Computational Web Portals

Tomasz Haupt
Mississippi State University
What is a portal?

There is something going on behind the scene!
Synopsis

• URL
• TCP/IP
• SSL
• HTTP
• HTTPS
• PKI
• Kerberos
• HTML
• XHTML
• XML
• XSLT
• SOAP
• WSDL
• UDDI
• UML
• XMI
• WSIL
• WSFL
• J2EE
• CGI
• JSP
• EJB
• DCOM
• .NET
• Active-X
• Xlink
• Xpath
• URI
• CORBA
• RMI
• JXTA
• JINI
• JMS
• peer-to-peer
• SQL
• JDBC
• ODBC
• OGSA
• GRAM
• MDS
• GARA
• RSL
• Globus
• myProxy
• JetSpeed
• Tomcat
• X.509 certificate
• JavaScript
• Java
• C#
• JWS
• JCE
• JAAS
• IDL
• RSL
• IIOP
What is a portal?

• Cell phone is an example of a portal to telephony network
• Main features:
  – easy to operate, hides complexity
  – platform independent
  – provide access to variety of services
How does it work?

Portal (client, front end)

Middleware

Network resources & services
Web Portal

Portal (client, front end)

Middleware

Network resources & services

- Web Browser
- Web Portal
- Web server
- Web server
- Web server
- HTML documents
- Images, movies, sounds
- Database
- Third party services NASDAQ, Credit Cards
- Instruments
- HTML documents
Server Side Processing

- **Web Browser**
  - x.html y.jsp
  - /cgi-bin/service

- **HTTP server**
  - request
  - response

- **Gateway**
  - Dynamic contents
    - CGI, servlets
    - Java Server Pages (JSP)
    - .NET

- **Application server (J2EE)**
  - CORBA, RMI, DCOM

- **Database**
  - Third party services
    - NASDAQ, Credit Cards

- **HTML documents**
  - Images, movies, sounds
Client Side Processing

- Web Browser
  - JavaScript
  - Java Applets
  - Active X

- HTTP server
  - HTML documents
  - Images, movies, sounds

- Gateway
  - Dynamic contents
    - CGI, servlets
    - Java Server Pages (JSP)
    - .NET

- Application server (J2EE)
  - CORBA, RMI, DCOM

- Database
  - Third party services
    - NASDAQ, Credit Cards

Scalable, but
- security
- compatibility
- data
• Early HTML is very simple (and ugly!)
  – easy to learn
  – easy to interpret (browsers)
  – forgiving browsers (who cares standards?)
  – contaminated by junk generated during the browser war
  – mixes contents with presentation
  – has (almost) no logical structure which makes search difficult

• The last standard (4.01) fixes some problems by introducing style sheets (CSS)

• XHTML - transition to the future: HTML specification conforming to XML
XML

- XML is essentially a structured text document with no presentation. This makes intelligent searches easy.
- Can be made compatible with SQL (databases). This allows for error checking.
- You need to apply a style sheet (CSS or XSL) to render. This complicates browser implementation.
- Was expected to supersede HTML, but it will not happen anytime soon, if ever.
Processing XML

Internet Explorer

response
XML + style sheet

Web Server

Netscape

response
HTML

Web Server

Gateway

HTML

XSLT

XML

XSL
Why so much hype about XML?

XML is good!
- search
- process
- exchange
XSLT is good!
- transform to whatever you need!
Metadata

• XML provide a mechanism to markup the logical structure of a text document.

• What about binary files? (say, images)
  – create a description file (of course, in XML).
  – This will make your “photo album” searchable and exchangeable
In distributed computing, you need to define a protocol ("a language") for exchanging messages.
Examples: CORBA (IIOP), EJB (RMI), DCOM. Those are "proprietary" solutions. Typically, they do not penetrate firewalls.
Let’s use http to send XML messages describing what the server should do?

This is essentially SOAP (Simple Object Access Protocol)

---

**SOAP**

- SOAP Client
- Web Server
- SOAP processor
- Application Server

Can be a web page
Can be programmatic!
WSDL
Web Service Description Language

• How do we know what the server can do for us?
• It publishes the service interface
  – e.g.: double price = getQuote(string:symbol)
    integer volume = getVolume(string:symbol)
  – actually, it is encoded, surprise!!!, in XML

Two universes exchanging ASCII over HTTP!
Does not matter: Microsoft, IBM, Mac, Sun, hobbit Frodo.
Does not matter: Cobol, C, C++, C#, C%?--
Data types are standardized by XML, though
Grid Computing

