Designing a Flexible Services Based Architecture for Internet Applications

Christo Angelov, Senior Consultant, CAngelov@kpmg.com
Jim August, Senior Manager, JAAugust@kpmg.com
KPMG Consulting, Incorporated

OOPSLA 2000
Minneapolis, Mn.
October 17, 2000
Agenda

- Scope of the Problem
- Architectural Goals
- Architectural Abstraction
- Current Implementation Architecture
- Context Processing Layer
- Business Logic Layer
- Services View
- Implementation Genealogy
- Lessons Learned
Scope of the Problem

- Internet/Intranet/Extranet
- IVR/Call Center
- Desktop Applications

Abstraction Layer/Context Processor

- html
- xml
- Dcom, xml Corba, etc.

XML Services Drivers

- Legacy Systems
- RDBMSs
- Existing Applications/Components
- New XML Enabled Components
Architectural Goals

- Provide consistent treatment of services across all access channels
- Be scalable, flexible, adaptable and extensible to support a changing business environment (not to mention a changing technology environment)
- Support a consistent strategic architectural vision
- Reduce the cost of technology ownership
- Leverage existing assets
- Leverage/Enhance/Buy/Build
- Support short-term tactical implementations in support of a long-term vision
Architectural Abstraction

HTML Form → Context Processor → Business Logic → Interfaces

Client Layer

Context Processing Layer

Business Logic Layer

Interface Layer

DBMS
**Current Implementation Architecture(s)**

- **Client Layer**
  - Browsers
  - Other Channels

- **Context Processing Layer**
  - Web Server
  - http, tcp/ip

- **Business Logic Layer**
  - App. Server
  - TCP/IP, IIOP
  - JDBC, TCP/IP Proprietary
  - RDBMS
  - TCP/IP, Service Drivers

- **Interface Layer**
  - Existing Systems
  - IIS
  - Apache
  - NES 4.0
  - Lotus XSL
  - Sybase Jaguar
  - WebSphere 3.5
  - NAS 4.0
  - Oracle
  - UDB 5.2
  - MQ Series
  - OS/390
  - Powerbuilder (NT)
  - HTML
  - JavaScript
  - IE 4.0+
  - NS 4.07+
## Application Service Architecture

<table>
<thead>
<tr>
<th>Application Services</th>
<th>Presentation Services</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Business Logic Services</td>
</tr>
<tr>
<td></td>
<td>Analytics</td>
</tr>
<tr>
<td></td>
<td>Application Security</td>
</tr>
<tr>
<td></td>
<td>Utilities/Helper Classes</td>
</tr>
<tr>
<td></td>
<td>Application Server</td>
</tr>
<tr>
<td></td>
<td>Database Server</td>
</tr>
<tr>
<td></td>
<td>Network Connectivity</td>
</tr>
<tr>
<td></td>
<td>Operating System</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Application Services

- **Analytics**
  - Connection/Transaction Management
  - Error Handling
  - Security/Encryption (SSL/LDAP)
  - State/Session Management
  - XML Meta-Data Services
  - XML Service Drivers

- **Runtime/Deployment Infrastructure Services**
  - Application Server
  - Database Server

- **Application Arch. Services**
  - Application Services
  - Utility Services

- **Application Utility Services**
  - Analytics
  - Application Security
  - Utilities/Helper Classes

- **Operating System**
## Implementation Genealogy

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Context Processing Functions</th>
<th>Implementation</th>
<th>Messaging</th>
<th>Application Server Components</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Function Routing XML/XSL Merge</td>
<td>MS VB DLL</td>
<td>XML</td>
<td>Business Only</td>
<td>COM VB Components</td>
</tr>
<tr>
<td>2</td>
<td>Function Routing Name/value pair conversion</td>
<td>MS VB DLL, Java Script</td>
<td>Native</td>
<td>Business Only</td>
<td>Java Servlets</td>
</tr>
<tr>
<td>3</td>
<td>Function Routing XML/XSL Merge</td>
<td>MS VB DLL</td>
<td>XML</td>
<td>Business Only</td>
<td>Java Servlets</td>
</tr>
<tr>
<td>4</td>
<td>Function Routing XML/XSL Merge</td>
<td>Java Servlet</td>
<td>XML</td>
<td>Business Only</td>
<td>Java Servlets</td>
</tr>
<tr>
<td>5</td>
<td>Function Routing</td>
<td>Java Servlet</td>
<td>XML</td>
<td>XML/XSL Merge</td>
<td>Java Servlets</td>
</tr>
<tr>
<td>6</td>
<td>Function Routing</td>
<td>Java Servlet</td>
<td>XML</td>
<td>XML/XSL Merge</td>
<td>EJBs</td>
</tr>
</tbody>
</table>
Lessons Learned

- Provide consistent treatment of services across all access channels
  - 2 week implementation of IVR from browser based channel
- Be scalable, flexible, adaptable and extensible to support a changing business environment
  - Environments supported, stress test results
- Support a consistent strategic architectural vision
- Reduce the cost of technology ownership
- Leverage existing assets
  - Legacy systems supported
- Leverage/Enhance/Build/Buy
  - Integrate with Vignette, Oblix, BowStreet, e.Piphany, Chordiant
- Support short-term tactical implementations in support of a long-term vision
  - MyBenefits vs. Hyatt Enrollment
- Other “gotchas”
  - Develop process for communicating XML contracts up front
  - Object model to data model to code
  - XML is preferred when the component clearly will be exposed/re-purposed
  - Scalability is dependent upon application design as well as infrastructure tools chosen (i.e., WebSphere)
  - HTTP get vs. http Post