



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.

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.

Operation	Mean	Std Dev	Std Error	Outlier	Min	Max	Mem (Bytes)
Create an XMLBeans based Envelope Document	121.29	25.77	2.65	6	110	333	2192
Create an Axis based SOAPMessage	85.76	79.36	8.22	7	34	540	1824
Convert an EnvelopeDocument to a SOAPMessage	3503.8	758.48	80.85	12	2632	5406	57152
Convert SOAPMessage to EnvelopeDocument	730.08	392.35	41.58	11	327	1911	34424
Create a WS-Addressing EPR (Contains just a URL address)	84.61	25.61	2.67	8	72	301	2072
Create a WS-Addressing EPR (Contains WSA ReferenceProperties)	133.13	35.64	3.71	8	114	354	2648
Create an Envelope targeted to a specific WSA EPR	157.98	12.19	1.27	8	140	219	7184
Create an Envelope targeted to a specific WSA EPR with most WSA message information headers	263.20	35.73	3.74	9	240	471	13880

Operation	Mean	Std Dev	Std Error	Outlier	Min	Max	Mem (Bytes)
Parse an EnvelopeDocument to retrieve WSA Headers	711.74	231.61	23.76	5	555	1317	61024
Create a Wsrm Fault	413.80	239.17	25.07	9	271	1212	18096
Create a Wsrm SequenceRequest	268.95	37.93	3.97	9	212	374	16392
Create a Wsrm SequenceResponse	234.97	17.40	1.81	8	212	324	18160
Create a Wsrm SequenceDocument	43.812	2.99	0.30	4	42	53	2424
Add a WsrmSequenceDocument to an existing envelope. (Contains sequence identifier and message number)	13.01	0.57	0.05	4	11	15	464
Create a WSRM SequenceAcknowledgement based on a set of message numbers	461.17	172.40	18.27	11	301	1043	20624
Create a WSRM TerminateSequence	20.95	1.30	0.13	4	20	25	2072

Conclusions

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