• Aim to provide ubiquitous computing by connecting distributed resources shared by multiple sites and provide “virtualization” of compute resources
• Address issues related with heterogeneity and interoperability of compute resources, security, site autonomy, resource management, job scheduling, monitoring, and so on
• Grid services provide system level services with no application level support, thus there is no direct support for application developers to exploit the benefits of Grid computing and hide it’s complexity
• Several efforts under review by the Grid Computing Environments working group at the Grid Forum
Enterprise Computational Services (ECS)

Grid Portal

middleware

- metadata repository
- task composition
- scripting tools
- user space
- task repository
- advance scheduling
- resource broker
- workflow manager
- job table
- RSL and script generator
- cron
- logging
- security
- status
- resource allocation
- file transfer
- access to remote file systems
- access to data servers and databases

The Grid
Distributed Simulation System

- Distributed Simulation System
- Application Meta-data Repository
- Dataset Meta-data Repository
- User Workspace
  - Compose and Configure
  - Shared Tasks List
  - Configured Tasks List
  - Save
  - Import/Export
- Job Submission And Monitoring
  - Job Status
  - Third-party File Transfer

- GUI
- Developer
- Analyst
- Operator
- Developer
- Analyst
- Operator
- Developer
- Analyst
- Operator
- Customer
Computational Web Portal

Web Browser

web server

grid server

web server

HTML documents

Images, movies, sounds

database

Third party services

NASDAQ, Credit Cards
Computational Web Portal

Graphical User Interface

https

Web Server integrated with EJB container

Java Server Pages

Java Beans: EJB Clients

EJB container

- metadata repository
- task composition
- scripting tools
- user space
- task repository
- advance scheduling
- resource broker
- workflow manager
- job table
- RSL and script generator
- cron
- logging
- security
- status
- resource allocation
- file transfer
- access to remote file systems
- access to data servers and databases

GSI

Computational Resources
Task: GLAB

Task Description
Sample run for Greater L.A. Basin

This task contains the following applications:
- Run-1 (Ground Motion)
- Run-2 (Structure Response)
- Run-3 (Viz Package)

Select action on Application:
- Show script
- Reconfigure application
- Remove application
- Submit batch job
- Run interactively
- Show status
- Show stdout

Select action on Task:
- Task Log
- Submit Task
- Show Jobs
<table>
<thead>
<tr>
<th>job Id</th>
<th>submitted on</th>
<th>completed on</th>
<th>status</th>
<th>project</th>
<th>task</th>
</tr>
</thead>
<tbody>
<tr>
<td>10033</td>
<td>2002-03-11 12:03:04</td>
<td>2002-03-11 12:03:06</td>
<td>DONE</td>
<td>Project-1</td>
<td>Task-1</td>
</tr>
<tr>
<td>10044</td>
<td>2002-03-12 11:42:56</td>
<td>2002-03-12 11:42:57</td>
<td>DONE</td>
<td>Project-2</td>
<td>Task-1</td>
</tr>
<tr>
<td>10046</td>
<td>2002-03-12 11:54:54</td>
<td>2002-03-12 11:55:15</td>
<td>DONE</td>
<td>Project-2</td>
<td>Task-1</td>
</tr>
<tr>
<td>10062</td>
<td>2002-02-28 16:26:16</td>
<td>2002-02-28 16:26:16</td>
<td>FAILED</td>
<td>Project-1</td>
<td>Task-1</td>
</tr>
<tr>
<td>10063</td>
<td>2002-02-28 16:26:37</td>
<td>2002-02-28 16:26:37</td>
<td>FAILED</td>
<td>Project-1</td>
<td>Task-1</td>
</tr>
<tr>
<td>10064</td>
<td>2002-02-28 16:29:57</td>
<td>2002-02-28 16:29:57</td>
<td>FAILED</td>
<td>Project-1</td>
<td>Task-1</td>
</tr>
<tr>
<td>10067</td>
<td>2002-02-28 16:32:07</td>
<td>2002-02-28 16:32:09</td>
<td>DONE</td>
<td>Project-1</td>
<td>Task-1</td>
</tr>
<tr>
<td>10069</td>
<td>2002-02-28 16:33:34</td>
<td>2002-02-28 16:33:34</td>
<td>FAILED</td>
<td>Project-1</td>
<td>Task-1</td>
</tr>
<tr>
<td>10132</td>
<td>2003-03-08 16:39:04</td>
<td>2003-03-08 16:39:05</td>
<td>DONE</td>
<td>Project-2</td>
<td>Task-1</td>
</tr>
</tbody>
</table>