ebXML and Web Services Overview

Dr. David Wai-lok Cheung (張偉犖博士)
Center for E-Commerce Infrastructure Development
The University of Hong Kong
Agenda

- Background of ebXML
- ebXML Movement
- ebXML Architecture
ebXML Mission

“To provide an open XML-based infrastructure enabling the global use of electronic business information in an interoperable, secure and consistent manner by all parties”

Technically: an asynchronous web service
Background

- A joint global initiative by
  - **UN/CEFACT** – United Nations Center For Trade Facilitation And Electronic Business
  - **OASIS** – Organization for the Advancement of Structured Information Standards
- An 18-month project, with version 1 specs completed in May 2001
  - Open-source implementations are undergoing in some technical committees
  - Companies are developing ebXML products, e.g. Innodigital, XML Global
Example of ebXML Adoption

- Pan-Asian Alliance formed by five e-commerce service providers:
  - Tradelink (Hong Kong)
  - Infoshare (China)
  - Trade-Van (Taiwan)
  - KTNET (Korea)
  - SNS (Singapore)

- Combined membership of over 120K organizations

“...we finally decide PAA Technical Architecture based on ebXML...ebXML is the only global e-business framework.”
ebXML Asia Committee

- Congregation of EC/EB experts with the mandate to spread ebXML in the Asian region
to contribute to ebXML work and promote the implementation of ebXML in the Asian region cooperatively
- Reps from Chinese Taipei, Hong Kong, Malaysia, Japan, Korea, Pakistan, Singapore, Sri Lanka and Thailand
- 7th meeting in Hong Kong in November hosted by HKU CECID
Recent Development

- Covisint (global B2B supplier exchange) implements ebXML for Automotive Industry
- Korea Institute for Electronic Commerce and the Finnish Information Society Development Center cooperate on the practical implementation of ebXML.
- OpenTravel Alliance Endorses ebXML
- RosettaNet Supports ebXML Messaging Services in RNIF
- Global Commerce Initiative (GCI) Adopts ebXML as the Backbone of Data Exchange Standard for B2B Trade in the Consumer Goods Industry
- Japan Electronics and Information Technology Industries Association (JEITA) adopts ebXML Messaging Services in Collaborative EDI project (NEC, Pioneer, Sony, Toshiba, …)
freebxml.org open-source initiative

- Pronounced as ‘free bee XML’
- Initiated by CECID R&D Team and Farrukh Najmi (PM of ebxmlrr team)
- foster the development and adoption of ebXML and related technology
- Founding members
  - Cyclone Commerce (USA)
  - Kasetsart University (Thailand)
  - Korea Institute for Electronic Commerce (Korea)
  - Sterling Commerce (USA)
  - Sun Microsystems (USA)
  - Sybase (USA)
  - XML Global (Canada)
  - University of Hong Kong (Hong Kong, China)
- Hosted by CECID to publish open-source ebXML products
- Welcome participation of other organizations who donate ebXML open-source products
ebXML Architecture
Key Specifications

- Messaging Services
  - Or Transport, Routing, Packaging (TRP)
- Registry / Repository
- Business Process Specification Schema
- Collaboration Protocol Profile / Agreement
ebXML Business Process Model

- Profile of Company A
- Profile of Company B
- Collaboration Protocol Agreement (CPA)
- Transport, Routing and Packaging of Business Documents
Usage: Design Time

- Industry specialist
- Business manager
- Software developer
- Business Modeling
- Software Construction
- BPS
- XML schemas
- CPP
- ebXML System
- Registries / Repositories
- Enterprise deployment

Industrial service provider
Enterprise
Software vendor

Center for E-Commerce Infrastructure Development
Usage: Runtime

1. Publish company profile

2. Discover company B

3. Download B’s profile

4. Negotiate CPA

5. Install CPA

6. Transact with TRP

Company A

Company B

ebXML System

CPA

Reg / Rep
Other Alternatives?

