An Analysis of Reliable Messaging specifications for Grid and Web Services

Shrideep Pallickara, Geoffrey Fox and Sangmi Lee Pallickara

Community Grids Lab
Indiana University
Web Services?

“A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.” – W3C working group note 11 Feb. 2004

Features different from other traditional distributed systems (e.g. CORBA, RMI..) :

- Interface defined with XML based WSDL.
- Wire format of messages is in XML-based SOAP.
- Far greater proliferation of tools and utilities.
Why reliable messaging?

- TCP is not enough
  - Provides per-hop guarantees.
  - No scheme to recover from failures.

- Applications can sometimes require strong reliability guarantees.
  - Transactions need to be assured to be processed exactly-once.
  - May delegate complexity of guarantees to the protocol implementation.
Reliable Messaging: General Requirements

- Must be transport independent.
  - Underlying link may be lossy, may possibly garble order and generate multiple copies of same message.

- Support a variety of delivery assurances, each of which may be leveraged by different applications.

- Must support recovery from failures.

- Must lend itself to incremental addition of capabilities such as security and notifications.
Application layer reliability

- XML-based collaborative application: RosettaNet/BizTalk framework; provides business-level reliability scheme.

- ebXML with its ebXML Message Service (ebMS): First reliability scheme binding with XML messages and the antecedent of the WS-Reliability specification.
  - Typical exchange involves the ebMS Message Service Handler (MSH) responding to a message with an Acknowledgement message

- WS-Reliability from Fujitsu and SUN extended and modified the basic ebMS scheme for Web services.

- WS-ReliableMessaging from IBM, and Microsoft, provides reliable messaging architecture within the Web services domain.
A note on Acknowledgements

- Sender initiated protocols: ACKs (+ve acknowledgements)
  - Confirms the receipt of a specific event
  - Sender identifies holes in the delivery sequences, based on Acks, and initiate retransmissions to remedy this error.

- Receiver-initiated protocols: NAKs (-ve acknowledgements)
  - Detect the error in the received sequences and send the negative acknowledgements to plug these gaps in the delivered sequences

- ACK-based scheme can exist by themselves, but NAK-based scheme cannot.
  - A NAK-only scheme will require a source to keep messages forever, since there is no way to know if the message was received.
General Reliable Messaging Assurances

- At-Least-Once: Guaranteed message delivery
- At-Most-Once: Guaranteed message duplicate elimination
- Exactly-Once: Guaranteed message delivery and duplicate elimination
- Guaranteed Message Ordering within a group of the messages
WS-Reliable Messaging

- Specification from IBM, and Microsoft
- Leverages the WS-Addressing and WS-Policy specifications.
- Provides support for both positive and negative acknowledgements.
- Provides operations for explicit creation and termination of sequences.
- Delivery assurance modes supported include at-least-once, at-most-once, exactly-once, and ordered delivery.
WS-Reliability

- Specification from Fujitsu, Oracle, and Sun
- Provides support for positive acknowledgements.
- Provides support for a variety of message-exchange patterns.
  - Request/Response, One-way, Polling
- Delivery assurance mode supported
  - Unreliable, at-least-once, ordered-and-exactly-once
- Is currently an OASIS standard.
  - (Note WS-Security is also an OASIS standard)
WSRM & WSR Similarities

- Messages are part of a sequence/group of messages.
- Addresses issues pertaining to ordering and duplicate detection.
- Quality of service constraints are applied to groups of messages.
- Recommends message-level security, specifically WS-Security, for secure delivery of messages.
- Provides framework for reporting faults/errors in processing between endpoints involved in reliable delivery.
- Provide support for acknowledging multiple range of messages received within a group/sequence.
WSRM/WSR Differences - I

- WSR has no support for negative acknowledgements. WSRM supports negative acknowledgements.
  - Error correction can be initiated at the sender side in WSRM.

- In WSR application faults are mixed with protocol faults.
  - Acknowledgements in WSR also include fault reporting. WSRM does not do so. Also, WSRM does not concern itself with application faults.

- WSRM has an explicit operation for the creation of sequence and associated sequence identifier. WSR has no such operation, a new group begins when a receiver has encountered a new groupId.
  - disadvantage: difficult to resolve collisions in group id namespace

- WSRM message numbering begins at 1. In WSR it starts at 0.

- WSR supports multiple message exchange patterns (Response, Callback and Poll). Acknowledgements can cover not just multiple messages in a group, but also multiple groups of messages.
WSRM/WSR Differences - II

- WSRM uses WS-Addressing while WSR doesn’t specifically mandate its use.
  - WS-Addressing has sophisticated rules for EPR management and fault reporting.
- Order is always tied to duplication elimination in WSR. WSRM allows order and duplication detection to exist independent of each other.
- WSR incorporates a HTTP binding for its specification. WSRM currently does not have one; though one can simply use SOAP’s HTTP binding.
- WSRM has an explicit exchange for the termination of sequences. WSR groups cannot be terminated. They are first closed and then removed.
- WSRM uses WS-Policy for specifying and exchanging featured info. WSR does not advocate any specification though it alludes to an abstract concept of agreements.

http://www.naradabrokering.org
ITCC 2005 Track on Modern Web and Grid Systems
Federation, Why?

- WSRM being supported by powerful industry consortium led by IBM/Microsoft.
- WSR is an OASIS standard.
- It is quite possible that these specifications will continue to co-exist for a while.
- Federation would allow end-points belonging to different specifications to communicate with each other.
Federation, How?

- Mapping of physical (XML) elements and semantics associated with these specifications.
  - Mapping of sequence numbering. WSRM starts at 1, WSR starts at 0.
  - NAKs in WSRM messages will simply be ignored, since WSR does not support it.

