Standards
  - XML Pattern Languages and XML Model-Based Applications
- Web Services
  - XML-RPC
  - SOAP
  - UDDI
  - WSDL
- Summary
- Readings
- Assignment #8 (due: 12/10)
Summary of Previous Session

- Channel Independence and Pervasive Devices
  - Connected Devices
- XML in Component-Based Development Environments
  - XML and JavaBeans/EJBs
  - Deployment Descriptors for OMA Services
  - More on the OMA Persistence Service
- EAI Environments
- B2Bi Environments (e.g., WebMethods Framework)
- Summary
- Readings
- Assignment #8 (will be handed out on 11/26, due: 12/10)

Application Servers Architectures

- Application Servers for Enhanced HTML (traditional)
  - a.k.a., Page-Based Application Servers
  - Mostly Used to Support Standalone Web Applications
- New Generation Page-Based Script-Oriented App. Servers
  - First Generation Extensions (e.g., Microsoft IIS with COM+/ASP)
  - Servlet/JSP Environments
  - XSP Environment
  - Can now be used as front-end to enterprise applications
  - Hybrid development environments
- Distributed Object Computing Platforms
  - Provide an infrastructure for distributed communications enabling
  - Still need to merge traditional web-oriented computing with object computing
- Object Management Architectures
  - DOC Platform + APIs to reusable services and facilities
- OMAs + Component Models -> J2EE, CCM, DNA
- MDAs with XML/Web Services/Channels/B2Bi-Enabling services
Part I
(continued from Session 11)

EAI and B2Bi
Enterprise Application Integration
and Business to Business Integration

Also See Session 11 Presentations:
“Enterprise Application Integration (EAI)”
“Practical Application of EAI”
“The STP and T+1 Application”

and the Session 11 Handout on:
“Enterprise Application Integration (EAI)”

Part II
(background information)

XML Modeling Capabilities
XML Physical and Logical Structure

- **Physical Structure**
  - Governs the content in a document in form of storage units
  - Storage units are referred to as entities
  - See [http://www.w3.org/TR/REC-xml#sec-physical-struct](http://www.w3.org/TR/REC-xml#sec-physical-struct)

- **Logical Structure**
  - What elements are to be included in a document
  - In what order should elements be included
  - See [http://www.w3.org/TR/REC-xml#sec-logical-struct](http://www.w3.org/TR/REC-xml#sec-logical-struct)

XML Physical Entities

- Allow to assign a name to some content, and use that name to refer to it
- Eight Possible Combinations:
  - Parsed vs. Unparsed
  - General vs. Parameter
  - Internal vs. External
- Five Actual Categories:
  - Internal parsed general
  - Internal parsed parameter
  - External parsed general
  - External parsed parameter
  - External unparsed general

---

7

---

8
Logical Structure: Namespaces

- See Namespaces 1.0
- Sample Element:
  
  <z:a z:b="x" c="y" xmlns:z="http://www.foo.com"/>

- Corresponding DTD Declaration
  
  <!ELEMENT z:a EMPTY>
  <!ATTLIST z:a
    z:b CDATA #IMPLIED
    c CDATA #IMPLIED
    xmlns:z CDATA #FIXED "http://www.foo.com">

Logical Structure: DTDs

- Shortcomings
  - Separate Syntax
    
    <!ELEMENT Para (#PCDATA)*>  
    <Para>Some paragraph</Para>
    vs.
    <ElementType name="Para">
    <ContentModel><PCData/></ContentModel>
    </ElementType>

  - Lack of Support for Data-typing
    - DTD Treats an XML Structure as a String of Characters
      
      <Price currency="USD">1450</Price>
      <Price currency="USD">too high</Price>
Logical Structure: XML Schemas

- Structures
  - How elements and attributes are setup in an XML document

- Datatypes
  - Built-in datatypes (e.g., String, Boolean, numbers)
  - Generated datatypes (e.g., dates, times, real values)
  - Support for user generated datatypes
  - Backward compatibility with functional subset (DTD)
    - ID, IDREF, NMTOKEN, and SGML-based types

- Grouping of Elements/Attributes
  - Archetypes and Attribute Groups

- Inheritance
  - Via Basetypes, and Archetypes/Attribute Groups

Logical Structure: XML Schemas (continued)

```xml
<datatype name='AgeInYears'>
  <basetype name='integer'
    URI="http://www.w3.org/xmlschemas/datatypes"/>
  <minInclusive>0</minInclusive>
  <maxInclusive>140</maxInclusive>
</datatype>

<attribute name="employeesAge"
  type="AgeInYears"/>
```
Logical Structure: Navigation

