Can You SPEL BPEL?

Applying Business Process Tools to Satellite Operations
Spacecraft Operational Procedures

• Most command & telemetry (C&T) systems provide some form of scripting to automate both spacecraft operations and ground system configuration.
• There are over a dozen languages to choose from (ATLAS, CCL, CECIL, CSTOL, PLUTO, STOL, TCL, etc.) and additional variation in individual installations due to ground equipment specific directives.
• Translation and validation of operations procedures can represent a major cost in acquiring a new spacecraft or changing/updating a ground system for an existing spacecraft.
**WS-BPEL**

- Web Services Business Process Execution Language (WS-BPEL)
  - An orchestration language for web services that allows creation of higher level services from services defined in Web Services Description Language (WSDL)
  - XML-Based, published by OASIS in 2007 as WS-BPEL v2.0
  - Structured statements (if, while, for)
  - Parallel operations
  - Waits, event handling, fault recovery
**WS-BPEL is well-supported**

- Execution engines & development tools:
  - ActiveBPEL®
  - NetBeans®
  - Oracle / BEA
  - IBM Websphere®
  - Eclipse WebTools

- Features:
  - Simulation
  - Debugging
  - Visual Execution

- Model-Driven Architecture:
  - Rational Rose®
  - MagicDraw®
XTCE – an OMG & CCSDS specification

- The XML Telemetry and Command Exchange (XTCE) specification provides a way to define command and telemetry formats and processing for C&T systems.
XML Stylesheet Language (XSL)

- An XTCE spacecraft document could be translated via a standard XSL style sheet into a spacecraft-specific Web Services Definition Language (WSDL) document.
- A published style sheet would provide consistent definition of service interfaces for spacecraft control given the same XTCE document.
Spacecraft Process Execution Language (SPEL)

- A WS-BPEL/SPEL tool uses the WSDL to edit operations procedures and insure valid commands and parameters are used in the procedure.
- SPEL execution results in service requests to the C&T system to request telemetry and transmit commands.
Tool Support for WSDL is Good

- Development tools provide code generation to help implement the WSDL service interfaces for the C&T system.
### Performance

- Latency added by Web Services is more variable than with local scripting and about an order of magnitude slower.
- What we found:

<table>
<thead>
<tr>
<th>Measurement</th>
<th>WS-BPEL (Glassfish)</th>
<th>Python</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command Latency</td>
<td>18 ms</td>
<td>3 ms</td>
</tr>
<tr>
<td>Telemetry Value Request</td>
<td>13 ms</td>
<td>&lt;1 ms</td>
</tr>
<tr>
<td>Command Rate With Receipt Verification</td>
<td>108 cpm</td>
<td>118 cpm</td>
</tr>
<tr>
<td>Command Rate Open Loop</td>
<td>2571 cpm</td>
<td>16364 cpm</td>
</tr>
<tr>
<td></td>
<td>43 cps</td>
<td>273 cps</td>
</tr>
</tbody>
</table>
Summary

- Latency and execution time for WS-BPEL procedures are reasonable as a human-in-the-loop view or human-in-the-loop replacement.
  - Sequential command transmission is not significant compared to a typical command counter telemetry response.
- Translation to and from native scripting languages to support higher performance is possible.
- Tools are drag-and-drop, but good ease of use would require some domain-specific enhancements. Building command requests in XML in the NetBeans BPEL-mapper is awkward compared to a language sensitive editor for text scripts.
- There may be some database/script versioning experience in space that is applicable to WS-BPEL.
References

- “Goals for the BPEL4WS Specification”, http://xml.coverpages.org/
- WS-BPEL v 2.0 http://www.oasis-open.org/
- WSDL Specification, http://www.w3.org/TR/wsd1
- XSL Specification, http://www.w3.org/TR/xsl/