The Final Assignment...

- Your own application
  - Discussion board
    - Think: Paper summaries
  - Web cam proxy
    - Think: George Orwell or JenCam
  - Visitor announcement and tracking
    - Look at 7th floor lobbies at 715 Broadway
The Final Assignment... (cont.)

- Ground rules
  - Implemented as a web proxy or SOAP service
    - For SOAP services you need to implement a client as well
  - Built on top of Munin, uses HTTP/1.1
    - A continuation of the semester-long group effort
  - One paragraph summary due 11/18
The Final Assignment... (cont.)

- Due 12/2 before class
  - In-class talk and live demo
  - 20 minutes per group
  - 3 introductory slides
    - Application overview (what does it do?)
    - Implementation overview (how does it work?)
    - Lessons learned (what did you get out of it?)
The overall challenge: Provide metadata
  - How to find services?
  - How to access services?
  - How to compose services?

Today’s journey
  - WSDL as the IDL for web services
  - RDF as a general description language
  - OWL as a way to reason about descriptions
WSDL Overview

- WSDL provides a contract between clients and services
  - Functions
  - Data types
  - Wire protocol
  - Address
- Naturally, this contract is expressed in XML
WSDL Elements

- The five main elements (children of definitions)
  - types
    - Complex types
  - message
    - Definition of messages
  - portType
    - Combination of messages into operations (think request/response)
  - binding
    - Description of wire protocol (think SOAP)
  - service
    - Address of service (think URL)
WSDL Elements (cont.)

- Two utility elements
  - documentation
    - Human-readable information
  - import
    - Inclusion of other WSDL documents
A WSDL Example

- `<xml version="1.0" encoding="UTF-8"?>
  <definitions name="HelloService"
    targetNamespace="http://www.ecerami.com/wsd/HelloService.wsdl"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:tns="http://www.ecerami.com/wsd/HelloService.wsdl"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">`
<message name="SayHelloRequest">
    <part name="firstName" type="xsd:string"/>
</message>

<message name="SayHelloResponse">
    <part name="greeting" type="xsd:string"/>
</message>

<portType name="Hello_PortType">
    <operation name="sayHello">
        <input message="tns:SayHelloRequest"/>
        <output message="tns:SayHelloResponse"/>
    </operation>
</portType>
<binding name="Hello_Binding" type="tns:Hello_PortType">
    <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="sayHello">
        <soap:operation soapAction="sayHello"/>
        <input>
            <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                namespace="urn:examples:helloservice" use="encoded"/>
        </input>
        <output>
            <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                namespace="urn:examples:helloservice" use="encoded"/>
        </output>
    </operation></binding>
A WSDL Example (cont.)

- `<service name="Hello_Service">
  <documentation>WSDL File for HelloService</documentation>
  <port binding="tns:Hello_Binding" name="Hello_Port">
    <soap:address location="http://localhost:8080/soap/servlet/rpcrouter"/>
  </port>
</service></definitions>`
WSDL Discussion

- Who generates WSDL?
- Who generates corresponding code?
- Why do we need inclusions?
- What is missing?
  - What service is provided?
    - Think discovery
  - When should I use this service?
    - Geographic limitations, quality, cost
  - How does the service work?
RDF

- A language for representing meta-data
  - Is more structured than plain XML
  - Includes its own schema language (RDF-Schema)
- Based on a simple (but powerful) model
  - *Statements* consist of `<subject, predicate, object>` or `<resource, property, value>`
    - To express ontologies
  - Powerful combinations
    - Predicates, objects, and statements can become subjects themselves
  - Formal semantics
“There is someone, whose name is Eric Miller, whose email is em@w3.org, and whose title is Dr.”
The Corresponding XML Representation

- `<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">
  <contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:fullName>Eric Miller</contact:fullName>
    <contact:mailbox rdf:resource="mailto:em@w3.org"/>
    <contact:personalTitle>Dr.</contact:personalTitle>
  </contact:Person>
</rdf:RDF>`
But RDF Is Not Enough

- Several successive extensions
  - DAML (DARPA Agent Markup Language)
  - OIL (Ontology Inference Layer)
  - DAML+OIL
  - OWL (Web Ontology Language)
    - W3C!!!
- Common push
  - Richer way of restricting and relating classifications
Three sublanguages
- Lite, DL, Full
  - Simple classification hierarchy
  - Maximum expressiveness while also providing computational completeness and decidability
  - “It is unlikely that any reasoning software will be able to support complete reasoning for every feature of OWL Full.”

Features
- Basic definitions
  - Class, property, subclass, subproperty, domain, range
  - (In)Equality, property characteristics, type restrictions, cardinality restrictions, intersection, versioning,…
The Three Faces of XML

- Documents
  - Plain XML, DTDs
- Serialized data
  - Structured data, XML-Schema
- Metadata
  - RDF, OWL
Discussion

- Who writes descriptions?
- Who manages evolution of descriptions?
- Is there one ontology or many ontologies?
  - If there are many, how do we map between them?
- How do we use descriptions?