- URIs/URLs
  - Syntax for encapsulating a name in any registered namespace, and label it with the namespace
  - Produce a member of the universal set of reachable objects
  - See http://www.w3.org/Addressing/
- XPath
  - Used to locate certain parts of an XML document
  - See Session 3 handout on “Processing XML documents in Java using XPath and XSLT”

XML Linking/Pointer Language

- XLink
  - Allows elements to be inserted into XML documents in order to create and describe links between resources
  - See Article at http://www.simonstl.com/articles/xlink/
- XML Base
  - Equivalent of HTML BASE functionality generically in XML documents by defining an XML attribute named xml:base
- Xpointer
  - Language used as a fragment identifier for any URI-reference that locates a resource of Internet media type text/xml or application/xml
  - Based on XPath
  - See Tools at http://www.xmlsoftware.com/xlink/
XLink Example

- `<my:crossReference
xmlns:my="http://example.com/
xmlns:xlink="http://www.w3.org/1999/xlink"
xlink:type="simple"
xlink:href="students.xml"
xlink:role="studentlist"
xlink:title="Student List"
xlink:show="new"
xlink:actuate="onRequest">
Current List of Students
</my:crossReference>`

XPointer Example

- `<!DOCTYPE SPEECH[
<!ELEMENT SPEECH (#PCDATA|SPEAKER|DIRECTION)*>
<!ATTLIST SPEECH
ID ID #IMPLIED>
<!ELEMENT SPEAKER (#PCDATA)>
<!ELEMENT DIRECTION (#PCDATA)>]
<SPEECH ID="a27"><SPEAKER>Polonius</SPEAKER>
<DIRECTION>crossing downstage</DIRECTION>Fare you well,
my lord. <DIRECTION>To Ros.</DIRECTION>
You go to seek Lord Hamlet? There he is.</SPEECH>

- `id(a27).child(2,DIRECTION)
Selects the second "DIRECTION" element (whose content is "To Ros.")`

- `id(a27).child(2,#text)
Selects the second text region,"Fare you well, my lord." (The line break between the SPEAKER and DIRECTION elements is the first text region.)`
**XML Base Example**

- `<?xml version="1.0"?>
  <html xmlns="http://www.w3.org/TR/xhtml1/strict"
       xml:base="http://somewhere.org">
    <head>
      <title>Virtual Library</title>
    </head>
    <body>
      <p>See <a href="new.xml">what's new</a>!</p>
      <p>Check out the hot picks of the day!</p>
      <ol xml:base="/hotpicks">
        <li><a href="pick1.xml">Hot Pick #1</a></li>
        <li><a href="pick2.xml">Hot Pick #2</a></li>
        <li><a href="pick3.xml">Hot Pick #3</a></li>
      </ol>
    </body>
  </html>`

  - "what's new" resolves to the URI "http://somewhere.org/new.xml"
  - "Hot Pick #1" resolves to the URI "http://somewhere.org/hotpicks/pick1.xml"
  - "Hot Pick #2" resolves to the URI "http://somewhere.org/hotpicks/pick2.xml"
  - "Hot Pick #3" resolves to the URI "http://somewhere.org/hotpicks/pick3.xml"

---

**XML Data Binding**

- Aims to automatically generate substantial portions of the Java platform code that processes XML data
- A Sun project, codenamed “Adelard”
- See JSR-31 XML Data Binding Specification

  - see [http://java.sun.com/xml/jaxp-1.0.1/docs/binding/DataBinding.html](http://java.sun.com/xml/jaxp-1.0.1/docs/binding/DataBinding.html)
DTD and XML Schema Design

- **Extensibility**
  - Reuse existing DTDs or XML Schemas
  - DTDs and XML Schemas must be readable/maintainable

- **Compatibility**
  - DTDs/XML Schemas updates must be carefully crafted

- **Representing Non-Tree-Structured Data**
  - Designs must be crafted with efficiency in mind

---

Canonical XML

- W3C Recommendation (March 15, 2001)
- Canonical XML will be used to represent the result of parsing an XML document
- Canonical XML is necessary to establish the logical “equivalence” of XML documents
- Every well-formed XML document has a unique structurally equivalent canonical XML document
- When “Canonicalizing” an XML document, parsers preserves the minimum information needed by an XML application
Custom Markup Languages