- Web doesn't encourage B2B e-commerce automation
- EDI – designed 20 years ago, too expensive to deploy, not on Internet, not for SMEs
- Web services: J2EE, .NET, SOAP, WSDL, UDDI, WSFL... – proprietary, only tackling particular technical problems
- ebXML are actually putting the above together into a framework
Messaging Services

- Also known as: **Transport, Routing, Packaging**

- **Transport**
  - Supports multiple Internet protocols: HTTP, SMTP, FTP
  - Supports secure transport channels, e.g. SSL for HTTP, S/MIME for e-mail

- **Packaging**
  - SOAP with Attachments
  - Uses XML signature to sign a message

- **Routing**
  - Supports multiple hops of message handlers
ebXML Message Service (ebMS)

- Ensure a reliable, secure and open business document exchange infrastructure
- Packaged as “SOAP with Attachments (SWA) extension”
- A Message Package is a MIME envelope with two logical MIME parts, namely **Header Container** and **Payload Container**
ebXML Message Structure

Communications Protocol Envelope (HTTP, SMTP, etc.)

SOAP with Attachments MIME Envelope

MIME Part

SOAP-ENV: Envelope

SOAP-ENV: Header

eb: MessageHeader

eb: TraceHeaderList

other: Etc ...

SOAP-ENV: Body

eb: Manifest

eb: Etc ...

other: Etc ...

MIME Part(s)

Payload(s)

Center for E-Commerce Infrastructure Development

Message Package

Header Container

Payload Container(s)

Extracted from ebXML Message Service Specification
Header Container

* Contain a SOAP 1.1-compliant message with a SOAP header and SOAP body

* SOAP header
  * includes the traditional functions found in business message headers, such as identification of the parties to the transaction

* SOAP body
  * carries data cataloging the message contents, which is called a manifest in ebXML parlance
Payload Container

- A payload container contains a series of payload (MIME) attachments
- A payload attachment contains a business document
- Multiple payloads are allowed
- A payload that can be in XML file or any other digitized format
Sample ebXML Message
extracted from ebXML Message Specification

POST /servlet/ebXMLhandler HTTP/1.1
Host: www.example2.com
Content-Type: text/xml; charset="utf-8"
Content-Length: 628
SOAPAction: "ebXML"
Content-type: multipart/related;
            boundary="BoundarY"; type="text/xml";
            start= "<ebxhmheader111@example.com>"
--BoundarY
Content-ID: <ebxhmheader111@example.com>
Content-Type: text/xml
......
<?xml version="1.0" encoding="UTF-8"?>
  <eb:MessageHeader eb:id="..." eb:version="2.0" SOAP:mustUnderstand="1">
    <eb:From>
      <eb:PartyId>http://example.com</eb:PartyId>
      <eb:Role>http://rosettanet.org/roles/Buyer</eb:Role>
    </eb:From>
    <eb:To>
      <eb:PartyId>QRS543</eb:PartyId>
      <eb:Role>http://rosettanet.org/roles/Seller</eb:Role>
    </eb:To>
    <eb:CPAId>http://www.oasis-open.org/cpa/123456</eb:CPAId>
    <eb:ConversationId>987654321</eb:ConversationId>
    <eb:Service eb:type="myservicetypes">QuoteToCollect</eb:Service>
    <eb:Action>NewPurchaseOrder</eb:Action>
    <eb:MessageData>
      <eb:MessageId>UUID-2</eb:MessageId>
      <eb:Timestamp>2000-07-25T12:19:05</eb:Timestamp>
      <eb:RefToMessageId>UUID-1</eb:RefToMessageId>
    </eb:MessageData>
    <eb:DuplicateElimination/>
  </eb:MessageHeader>
</SOAP-ENV:Envelope>
<SOAP-ENV:Body>
  <eb:Manifest eb:id="Manifest" eb:version="2.0">
    <eb:Reference eb:id="pay01"
      xlink:href="cid:payload-1"
      xlink:role="http://regrep.org/gci/purchaseOrder">
      <eb:Schema
        eb:location="http://regrep.org/gci/purchaseOrder/po.xsd"
        eb:version="2.0"/>
      <eb:Description xml:lang="en-US">
        Purchase Order 10,000 widgets</eb:Description>
    </eb:Reference>
  </eb:Manifest>
</SOAP-ENV:Body>
--BoundaryContent-ID: <cid:payload-1>
Content-Type: text/xml
<?xml version="1.0" encoding="UTF-8"?>
<purchase_order>
  <po_number>1</po_number>
  <part_number>123</part_number>
  <price currency="USD">500.00</price>
</purchase_order>
--Boundary--
The *ebXML Message Service* may be conceptually broken down into the following three parts:

