

15.561  
Information Technology Essentials

# Session 14

## Systems that span multiple enterprises

Acknowledgments:

Slides marked "SM" are adapted from Stuart Madnick, MIT.

Slides marked "CD" are adapted from Chris Dellarocas, MIT.

# Three generations of the Web

- **1<sup>st</sup> generation (early 1990's)**
  - You see static web pages
  - Needed: URL, HTTP, HTML
- **2<sup>nd</sup> generation (late 1990's)**
  - Simple interactions between humans and remote applications
  - E.g., You fill out a form on a web page to order a product
  - Needed: Forms, CGI, ASP, Cookies, Java applets, ...
- **3<sup>rd</sup> generation (2000's)**
  - Complex interactions among humans and multiple agents (services)
  - E.g., Your agent interacts with many other remote agents to book your travel reservations
  - Needed: XML, SOAP, UDDI, Semantic Web, ...

# Problem:

## What does a web page “mean”?

- HTML specifies appearance, not meaning.
- That’s fine, if the only readers of the information will be humans.
- But if you want your agents (programs working on your behalf) to be able to process the information, they need to be able “understand” what is on a web page.
- It is possible to write “screen scrapers” that know the formats of each different web page and pick out the data they need.
- But this is very tedious and unstable.

## **Better solution: Machine readable web pages**

- **The information can be stored in a format that**
  - **is easy for computers to “understand”**
  - **has a “standardized” meaning upon which many people agree**

# What is XML?

- XML - eXtensible Markup Language
- a meta-language that allows groups of users to create and share their own vocabularies of machine-understandable elements (indicated by “tags”) within a document
  - E.g., a “customer” document might include “name”, “address”, etc.
- Provides an infrastructure for creating “interoperable” computer applications that cross firm boundaries

# XML Example: Companies

<Company>

<CompanyID=642>

<Ticker>WCOM</Ticker>

<Company\_name>MCI WorldCom Inc</Company\_name>

<Market\_cap>\$100,721,720</Market\_cap>

<Competitor\_reference=542/>

<Competitor\_reference=573/>

</Company>

<Company>

<CompanyID=542>

<Ticker>T</Ticker>

<Company\_name>AT&T Corp</Company\_name>

<Market\_cap>\$114,815,359</Market\_cap>

<Competitor\_reference=642/>

</Company>

# What do you (and your agents) need to understand an XML document?

- Two alternatives:
- **DTD – Document Type Definition**
  - Specifies the kinds of elements and attributes that are allowable within a given type of XML document
  - Very limited options for expressing what kinds of values are valid (e.g., cannot require a field to contain a date or numeric value)
- **XML Schema**
  - Much more flexible way of specifying what kinds of values can be included in a given type of XML document
  - E.g., date, time, integer, or combination of other types

# DTD fragment for previous example

...

```
<!ELEMENT Company (CompanyID, Ticker, Company_name,  
Market_cap, Competitor_reference*)>
```

```
<!ELEMENT CompanyID #PCDATA>
```

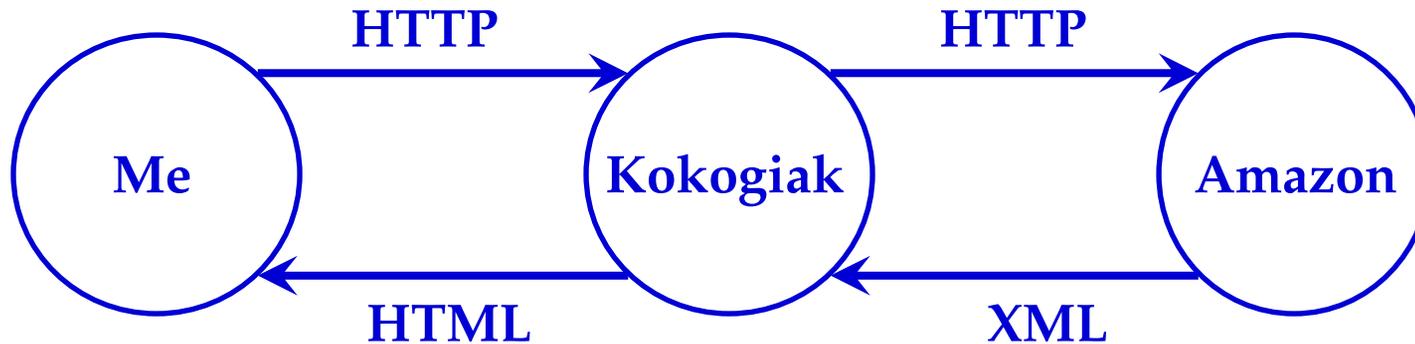
```
<!ELEMENT Competitor_reference CompanyID>
```