- Mathematical Markup Language (MathML)
- OpenMath
- Chemical Markup Language (CML)
- Geography Markup Language (GML)
- Wireless Markup Language (WML)
- Synchronized Multimedia Integration Language (SMIL)
- Synchronized Vector Graphics (SVG)
- Extensible 3D (X3D)
- XML-Based User Interface Language (XUL)
- Extensible Log Format (XLF)

Industry Specific Markup Languages

(see http://www.oasis-open.org/cover/xml.html#contentsApps)

- Ontology Interchange Language (OIL)
- OMG Common Warehouse MetaData Interchange (CWMI)
- OMG Model Driven Architecture (MDA)
- Open Financial Exchange (OFX)
- Straight Through Processing Markup Language (STPML)
- Electronic Commerce Modeling Language (ECML)
- OASIS Business Transactions Technical Committee (OASISBT)
- BizTalk Framework (BizTalk)
- Commerce XML (cXML)
- RosettaNet (RosettaNet)
- Business Process Modeling Language (BPML)

Also See Session 12 Handout on:
“Secure Message Exchange on the Internet”

XML and Transport Protocols

- Asynchronous Transport Protocols Alternatives
  - Simple Mail Transfer Protocol (SMTP, RFC-821)
    - Simple but no guaranteed delivery and acknowledgment
  - MQSeries
    - Reliable but proprietary and high cost
  - BEEP
- Synchronous Transport Protocols Alternatives
  - Sockets (TCP)
    - Flexible but requires low-level programming
  - HTTP 1.1 (RFC 2068), HTTPR
    - Widely used, supports firewall connections, and security (SSL)
    - Inefficient and session-less (connection- and state-less)
  - CORBA/IIOP
    - Efficient and cross-language/platform compatible but complex
  - SOAP-RP
XML Data Exchange Protocols

- Message Formats Alternatives for Data Exchange
  - Text-based (e.g., EDI, RFC822, SGML, XML)
  - Binary (e.g., ASN.1, CORBA/IIOP)

- Common XML/EDI Interfaces
  - An API that provide a common interface to work with EDI or XML/EDI objects is supported by OpenBusinessObjects

- Guidelines for using XML for EDI
  - [http://www.geocities.com/WallStreet/Floor/5815/guide.htm](http://www.geocities.com/WallStreet/Floor/5815/guide.htm)
  - [http://www.xmledi-group.org/](http://www.xmledi-group.org/)

- Electronic Business XML: ebXML

DOC Application-Level Protocols

- CORBA 3
  - IDL over IIOP v.s. XML over HTTP (XIOP)
    - XML Protocol ([XMLP](https://www.xmlp.net/)): XML-Based Messaging Systems
      - Standardized application to application XML messaging (via HTTP, and MQSeries)

- J2EE
  - JMS v.s. XML-based messaging
    - JMS
      - JMS API’s `TextMessage`
      - Custom JMS extensions
        - BEA's WebLogic XMLMessage subclass
    - XML
      - Sun’s Java API for XML Messaging (JAXM)
XML-Based e-Services Protocols

- XML-RPC and Peer-to-Peer Computing
  - http://xml.coverpages.org/xml-rpc.html
- Simple Object Application Protocol (SOAP)
  - http://soap.develop.com/xmlrpc/
- Universal Description, Discovery, and Integration (UDDI)
- Web Service Definition Language (WSDL)
  - http://www.w3.org/TR/wsd1

---

XML Fragment Interchange

- Defines a way to send fragments of an XML document without having to send all of the containing document up to the fragment
- Fragments are not limited to predetermined entities
- The approach captures the context that the fragment had in the larger document to make it available to the recipient
- See http://www.w3.org/TR/WD-xml-fragment
Organizing Message Data

- Single XML Messages
  - Include everything including binary data into a single XML message
  - Easier to maintain consistency and integrity of constituent parts
  - Popular method is to use Base64 encoding to embed binary data in XML documents
- Separate MetaContent and Resources
  - Send everything at once or let recipient fetch resources as needed

Message Validation and Encodings

- Validation must be performed on messages to accept a fixed set of DTDs or XML Schemas
- Messages must use encodings that are supported by conforming XML processors
  - UTF-8, and UTF-16 provide space efficient alternatives
**XML and Security Standards**

- **Security Alternatives**
  - IPS Security (IPSec, RFC-2401-2412)
    - IP level security, widely used for Virtual Private Networks (VPNs)
  - Secure Sockets Layer (SSL)
    - HTTP connection security
    - Transport Layer Security (TLS) standard
  - S/MIME (PKCS#7)
    - Secure/Multipurpose Internet Mail Exchange
    - Secure mail standard
    - Uses X.509 certificates for authentication
  - Pretty Good Privacy (PGP)
    - De facto standard for e-mail security
    - Can be used to send encrypted digital signatures, and encrypt stored files

