Web Services Reliability Patterns

Ingrid Buckley[1], Eduardo B. Fernandez[1], Gustavo Rossi[2] and Masoud Sadjadi[3]
[1] Florida Atlantic University, Boca Raton, Fl, 33431, USA, {ibuckley,ed}@cse.fau.edu
[2] Universidad Nacional de La Plata, Argentina, La Plata, gustavo@lifia.info.unlp.edu.ar
[3] Florida International University, Miami, Fl, 33199, USA, sadjadi@cs.fiu.edu

Abstract

Due to the widespread use of web services by enterprises, the need to ensure their reliability has become crucial. There are several standards that intend to govern how web services are designed and implemented, including protocols to which they must adhere. These standards include the WS-Reliability and WS-Reliable Messaging standards that define rules for reliable messaging. We present here patterns for these standards which define how to achieve reliable messaging between entities. We compare their features and use.

Keywords: Web Services, Reliability, Patterns.

1. Introduction

Web services have become the most popular means used by enterprises to offer services to their customers and to interoperate with business partners. These services are accessed through messages. Since messaging is crucial to the enterprise in terms of the services and transactions that are exchanged between businesses and customers, it has become essential to ensure reliable messaging. Reliable messaging, as used in this context, is the act of sending a message without duplication, ensuring guaranteed delivery as well as message ordering and message state disposition [Oas04, Oas07]. The implications of a failure in this respect can have a damaging impact on businesses that rely on the availability and reliability of the services offered to customers.

In general, standards defined by committees are rather complex and their descriptions are given in uniform detail, which together with their length, make understanding of the standards rather difficult. In particular, web services standards are expressed using XML, are relatively complex and lengthy, between 57 and 120 pages. By expressing these standards as patterns, including precise UML models, we attempt to make them more understandable and easier to compare to other standards with similar objectives.

The WS-Reliability and WS-Reliable Messaging Standards are defined by OASIS and the former has borrowed from the ebXML Message Service Specification 2.0 technology. WS-Reliability is a SOAP-based (SOAP 1.1 and SOAP 1.2 Part 1) specification that fulfills reliable messaging requirements critical to some applications of Web Services [W3c07]. The WS-Reliability standard utilizes quality of service (QOS) contracts, and uses conditions attached to the invocation of a set of operations; namely deliver, submit, respond and notify [Oas04]. To perform reliable delivery it uses the concept of Reliable Message Processor (RMP). The WS-Reliable Messaging standard provides guaranteed delivery, message ordering and duplicate elimination [Oas07]. To support interoperable web services, a SOAP binding is defined within this specification. However the protocol depends upon other web services specifications for identification of service endpoint addresses and policies [Oas07]. It is also possible to consider reliability at the higher levels, for example [Dob06].

The rest of this paper is organized as follows. Section 2 presents the WS-Reliability pattern, and section 3 presents the WS-Reliable Messaging pattern. Section 4 compares these two standards. We end with some conclusions in Section 5. We show only parts of the patterns for lack of space. A more complete report is available from the first author [Buc08].

2. WS-Reliability

2.1 Intent

WS-Reliability ensures that a notification is always sent in response to a failure, it also provides guaranteed message delivery, message ordering, and duplicate elimination whenever messages are sent from one entity to another.

2.2 Context

Institutions, business-to-business (B2B) applications, and critical infrastructure systems that need to send and receive messages in real-time.

2.3 Problem Some applications need reliable messaging in order to fulfill their business operations effectively and successfully. Many people communicate via the internet, thus creating heavy network traffic. Many companies offer services to consumers across the internet, which gives rise to bandwidth and availability problems. Enterprises are concerned about how to achieve reliable messaging given

some of the factors mentioned previously; more specifically, to ensure that messages are delivered with acknowledgment of receipt, in the order sent, and without duplication. How do we ensure that messages that are sent are delivered, acknowledged, sent in order, and without duplication? The solution to this problem is affected by the following ***forces:***

- Dissimilar internet connection speed used by both sides (receiving and sending parties) can affect how quickly messages are sent and received.
- Network traffic affects the time it takes a message to reach a recipient; this may increase the delay time for the messages and may change their order.
- The receiving or sending party may become unavailable and some or all messages may not get sent or received.
- Unordered and delayed messages can lead to problems for online transactions especially in banking systems and critical infrastructures.
- The response time to messages contributes to delay; when messages get lost or arrive to a recipient unordered, the recipient may take more time to respond, thus increasing the delay time.