- an abstract *Service Interface*,
- functions provided by the Message Service Handler (MSH)
- the mapping to underlying transport services

Best-effort

Once-and-only-once
ebXML Reliable Messaging Services

- handles the delivery and acknowledgment of ebXML Messages
- handling for persistence
- duplicate elimination
- error notification
- acknowledgment
- retry
ebMS Recent Movement

- OASIS approved ebMS Specification version 2.0 as a standard on 5 Sep, 2002
- Sybase incorporated ebMS v2 in Web Services Integrator product
- CECID launched ebMS v2 open-source product – Hermes on 13 Sep, 2002
References

- XML Signature, http://www.w3c.org/Signature/
References

Agenda

■ ebXML Deployment Stages
■ Case Study
  ■ e-Logistics Management (XMLDG)
  ■ B2G e-Submission (IEPPXML)
  ■ B2B e-Procurement (MTRC & Saggio)
ebXML Deployment Stages

- Internal adoption of XML business vocabulary
- Exchange of XML documents with trading partners via ebXML TRP
- Use of ebXML Registry within an organization or within an e-commerce community
- Automation of business processes with ebXML BPSS
e-Logistics Management

Electronic Submission of Dangerous Goods Manifests
XMLDG : Background

- A shipping company must submit a **Dangerous Goods (DG) Manifest** to Marine Department >48 hours before the arrival of the vessel required by law
- The manifests will be scrutinized and acknowledged by fax **within the next working day** of submission
- Paper-based and Web-based submission
  - Web submission launched since 1998
  - ~4600 submissions per month
  - 95% of submissions already use Web
Current Process

Shipping Company

Web submission

Internet

Marine Dept

Shipping Company IS System

1 Paper submission

2 Web submission
XMLDG : Project

- Facilitates shipping companies to automate submission of DG Manifests to Marine Dept in form of XML messages

- DG messages will be sent from shipping companies to Marine Department using the ebXML messaging service protocol through the Internet

- **International Marine Organization (IMO)** recommends EDI or XML messages (EDI/XML conversion)

- Expected project duration: Jun – Dec 2002
XML submission

ebXML Message Service over HTTPS

Paper submission

Internet

Paper submission

Web submission

Marine Dept

XMLDG Project

ebXML Message Packaging Library

Shipping Company IS System

ebXML Message Service Handler

Shipping Company IS System
XMLDG : Project Participants

- Hong Kong SAR Government
  - Marine Department
  - Information Technology Services Department

- The University of Hong Kong
  - Center for E-Commerce Infrastructure Development (CECID)

- Two shipping companies
  - Orient Overseas Container Line (OOCL)
  - American President Line (APL)
  - ~10% of submissions (~500 submissions per month)
XMLDG: Project Scope

- Working group to define a XML document schema for Dangerous Goods Manifest
- MD to deploy an Internet message gateway for receiving XML DG messages
  - May use ebXML Message Service Handler provided by CECID
- Shipping companies to generate XML DG messages directly from backend systems
  - May use XML schema packaging library provided by CECID
  - Or write their own program to generate an XML document
- All ebXML messages are exchanged through HTTPS protocol
XMLDG : Benefit

- Double data entry eliminated
  - Some large shipping companies have one full time staff typing data to DG Web form according to printouts from backend systems
  - Higher productivity and accuracy

- DG documents could be digitally signed (or password-authenticated) and encrypted
  - Better security
B2G e-Submission

Electronic Submission Prototype of Pharmaceutical Import/Export License Forms in XML Messages
IEPPXML: Background

- Users: pharmaceutical companies
- Paper submission of Import/Export license forms of pharmaceutical products
- Processing time for each submission: 2 days
- ~7,000 submissions per year
- Applicants usually submit forms and collect processed applications in person
IEPPXML: Pilot Project

To develop a prototype to study the feasibility of electronic application of pharmaceutical import/export licenses using XML/ebXML

Participants

Hong Kong SAR Government
- Department of Health (DH)
- Information Technology Services Department (ITSD)

The University of Hong Kong
- Center for E-Commerce Infrastructure Development (CECID)
Manual entry mode through **ebMail**