\* = zero, 1, or several

## Example: Custom website using “backend” from Amazon

- Request to Amazon web service for best-selling books on dogs (result is XML document containing information about dog books):
  - <http://xml.amazon.com/onca/xml2?t=webservices-20&dev-t=D2ED5GR7A6RZ7Y&KeywordSearch=dogs&mode=books&type=lite&page=1&f=xml>
- Web site that uses the Amazon web service to create a customized on-line store:
  - <http://www.kokogiak.com/amazon4/>

# Simple Web Services Architecture



- **Other examples:**
  - [www.googlealert.com](http://www.googlealert.com)
  - [www.googleduel.com](http://www.googleduel.com)
  - [www.ipilot.net](http://www.ipilot.net)

# RDF

## Resource Description Framework

- A precise XML format for making statements about the world
- E.g.:
  - » The product described on [www.example.com/id0304](http://www.example.com/id0304) is a car.
  - » A car is a kind of motor vehicle.
  - » Richard Schmalensee thinks that [www.example.com](http://www.example.com) is a “great website”.

# Example: RDF definitions of different kinds of vehicles

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema" >

  <rdf:Description rdf:ID="MotorVehicle">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema Class"/>
  </rdf:Description>

  <rdf:Description rdf:ID="PassengerVehicle">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema Class"/>
    <rdfs:subClassOf rdf:resource=" MotorVehicle"/>
  </rdf:Description>

  <rdf:Description rdf:ID="Truck">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema Class"/>
    <rdfs:subClassOf rdf:resource=" MotorVehicle"/>
  </rdf:Description>