**SSL Secure Messaging**

- **Confidentiality via symmetric cryptosystem**
  - Data Encryption Standard (DES, 56-bit key)
  - Ron Rivest Code (RC-4, 40-128 bits variable key length)

- **Integrity via Message Authentication Code (MAC)**
  - MAC based on secure hash function such as MD5 (Message Digest 5) and SHA 1 (Secure Hash Algorithm 1)

- **Authentication**
  - Optional for client (via HTTP/SSL or SSL Certificate-Based), but mandatory for server
  - Server’s digital certificate issued by Certificate Authority (CA)
    - Verisign, Inc’s certificates
    - SSL uses X.509 certificate format (ITU-T standard)
  - Non-Repudiatability (via digital signatures, e.g. PKCS#7)
    - DOMHash/Java Cryptography Architecture (JCA)
XML Signatures

- Joint W3C and IETF Working Group
- W3C Candidate Recommendation (April 19, 2001)
- XML compliant syntax used for representing the signature of Web resources and portions of protocol messages
- Procedures for computing and verifying such signatures (i.e., integrity and authentication)
- Does not address encryption, and authorization

Part IV

**MDA Technology**

*Also See [http://www.omg.org/mda](http://www.omg.org/mda), and associated Webcast and Session 12 handout on “Application Modeling Using XML”*
Model Driven Architectures (MDA)

- MDA Technology Relies on:
  - UML, MOF, and CWM
- Applications Based on MDA are Platform Independent
- Implementations/Realizations can be Targeted to Any Application Server Platform
  - XML-Based Web-Enabled/Web Services-Enabled Platforms Based on CORBA, J2EE, Microsoft

UML’s Business Engineering Methodology

- Business Model/Architecture
  - Use Case View/Model
- Application Model/Architecture
  - Logical and Process View/Models
    - Content, Data, and Process Model (e.g., OIM’s knowledge management, and database/data warehousing models)
- Application Infrastructure Model/Architecture
  - Implementation View
    - Component Model (e.g., OIM’s component and object model)
- Technology Model/Architecture
  - Deployment View/Model
UML and Modeling Methodologies

- UML: object modeling
- XML: content modeling
- ORM: data modeling

Architecture Development Map

As information is collected, work effort, estimates and solution becomes concrete

Domain Independent
Domain Dependent
Infrastructure
Application

Concrete
Abstract

Proposal End Point
Customer Information Gathering

Domain Specific Information

Technology & Software

Concrete Costs
Application Costs
Application Costs
Application Costs

Application Development
Technical Solution Development

Abstract Costs
Application Costs
Application Costs

Domain Analysis

Customer & Business Models

Domain Specific Information

Client Preferences

Initial Costs
Abstract Costs
Concrete Costs

Application Production (Assembly/Design)
Technical Solution Development
Domain Analysis

Information Not Specific to Domain

Questions?

As information is collected, work effort, estimates and solution becomes concrete.
Towards XML Model Based Computing
(review)

- Step 1: Document Object Model
- Step 2: XML Data Binding
- Step 3: Standard XML Information Models
- Step 4: XML Application Services Frameworks
  - Processing, Rendering, Querying, Secure Messaging
- Step 5: XML-Based “Web Object Model”
  - Web Services Architecture
- Step 6: XML Model Driven Architectures (to come)

Current XML-Based Software Development

- Business Engineering Methodology
  - Language + Process + Tools
  - e.g., Rational Unified Process (RUP)
- XML Application Development Infrastructure
  - Metadata Management (e.g., XMI)
  - XML APIs (e.g., JAXP, JAXB)
  - XML Tools (e.g., XML Editors, XML Parsers)
- XML Applications:
  - Application(s) of XML
  - XML-based applications/services
    - MOM & POP
    - Other Services
  - Application Infrastructure Frameworks
XML Metadata Management

- Issue: UML may not provide enough modeling views and enough expressive power in each view to represent a complete application
- Possible Solutions:
  - Extend UML
    - See OIM’s Analysis and Design Model
  - Use Different Modeling Languages:
    - See handout on “XML Information Modeling” (uses different models such as UML, XML, and ORM)
  - Use a Meta-Model: MOF and XMI
    - See handouts on “UML, MOF, and XMI” and “OMG’s XML Metadata Interchange Format (XMI)”