**Message generation mode**

**Internet**

**Dept of Health**

**ebXML Message Service Handler**

**pharmaceutical companies**

**Manual entry mode through ebMail**

**Message generation mode**

**Dept of Health**

**ebXML Message Service Handler**

**pharmaceutical companies**
IEPPXML : ebMail

- Provides GUI to prepare electronic business documents (XML format)
  - Plug-in can be downloaded to extend ebMail to handle different document types and business processes
- Sends and receives business documents through email using ebXML messaging standard
- Signs and encrypts documents to enable sender authentication and data confidentiality
- Communicates with ebXML Registry to access shared business data
- Supports adaptors for data import and export for external applications, e.g. MS Excel, web browser
IEPPXML : Project Scope

- To define XML document schemas for import and export license forms and corresponding ebXML packaging library
- To develop a pharmaceutical license plug-in for ebMail
- To deploy a message gateway (ebXML message service handler) in DH for processing email (ebXML) messages of licensing applications
- All ebXML messages exchange are through SMTP protocol
IEPPXML: Benefits

- **For pharmaceutical companies:**
  - Paperless processing: faster, lower cost and more accurate
  - Save the applicants for traveling twice to submit the application and collect the licenses

- **For DH:**
  - Paperless processing
  - Single (XML messaging) gateway to collect licensing submissions prepared by manual data entry / generated by systems
  - Extensible for other applications
B2B e-Procurement

Office Supplies Procurement between MTRC & Saggio
Background

- MTRC procures office supplies from Saggio
- Each MTRC department orders office supplies individually on SaggioDirect Website with an authorization structure in place
- Every month, MTRC Finance Office receives invoices and statements from Saggio, and manually
  - Reconciles invoices and statements
  - Enters data to create accounts payable records in its accounting system
- How to automate manual processing of invoices and statements with ebXML?
Current Process
<table>
<thead>
<tr>
<th>Step</th>
<th>Process</th>
<th>Saggio</th>
<th>MTRC Procurement</th>
<th>MTRC Users</th>
<th>MTRC Financial Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Procurement agrees pricing with Saggio</td>
<td>Negotiate Price List</td>
<td>Negotiate Price List</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Saggio posts product catalogue to Saggio Direct</td>
<td>Post Price List</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Cost Center Head / Delegate to set up user accounts and</td>
<td>Setup Account</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>authority limits and authorization hierarchy in Saggio Direct</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>User selects the required items into shopping cart in</td>
<td>Select Item</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Saggio Direct and becomes a purchase request</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Saggio Direct will notify the appropriate authorizer to</td>
<td>Authorize Purchase</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>authorize the purchase if the cost exceed the authority limit of the</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>requester</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>The authorizer may modify the purchase details or reject it</td>
<td>.Receive Delivery</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>through Saggio Direct</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Saggio Direct notifies the requester when an order is</td>
<td>Notify Requestor</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>authorized and received by Saggio</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Goods are usually received on the next day and the one who receives</td>
<td>Handle Request</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>the goods will sign on the hard copy delivery note. User will keep a</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>copy of delivery note</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Non-conforming delivery and damaged goods will be handled manual</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>between the user and Saggio</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>User can check status of request through Saggio Direct.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>However, expediting has to be done manually</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>At each month end, Saggio sends the monthly statements and</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Invoices to Financial Control. The monthly statements and Invoices will</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>be batched by Property Management and Railway Operation.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Financial Control distributes the monthly statements and</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>invoices to Cost Centre heads. Also, a reconciliation between the</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>monthly statements and Invoices is required by Financial Control.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Cost Centre Head / Delegate check the monthly statement</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Cost Centre Head / Delegate follows up with Saggio for</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>revision of statement if there is any discrepancy found in</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>the statement</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Cost Centre Head endorses / verifies / authorizes the</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>statement and forward it to Financial Control</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Financial Control creates a non-order invoice, consolidated</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>by the total amount of statement, in Accounts Payable</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Automated Process
1. MTRC users place an order through SaggioDirect.com
2. Saggio delivers goods to MTRC
3. Saggio prepares invoice related documents in XML format
4. Saggio sends XML documents
5. MTRC receive XML documents
6. MTRC process XML documents and stores data into database
7. MTRC users can check data through Intranet application
System Requirements for MTRC (1)

- XML Business Vocabulary for statements and invoices
  - Definition of XML schemas
  - Data extraction from received XML documents
- ebXML Message Service Handler for MTRC to receive statement and invoice documents in XML
- Database to store invoice and statement
Intranet front-end for MTRC Finance Office to:
- Consolidate statement data
- Generate AP Interface for Oracle Financials