</rdf:RDF>
```

From: Frank Manola, Eric Miller, *RDF Primer*, W3C Working Draft, January 23, 2003, [www.w3.org/TR/rdf-primer/](http://www.w3.org/TR/rdf-primer/)

# Ontology

A set of statements (and usually rules for reasoning) about the world

E.g.:

- » Publications (e.g., books, magazines, articles) and their properties (e.g., titles, authors, reviews, reprintings)
- » Vehicles (e.g., cars, trucks, vans) and their properties (e.g., prices, owners, manufacturers, repair records)

# Example ontology

## CIA World Factbook information on the United States

– In HTML:

[www.daml.org/2001/12/factbook/us.html](http://www.daml.org/2001/12/factbook/us.html)

– In earlier language upon which OWL (Ontology Working Language of W3C) is based:

[www.daml.org/cgi-bin/hyperdaml?http://www.daml.org/2001/12/factbook/us.daml](http://www.daml.org/cgi-bin/hyperdaml?http://www.daml.org/2001/12/factbook/us.daml)

# Web Services

- **Technologies for making it easy for computer programs to automatically share information (and computing tasks) with each other over the Internet**
- **An approach to developing software that takes advantage of these technologies**

Adapted from: Frank Coyle, *XML, Web Services, and the Data Revolution*, 2002.

# A web service

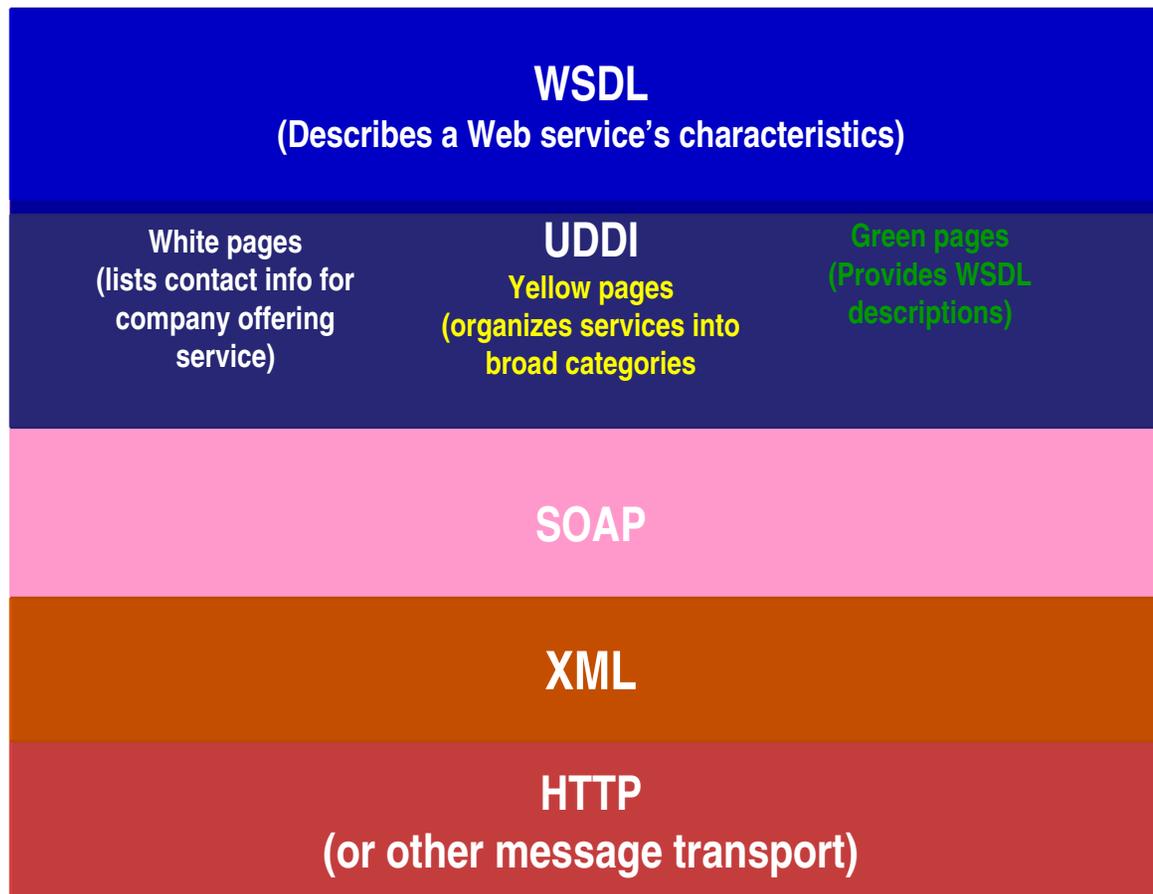
- A self-contained, server-based program
- Designed to perform a single function
- With interfaces expressed in a standard form (based on XML)
- Published in such a way that it can be called upon remotely (generally via HTTP)

Adapted from: PricewaterhouseCoopers, *Technology Forecast: 2002-2004*, Volume 1, 2002.

# Examples of possible web services

- Credit card authorization
- Currency converter (e.g., dollars to euros)
- Stock quote provider
- Shipping rate calculator
- Insurance rate quotes
- ...

# Web Services Protocol Architecture



Adapted from PricewaterhouseCoopers, *Technology Forecast: 2002-2004*, Volume 1, 2002. pp. 596.

# SOAP

- Simple Object Access Protocol
- Includes:
  - A format for specifying messages in XML
    - » including elements called: envelope, header, and body
  - Rules for servers to follow when they receive a SOAP message