- Mapping of policy elements and rules associated with where/when and the combination in which multiple policy/agreement elements may occur.

- Enforcing constraints on delivery. WSR provides a subset of the delivery modes available in WSRM

- Mapping of faults/error reporting

- Creation/Termination of sequence in WSRM have no equivalents in WSR.
  - So terminate-sequence in WSRM will trigger multiple requests/retransmission to ensure WSR has received everything. Group expiry time then needs to be updated at the WSR side.
Implementations of WSRM & WSR

- NaradaBrokering has released support for WSRM to the OMII, UK. This software will be available for General Open source release at http://www.naradabrokering.org on April 5th 2005.

- WSR has been implemented. We anticipate release in mid-May 2005.

- Other note, support for WS-Eventing (April 5th, 2005) currently available.
<table>
<thead>
<tr>
<th>Operation</th>
<th>Mean</th>
<th>Std Dev</th>
<th>Std Error</th>
<th>Outlier</th>
<th>Min</th>
<th>Max</th>
<th>Mem (Bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create an XMLBeans based Envelope Document</td>
<td>121.29</td>
<td>25.77</td>
<td>2.65</td>
<td>6</td>
<td>110</td>
<td>333</td>
<td>2192</td>
</tr>
<tr>
<td>Create an Axis based SOAPMessage</td>
<td>85.76</td>
<td>79.36</td>
<td>8.22</td>
<td>7</td>
<td>34</td>
<td>540</td>
<td>1824</td>
</tr>
<tr>
<td>Convert an EnvelopeDocument to a SOAPMessage</td>
<td>3503.8</td>
<td>758.48</td>
<td>80.85</td>
<td>12</td>
<td>2632</td>
<td>5406</td>
<td>57152</td>
</tr>
<tr>
<td>Convert SOAPMessage to EnvelopeDocument</td>
<td>730.08</td>
<td>392.35</td>
<td>41.58</td>
<td>11</td>
<td>327</td>
<td>1911</td>
<td>34424</td>
</tr>
<tr>
<td>Create a WS-Addressing EPR (Contains just a URL address)</td>
<td>84.61</td>
<td>25.61</td>
<td>2.67</td>
<td>8</td>
<td>72</td>
<td>301</td>
<td>2072</td>
</tr>
<tr>
<td>Create a WS-Addressing EPR (Contains WSA ReferenceProperties)</td>
<td>133.13</td>
<td>35.64</td>
<td>3.71</td>
<td>8</td>
<td>114</td>
<td>354</td>
<td>2648</td>
</tr>
<tr>
<td>Create an Envelope targeted to a specific WSA EPR</td>
<td>157.98</td>
<td>12.19</td>
<td>1.27</td>
<td>8</td>
<td>140</td>
<td>219</td>
<td>7184</td>
</tr>
<tr>
<td>Create an Envelope targeted to a specific WSA EPR with most WSA message information headers</td>
<td>263.20</td>
<td>35.73</td>
<td>3.74</td>
<td>9</td>
<td>240</td>
<td>471</td>
<td>13880</td>
</tr>
<tr>
<td>Operation</td>
<td>Mean</td>
<td>Std Dev</td>
<td>Std Error</td>
<td>Outlier</td>
<td>Min</td>
<td>Max</td>
<td>Mem (Bytes)</td>
</tr>
<tr>
<td>----------------------------------------------------------------</td>
<td>-------</td>
<td>---------</td>
<td>-----------</td>
<td>---------</td>
<td>------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>Parse an EnvelopeDocument to retrieve WSA Headers</td>
<td>711.74</td>
<td>231.61</td>
<td>23.76</td>
<td>5</td>
<td>555</td>
<td>1317</td>
<td>61024</td>
</tr>
<tr>
<td>Create a Wsrm Fault</td>
<td>413.80</td>
<td>239.17</td>
<td>25.07</td>
<td>9</td>
<td>271</td>
<td>1212</td>
<td>18096</td>
</tr>
<tr>
<td>Create a Wsrm SequenceRequest</td>
<td>268.95</td>
<td>37.93</td>
<td>3.97</td>
<td>9</td>
<td>212</td>
<td>374</td>
<td>16392</td>
</tr>
<tr>
<td>Create a Wsrm SequenceResponse</td>
<td>234.97</td>
<td>17.40</td>
<td>1.81</td>
<td>8</td>
<td>212</td>
<td>324</td>
<td>18160</td>
</tr>
<tr>
<td>Create a Wsrm SequenceDocument</td>
<td>43.812</td>
<td>2.99</td>
<td>0.30</td>
<td>4</td>
<td>42</td>
<td>53</td>
<td>2424</td>
</tr>
<tr>
<td>Add a WsrmSequenceDocument to an existing envelope. (Contains sequence identifier and message number)</td>
<td>13.01</td>
<td>0.57</td>
<td>0.05</td>
<td>4</td>
<td>11</td>
<td>15</td>
<td>464</td>
</tr>
<tr>
<td>Create a WSRM SequenceAcknowledgement based on a set of message numbers</td>
<td>461.17</td>
<td>172.40</td>
<td>18.27</td>
<td>11</td>
<td>301</td>
<td>1043</td>
<td>20624</td>
</tr>
<tr>
<td>Create a WSRM TerminateSequence</td>
<td>20.95</td>
<td>1.30</td>
<td>0.13</td>
<td>4</td>
<td>20</td>
<td>25</td>
<td>2072</td>
</tr>
</tbody>
</table>
Conclusions

- Need for reliable messaging in Web Services.
- Analysis of WSRM and WSR.
  - Similarities and Differences
- Preliminary results from implementation.