2.4 Solution

Use a protocol with acknowledgement of delivery or failure, message ordering, and duplicate message elimination. This is achieved by first having an enforceable contract between the sending and receiving parties, and the use of sending and receiving reliable message processors (RMPs) that send, deliver order and eliminate duplicate messages.

The WS-Reliability standard utilizes four primary conceptual units as illustrated in Figure1. The Producer creates and submits messages to the Sending RMP. The Consumer receives messages delivered by the Receiving RMP and sends an acknowledgement. The Sending RMP submits messages to and receives acknowledgements from the Receiving RMP. The Receiving RMP is responsible for delivering messages to the consumer and receiving and sending notification from the consumer to the Sending RMP. A QoS Contract binds the agreement made between the consumer and producer. A protocol contract binds the Sending and Receiving RMP.

Structure

A **contract** (Figure 2) defines the quality of service expected between the sending and receiving RMP as well as the terms of the relationship between the **Producer** and the **Consumer.** The contract includes a specification of the expected quality of service (**QoS**), which determines the quality of messaging service to the communicating parties, and the **Features** which define the operations and rules which are expected. The **Reliable Messaging Processor (RMP)** [Oas04] handle messages that are sent between a producer and a consumer and perform messaging as outlined in the contract in the form of requirements such as guaranteed delivery, duplicate message elimination, and message ordering. The implementation of the RMP is not specified by the standard, and can be implemented in many different ways (see implementation).

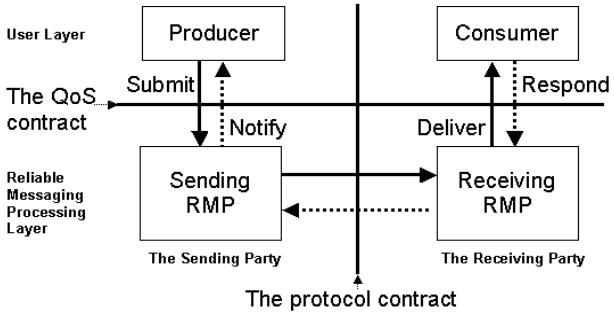

Figure 1: Structure and dataflow of the components involved in the WS-Reliability Standard [Oas04]

Processes may have two **ProcessRoles**, the **Producer** role creates messages and sends them to the **Sending RMP**. The **Consumer** role consumes messages that have been processed by the **Receiving RMP**. A **Message** can be a **Group Message** or an **Individual Message** with varying attributes depending on the type of message. The **SOAP MessageExchangePattern (MEPs)** defines different modes of response which can be sent from the Consumer to the Receiving RMP in response to a previously received message. The SOAP MEPs used is defined in SOAP 1.2 [W3c07].

Dynamics

Use cases Send a message and Establish an agreement are not shown for lack of space.

2.5 Consequences

The WS-Reliability pattern presents the following advantages:

- Messages sent between end points can be controlled by means of the RMP that ensures delivery with acknowledgment, ordering, and duplicate elimination of messages within the limits imposed by the network.
- Enterprises are able to obtain a higher degree of reliability for network communication because the sender and receiver confirm reception by an acknowledgment each time they communicate via a message.
- Quality of service defined by contracts can be maintained between businesses thus increasing reliability and supporting the accountability of business partners. Policies can be attached to the contracts that govern the modus operandi agreed by all communicating parties.