Intranet front-end for statements for departmental user to:
- Check invoices statements and
- Reconcile statements and invoices
- Endorse / verify / authorize statements
System Requirements for Saggio

- XML Business Vocabulary for statements and invoices
- Composition of statement and invoice XML documents
- ebXML Message Service Handler for OSS to send statement and invoice XML documents
MTRC side architecture

Unix based server with 3 java programs installed

1. **Message Service Handler** (provided by CECID)
   - receive ebXML document thru HTTP/HPPTS
   - pass ebXML document to Document Processor

2. MTRC Document Processor
   2.1 **Data Packaging Handler** (provided by CECID)
      - receive ebXML document from Messaging Service Handler thru HTTP
      - extract data from ebXML document
   2.2 **Data Insertion Handler** (provided by CECID)
      - talk to Data Packaging Handler thru Java API
      - store data into database

Windows 2000 with IIS installed
1. **ASP intranet front-end programs** (provided by CECID)

User access intranet front-end program using web browser
Web Services

Dr. David Wai-lok Cheung

Center for E-Commerce Infrastructure Development
The University of Hong Kong
Agenda

- Review on Distributed Computing
- Definitions of Web Services
- Features of Web Services
- Web Services Architecture
- SOAP and Its Extensions
- Vendor Tools on Web Services
- The Semantic Web
- ebXML vs. Web Services
Review on Distributed Computing
A Brief Review on Distributed Computing (1)

- 1980’s
- Not much care about network communications
- Concerns about communications between applications in the same machine
- It is already challenging enough
A Brief Review on Distributed Computing (2)

- Early 1990’s
- Component Object Model (COM) by Microsoft
- Common Object Request Broker Architecture (CORBA) by Object Management Group (OMG)
- Stand-alone PC ruled the Earth
A Brief Review on Distributed Computing (3)

- Mid 1990’s
- CORBA/IIOP (Internet inter-ORB Protocol) standard wired on TCP/IP
- Connection-oriented and stateful programming model
- High overhead, not scalable
A Brief Review on Distributed Computing (4)

- DCOM (Distributed COM)
- Developed by Microsoft, proprietary to Windows platform
- Conceptually similar to CORBA/IIOP
- Increased complexity to deal with other tasks such as garbage collection and security implementation
A Brief Review on Distributed Computing (5)

- RMI (Remote Method Invocation)
- Developed by Sun Microsystems, tightly bound to a single language platform – Java
- Connectionless protocol – more scalable
- Also implements garbage collection and is quite security conscious
Synchronous vs. Asynchronous Communication

- Both DCOM and RMI are request/response oriented, i.e. synchronous communication
- Java Message Service (JMS) and Microsoft Message Queuing (MSMQ) are introduced to provide asynchronous communication
- Advantages: guaranteed delivery, easy and fast inter-application communication
Definitions of Web Services
What are Web Services?

“Web Services are modular applications that are self-describing and that can be published, located, and invoked from anywhere on the Web or within any local network based on open Internet standards.”

- Professional XML Web Services, Wrox Press
Definition

“XML Web Services are the fundamental building blocks in the move to distributed computing on the Internet. Open standards and the focus on communication… Applications are constructed using multiple XML Web services from various sources that work together regardless of where they reside or how they were implemented.”

- Microsoft Developer Network
Features of Web Services
Features of Web Services

- Exposing functionality through a standard Web protocol
- Providing a way to describe the interfaces in enough detail to allow a user to build a client application to talk to them
- Registering the services so that potential users can find them easily
Web Services Operation Flow

1. Publish

2. Find Service

3. Service Information Response

4. Bind to Service

5. Use Service
Web Services Architecture
Web Services Architecture

- Basic technologies:
  - eXtensible Markup Language (XML)
  - Simple Object Access Protocol (SOAP)

- Three main building blocks:
  - Discovery
  - Description
  - Invocation
Web Services Stack

- **WSFL / XLANG** (Service Flow)
- **UDDI** (Service Delivery)
- **UDDI** (Service Publication)
- **WSDL** (Service Description)
- **SOAP** (XML Packaging)
- **HTTP/FTP/SMTP** (Transport)
Web Services – Discovery

