Integration of Text-based Applications into Service-Oriented Architectures for Transnational Digital Government

Andréa Matsunaga, Maurí cio Tsugawa and José A. B. Fortes
University of Florida

dg.o 2007 – May, 2007

Introduction: TDG project

- NSF-sponsored Transnational Digital Government (TDG) project goal:
  - An IT system for transnational collaboration on border control

- Research goal:
  - To advance the state-of-the-art of: spoken dialogue systems, machine translation, collaborative information management, internet portals and services, network performance optimization, and software requirements in the context of a specific TDG process.
Introduction: TDG project

Dominican Republic

Belize

Internet (Belize/Dominican Republic/OAS/USA)

Translation Service

Query Processor Service

SOA Web Services (WS)

Introduction: Text-based applications
WS Development and Deployment

Integrated Development Environment
- BEA Workshop, Eclipse, IBM, WebSphere Studio, NetBeans, Sun Java Studio, Visual Studio

Application

Web Service

Web Service Engine
- Axis, gSOAP, JBossWS, PHP SOAP, SOAPLite, Sun JWSDP, WCF

Web Server
- Apache, BEA WebLogic, JBoss, IBM WebSphere Application Server, IIS, Tomcat

Platform
- Java, .NET, PHP, Perl, Linux, Windows

Web Service Tools

Challenges Enabling Applications as Services

- Existing (legacy) applications are not WS-ready
- Modifying applications can be very expensive or in many cases impossible (source code unavailable)
- Application experts do not have enough WS skills
- System integration expert do not have enough information about applications
- Solution: automate the WS-enabling process of text-based applications in a simple and efficient manner

Application experts do not have enough WS skills

System Integration expert do not have enough systems laboratory
CLAWS

- Command-Line Application Wrapper Service
- Wraps text-oriented applications into WS
- Support for interactive applications
- Mechanisms for authentication and authorization
- Platform independence
**CLAWS Input**
- Service name
- Operation(s)
  - Command-line
  - Environment variables
  - Current working directory
  - Standard input
  - Mode: non-interactive, interactive stateless, interactive stateful
    - Interaction end-detector pattern
  - Rewrite rules
  - Security configuration: none, user-based, role-based

**TDG Applications**

<table>
<thead>
<tr>
<th>CS Conversation System</th>
<th>DQP Distributed Query Processor</th>
<th>M1 Machine Translator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generates SQL-like queries from natural language</td>
<td>Generates SQL queries to local databases from a SQL-like query with neutral names</td>
<td>Translates text from one spoken language to another</td>
</tr>
<tr>
<td>Maintains context of the conversation (stateful)</td>
<td>Java code accessible as WS for Windows or Linux</td>
<td>English to Spanish and Spanish to English</td>
</tr>
<tr>
<td>C code for Windows</td>
<td></td>
<td>Batch or interactive and stateless</td>
</tr>
<tr>
<td><img src="windows.png" alt="Windows" /></td>
<td><img src="windows.png" alt="Windows" /></td>
<td><img src="linux.png" alt="Linux" /></td>
</tr>
</tbody>
</table>
Related work

- Grid computing
  - SoapLab
  - GridLab – Grid Application Toolkit (GAT)
  - Virtual Application Services (VAS)
  - Generic Application Service (GAP)
  - Generic Application Factory (GFac)

Conclusions

- Savings in development and deployment time
- Researches can concentrate effort in their area of expertise
- Eases maintenance
- TDG project officially and successfully ended in April 2007. All software were handed to the agencies.
- CLAWS was essential in the final stages of the TDG system development, as components interface had to change to incorporate new features. CLAWS avoided system crashes by rapidly WS-enabling changed applications.