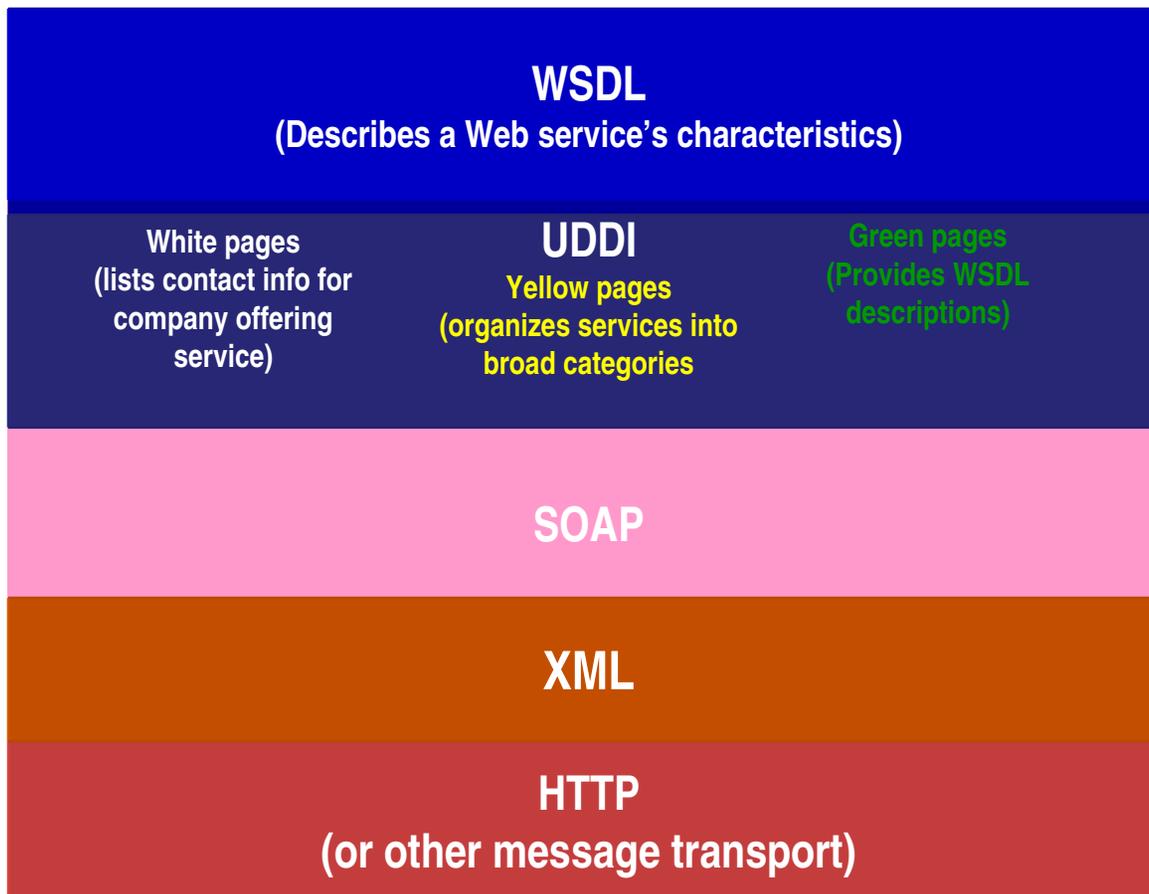
# UDDI

- **Universal Discovery, Description, and Integration**
- **Rules for responding to certain SOAP messages to provide information about available web services in 3 ways:**
  - **Yellow pages:** services listed by categories such as geography, industry, functionality
  - **White pages:** contact info for a given service
  - **Green pages:** WSDL descriptions for how a given service works

# WSDL

- **Web Services Description Language**
- **Describes the types of messages a service receives and the types of messages it returns in reply**
- **Together with UDDI, allows automated agents to assemble collections of web services “on the fly” to perform a given task**

# Example: Semantic Web Travel Agent



Agent finds airlines, condo rental agencies, car rental agencies, etc. using UDDI and WSDL.

Agent asks questions and gets answers via SOAP.

The questions and answers are formatted in XML.

The messages are transported via HTTP.

# How will programmers implement web services?

Two alternative development environments:

- .Net - Microsoft
- J2EE (Java 2 Enterprise Edition) – Sun, IBM, Oracle, and others

# Two ways of using web services

- **Inside a single company**
  - Hagel & Brown: Start at the edge and move toward the core
- **Between companies**

# Strategic implications for IT industry

- Web/browsers become a single front end through which a diverse set of applications can be accessed
- Web/browsers become an execution environment for B2B/web applications
  - The “Webtop” instead of the “desktop”
  - The Webserver instead of operating system
  - Implications for Microsoft/Windows?
  - Hence big efforts/fights about Web standards

# Strategic implications for other industries

<b>Role</b>	Aggregator		Orchestrator		Shaper
<b>Leveraged Growth Platform</b>	Value-added service portfolios	Vendor-sponsored communities	Process networks		Economic webs
			Closed	Open	
<b>Examples</b>	Schwab	IBM/Oracle	Cisco/Nike	Li & Fung	Microsoft/Intel

Adapted from *Out of the Box*, John Hagel III, Harvard Business School Press pp. 161.

**What else could you do?**