- Analogous to search engines, or DNS for Web Services
- Universal Description, Discovery and Integration (UDDI)
- Industrial initiative developed by UDDI.org (> 220 members)
- Describes how a provider can advertise the existence of their Web Services in a directory
- Uses Simple Object Access Protocol (SOAP) as transport layer
- Client may access the registry using Java API for XML Registries (JAXR)
UDDI (1)

- **Universal Description, Discovery and Integration**
- Version 3 was released on 19 July 2002
- Three basic functions: publish, find and bind
- “register once, published everywhere” concept
- UDDI documents include specifications for an API for automated interactions with a UDDI-registered site
- Three UDDI test nodes
UDDI (2)

- UDDI proposes 3 ways of listing companies in a registry:
  - **White pages**, or basic identification: name, address, and key points of contact
  - **Yellow pages**, or classification by a standard index of business and industries
  - **Green pages**, or technical capabilities and services related to the conduct of electronic business
Web Services – Description

- Description of the messages the Web Service can accept and generate

- Two levels:
  - Description of data types and data structures
    - Using XML Schemas
  - Description of the service
    - Using Web Services Description Language (WSDL)
WSDL (1)

- Web Services Description Language
- Initiated by IBM, Microsoft and Ariba
- Describes
  - Reliability
  - Capabilities
  - Sequencing of messages
  - Who sends what message
  - At what time
The services defined in WSDL have five major elements:

- **Ports** – the network addresses of the message senders and receivers, called end-points
- **portType** – the kind of operations supported by the senders and receivers
- **Binding** – the protocol and data format specifications defined by the portType
- **Message** – the data exchanged between end-points
- **Types** – data types defined in the messages exchanged between end-points
Web Services – Invocation

- Key component – Simple Object Access Protocol (SOAP)
- SOAP is simple and extensible computer-to-computer communication protocol that leverages existing Internet standards:
  - XML for message formatting
  - HTTP and other Internet protocols for message transport
SOAP and Its Extensions
SOAP

- Simple Object Access Protocol
- The SOAP specification contains:
  - A syntax for defining messages as XML documents
  - A model for exchanging SOAP messages
  - A set of rules for representing data within SOAP messages, i.e. SOAP encoding
  - A guideline for transporting SOAP messages over HTTP
  - A convention for performing remote procedure calls (RPC) using SOAP messages
SOAP Message Example (request)

POST /endpoint.asp HTTP/1.1
Content-Type: text/xml
Content-Length: ###
SOAPAction: "urn:wroxheroes"

<soap:Envelope
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    Soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <soap:Header>
        <h:from xmlns:h="http://www.wrox.com/Header"
            soap:actor="http://www.wrox.com/Header">
            SoapGuy@wrox.com
        </h:from>
    </soap:Header>
    <soap:Body>
        <w:GetSecretIdentity
            xmlns:w="http://www.wrox.com/heroes/"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <w:codename>XSLT-Man</w:codename>
        </w:GetSecretIdentity>
    </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: ###

<soap:Envelope
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    Soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <soap:Body>
        <w:GetSecretIdentityResponse
            xmlns:w="http://www.wrox.com/heroes/"
            xsi:type="xsd:string">Michael Kay</return>
        <w:GetSecretIdentityResponse>
    </soap:Body>
</soap:Envelope>
SOAP Extensions

- Additional modules of functionality that can be “plugged-in” to the protocol
- Implemented as additional header entries within the SOAP message

Some most desired SOAP extensions:
-Attachments
- Routing/Intermediaries
- Reliable Messaging
- Security
- Quality of Service (QoS)
- Context/Privacy
- Transactions Support
SOAP extension: Attachment

- A standard way of attaching or including non-XML or binary files
- Content is often encoded with MIME (Multipurpose Internet Mail Extensions)
- Binary contents are Base64 encoded

Three key ways of handling binary content
- Encoded content included within a SOAP message
- Encoded content attached with a SOAP message via MIME
- Link reference to the binary content (similar img tag in HTML page)
SOAP extension: Routing/Intermediaries

- Related to the process of routing SOAP message through intermediaries
- The SOAP routers at message level (not in network level)
- Address **scalability** issue in Web Services
- Two kinds of intermediary servers
  - Caching
  - Store-and-forward
SOAP extension: Reliable Messaging