Open Information Model

- Analysis and Design Model
  - Unified Modeling Language (UML) - uml.dtd
  - UML Extensions - umlx.dtd
  - Common Data Types - dtm.dtd
  - Generic Elements - gen.dtd
- Components and Object Model
  - Component Description Model - cde.dtd
- Database and Warehousing Model
  - Database Schema Elements - dbm.dtd
  - Data Transformation Elements - tfm.dtd
  - OLAP Schema Elements - olp.dtd
  - Record Oriented Legacy Databases - rec.dtd
- Knowledge Management Model
  - Semantic Definition Elements - sim.dtd
XML Support for UML Modeling

- Meta Object Facility (MOF)
  - CORBA Common Facility for the Management of Meta Information such as UML Models, Database Schemas, Programming Language Types, etc.

- XML Metadata Interchange (XMI)
  - Enables interchange of metadata between modeling tools
  - Develop models using Rational Rose or Java, and the XMI toolkit
  - Use Objects by Design xmi-to-html.xsl style sheet and Cocoon framework to present the resulting model

MDA Development Approach
Sample MOF MetaModel

Sample MOF MetaModel
(continued)
Model Driven Development Tool
(http://www.sygel.com/)

XML Information Modeling

- Steps
  - Documenting the Information Structure
  - Representing the Information Structure in XML Form
  - Defining XML DTDs and/or Schemas

- Modeling Techniques
  - UML: object modeling
  - XML: content modeling
  - ORM: data modeling
  - See Session 3 handout on “XML Information Modeling”

- UML, MOF and XMI
  - See handout
Using UML to Model MLs

+request_supplier
+reply_recipient

<<communicate>>
glossary_request

Translator

<<communicate>>
+reply_supplier
+request_recipient

<<communicate>>
glossary_reply

Query Glossaries

Using UML to Model MLs

(continued)

Glossaries

<<XSD complexType>>
Glossary

<<XSDcomplexType>>
Entry

<<XSDcomplexType>>
DialectInfo

<<XSD complexType>>
SignedEntry

<<XSD simpleType>>
DialectCode

Properties

string

(from XSD Datatypes)

string

(from XSD Datatypes)

string
XML Metadata Management  
(continued)  

- Practical Use of XMI:  
  - Develop a model using Rational Rose or Java, and the XMI toolkit  
  - Use Objects by Design xmi-to-html.xsl style sheet and Cocoon framework to present the resulting model  

- Current State of Tools:  
  - Forward-Engineering of UML into XML Schemas is possible  
    - Can export .mdl files from Rational Rose, and convert them into XMI using SoftModeler (www.softera.com)  
    - SoftModeler can convert UML into XML W3C Schemas  
  - Reverse-Engineering of XML Schemas into UML is more difficult  
    - Not supported by current tools  

Part V  

Web Services Technology  

Also See: http://ejbinfo.com/articles/00/10/28/0933234.shtml and http://www.clipcode.com/peer/beep_protocol_guide.htm
Towards Web Services ...

- The Web Object Model
  - Adding “methods” to XML (e.g., WIDL)
  - SOAP
- XML and UML based component modeling
  - Model-based applications via UML to EJB component binding
  - XML Metadata Interchange (XMI) and related technology
- XML and Model-based computing
  - Model-based computing frameworks

Towards a “Web Object Model”

- See dated article by Frank Manola (OBJS)
- Should compare Manola’s proposed approach for building a web object model to the approach suggested in today’s Web Services architecture
- Web Services use XML as a portable data language to interchange information
  - The binding to Java and/or other application languages is implemented via SOAP endpoint connectors (e.g., SOAP to JMS)
**SOAP**

- Lightweight protocol for information exchange
  - Envelope
    - Framework that describes a message and its processing
  - Set of encoding rules
    - Used to express instances of application datatypes
  - Convention
    - RPC calls and responses
- Apache distributes an implementation of SOAP referred to as “AXIS”

---

**Content, Discovery, Universal access, and Intelligent Software Agents**

- UDDI: Universal Description, Discovery, and Integration
  - Industry-wide effort to bring a common standard for business-to-business (B2B) integration
  - Set of standard interfaces for accessing a database of web services
  - jUDDI (pronounced "Judy") is an open source Java-based implementation of a UDDI registry
  - Also see
    - [http://uddi.microsoft.com/default.aspx](http://uddi.microsoft.com/default.aspx)
    - [http://www.oasis-open.org/cover/uddi.html](http://www.oasis-open.org/cover/uddi.html)
    - [http://www.itpapers.com/cgi/SubcatIT.pl?scid=436](http://www.itpapers.com/cgi/SubcatIT.pl?scid=436)
- Intelligent Software Agents: ATLAS, Aglets, etc.
XML-Based e-Services Protocols and Architectures