- Capability to guarantee delivery
- Need to address:
  - When delivery fails, how many retry before giving up
  - Determine duplicate received message
- Not only related to a transaction
- Ensure a message requesting a transaction or multiple transaction is processed only once
SOAP extension: Security

- Deal with message level security only
- Transport level security already available (e.g. apply SSL, which turn HTTP to HTTPS)
- XML Signature
  - Who sent the message
  - Was the message altered on-route to receiver
- XML Signature mechanism is independent of specific encryption technology, and key management protocol
SOAP extension: Quality of Service (QoS)

- Satisfaction of the user about overall service performance
- Service Level Agreements (SLAs) determine the fine details
- Refer to the reliable messaging XML Web Services extension
  - Service uptime
  - What actions to be taken if service down or fails
- Current Web Services do not provide standard mechanism to address SLAs and the QoS
SOAP extension: Context/Privacy

- Related to **Intelligent Web Services**
  - Context sensitivity
  - Share context with other services

- By knowing the situation of a Web Service, intelligent decisions can be made automatically on behalf of the user

- Provide service based on user profile context information

- Highly ties with privacy issue
SOAP extension: Transaction Support

- Permit a group of actions to either all succeed or all fail as a single unit
- Short-running transaction
  - Such as the ones found in database
- Long-running transaction
  - Web services transactions that span transaction infrastructures, models and domain
- Both types of transaction need to address in SOAP
Vendor Tools on Web Services
Vendor Tools and Frameworks

- e-Speak of HP
- IBM
  - UDDI4J, Websphere, SOAP toolkit
- Microsoft
  - .NET
- Sun Microsystems
  - Sun Open Network Environment (ONE)
  - JAX Pack: JAXP, JAXB, JAX-RPC, JAXM, JAXR
The Semantic Web
The Semantic Web (1)

“The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.”

– Tim Berners-Lee (creator of Web), James Hendler, Ora Lassila
The Semantic Web (2)

- The second generation of Web will slowly evolve into its new form, which the content is meaningful to computers.
- The Semantic Web would have content and services that contain description understandable to computers, based on XML, RDF, Web Services and some of the other XML based technologies.
ebXML vs. Web Services
ebXML vs. Web Services (1)

- At first sight, these 2 technologies are competing with each other
- The truth is that ebXML is an example of a Web Service
- ebXML
  - Focused on scalable and secure automation of B2B collaborative processes
  - More on contract based
- Web Services
  - Focused on a more general framework for automation of any web transaction
  - More on service based
# ebXML vs. Web Services (2)

<table>
<thead>
<tr>
<th></th>
<th>ebXML</th>
<th>Web Service</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Background</strong></td>
<td>EDI and B2B Collaboration</td>
<td>Internet / Web Pages</td>
</tr>
<tr>
<td><strong>Initiative</strong></td>
<td>Standard bodies (UN/CEFACT &amp; OASIS)</td>
<td>Vendors (IBM, Microsoft, etc)</td>
</tr>
<tr>
<td><strong>Objective</strong></td>
<td>Business process automation</td>
<td>Web transaction automation</td>
</tr>
<tr>
<td><strong>Focus</strong></td>
<td>Business – strong focus on process and semantics</td>
<td>Technology – focus on service and transport</td>
</tr>
</tbody>
</table>
### ebXML vs. Web Services (3)

<table>
<thead>
<tr>
<th>Category</th>
<th>ebXML</th>
<th>Web Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registry/Repository</td>
<td>ebXML RIM v2.0 &amp; ebXML RSS v2.0 (v3.0 in development)</td>
<td>UDDI v2.0 (v3.0 released for comments)</td>
</tr>
<tr>
<td>Business Process</td>
<td>ebXML BPSS</td>
<td>Proprietary XLANG (Microsoft), WSFL (IBM)</td>
</tr>
<tr>
<td>Semantic</td>
<td>Core Components, UBL</td>
<td>None</td>
</tr>
<tr>
<td>Messaging</td>
<td>ebXML MS v2.0 (based on SOAP 1.1 with security and reliability)</td>
<td>SOAP, SOAP 1.2 (reliability) in development</td>
</tr>
</tbody>
</table>
References

- http://www.uddi.org
- Microsoft Online MSDN Library
- E-Business Standards – First Steps Towards Scalable Interoperability, Red Wahoo
The End

Web Services