- XML-RPC and Peer-to-Peer Computing
  - http://xml.coverpages.org/xml-rpc.html
- Simple Object Application Protocol (SOAP)
  - http://soap.develop.com/xmlrpc/
- Universal Description, Discovery, and Integration (UDDI)
- Web Service Definition Language (WSDL)
  - http://www.w3.org/TR/wsdl
- Pervasive devices
- Resource Description Framework (RDF)
  - Platform for Internet Content Selection (PICS)
  - Platform for Privacy Preferences (P3P)
  - Channel Definition Format (CDF)
  - Rich Site Summary (RSS)
  - Blocks Extensible Exchange Protocol (BXXP)

XML-Based e-Services Protocols and Architectures (continued)

- XML Protocol (XMLP): XML-Based Messaging Systems
  - Standardized application to application XML messaging (via HTTP, and MQSeries)
- XML and User Identification/Security
- XML and Databases
  - XML and JDBC
  - XML Extensions and Tools for Oracle, Informix, IBM DB2, and Microsoft SQL Server
- Transaction Authority Markup Language (XAML)
  - Coordinated processing of transaction-supporting web services between internal fulfillment services and external services
XML-Based e-Services Protocols and Architectures (continued)

- Sun’s Open Net Environment (ONE)
- HP’s NetAction/e-speak platform
- Oracle’s Dynamic Services platform
- Microsoft .NET platform
- IBM WebSphere Architecture (WSA) platform

Microsoft .NET Platform
Smart Devices + Windows XP + Web Services

- http://www.microsoft.com/net
- First set of Microsoft Web Services
Part VI

Conclusion

Summary

- XML complements EAI/B2Bi technology, but future EAI/B2Bi frameworks still need to address XML limitations
- XML complements application server technology by conveying structured data in a portable way
- XML provides modeling capabilities and secure message exchange over the Internet
- MDA technology promotes “XML Model-Based Computing”
- Web services platforms provide a shift from traditional enterprise distributed object computing technology to P2P computing with support for pervasive devices, discovery and intelligent agent capabilities
Readings

- Readings
  - Building Application Servers: Part II, and III
  - Handouts posted on the course web site
  - Explore EAI/B2Bi, MDA, and Web Services Environments
  - Read related white papers/documentation on EAI/B2Bi, MDA, and Web Services environments

Project Frameworks

- Project Frameworks Setup (ongoing)
  - Apache Web Server (version 1.3.20, www.apache.org)
  - Perl (version 5.x, www.perl.com), PHP
  - Microsoft IIS with COM+ and ASP (), ChiliSoft
  - Apache Tomcat
  - Macromedia JRun
  - Apache Cocoon 2/XSP
  - Visibroker, Orbacus
  - RMI-IIOP
  - Windows 95/98/NT/2000 or Software AG’s EntireX
  - WebLogic, WebSphere, JBoss, Enhydra, OpenEJB
  - Inprise AppServer, iPlanet.com iPlanet, Sybase EAServer, Oracle 9i, IONA iPortal, Xoology Concerto, Aligo M-1, Advanced Network Systems WebIX
  - GOAL Group OpenCCM, ExoLab.org OpenCCM, iCMG K2-CCM (C++), MICO/E (Eiffel ORB), JavaCCM, TAO Group, IONA iPortal (no CCM), Borland AppServer (no CCM), Sourceforge ML3 (“Mission Impossible 3”) and CIF projects
  - Microsoft DNA
  - Apache’s XercesJ, XalanJ, XMLSpy, Antenna House XML Formatter, Apache’s FOP, X-smiles
  - JWS, XMI Toolkit, IBM’s Web Service Toolkit
  - POSE, KVM for J2ME, NanoXML
  - IBM Alphaworks WSTK/.Net, XMI Toolkit, EJBMaker, WebMethods, etc.
Assignment

- Explore the references to EAI/B2Bi, MDA, and Web Services-Enabled Application Server technology
- Homework #6 due date is 11/19/01
- Homework #7: due date is 11/26/01
- Homework #8: due date is 12/10/01

Next Session:
Web Services Platforms (Part II)

- C# and .Net
- Mainstream UDDI Registries and Browsers
- Mainstream Services Toolkits