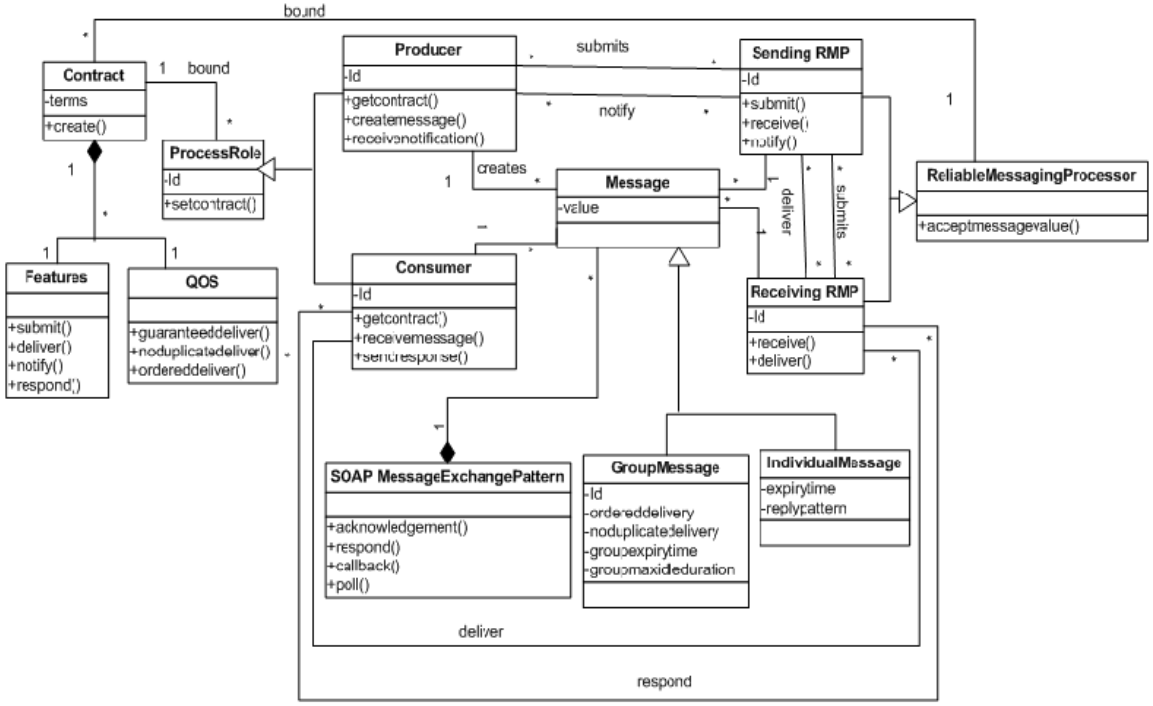

Figure 2: Class Diagram for the WS-Reliability pattern

The RMP sends an acknowledgment if the consumer becomes unavailable during the transmission of a reliable message.

The pattern also has some possible liabilities:

- The message and its response are passed between several components and not directly to the recipient or producer of the message. This process increases the time it takes the message to be delivered to the recipient and the time it takes to send the corresponding notification back to the producer of the message.
- The WS-Reliability pattern increases the complexity in the system.

3. WS-Reliable Messaging

3.1 Intent

WS-Reliable Messaging ensures guaranteed receipt in response to each message sent; it also provides, message state disposition, ordered delivery, and duplicate elimination whenever messages are sent between endpoints.

3.2 Context

Institutions, B2B applications, and critical infrastructure systems that need to send and receive messages in real-time.

3.3 Problem

Many errors can interrupt communication, messages can get lost, duplicated, or reordered; the host system may experience failures and lose volatile state and messages may also experience state loss during transmission.

Some applications need to have reliable messaging in order to fulfill their business operations effectively and successfully; therefore, lost, unordered and duplicate messages can have a negative affect on successful business operations. How do we ensure ordered delivery, guaranteed receipt, duplicate elimination and state disposition of messages? The solution to this problem is affected by the following forces:

- The receiving or sending host may become unavailable and some or all messages may not get sent or received.
- Messages may get lost during transmission.
- Unordered and delayed messages can lead to problems for online transactions especially in banking systems and critical infrastructures.
- The response time to messages contributes to delay in sending a receipt; when messages get lost or arrive to a recipient out of order, it may take more time to respond, thus increasing the response time.
- Dissimilar internet connection speed used by both sides (receiving and sending parties) can

affect how quickly messages are sent and received.
- Network traffic affects the time it takes a message to reach a recipient; this may increase the delay time for the messages and may change their order.

3.4 Solution

Use a protocol that performs guaranteed receipt, ordered delivery, state disposition, and duplicate elimination of messages. This is achieved by first having an agreement which includes a policy exchange, endpoint resolution and establishment of trust between end points.

The WS-Reliable Messaging standard utilizes four primary conceptual units as illustrated in Figure 3.

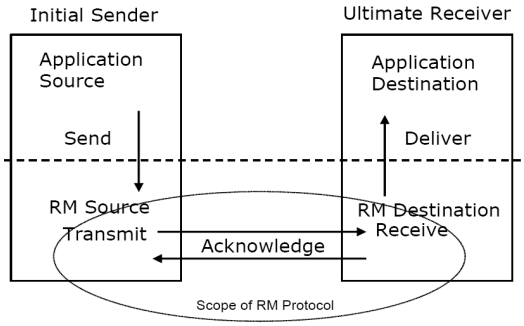

Figure 3: Structure and dataflow of the components involved in the WS-Reliable Messaging Standard [Oas07]

The application source creates and sends messages to the RM Source. The RM Source transmits messages to the RM Destination. The RM Destination receives messages transmitted from the RM source and sends a corresponding receipt of acknowledgement; the message is then delivered to the application destination/receiver.

Structure

An **Agreement** enforces policy exchange, end point resolution, and trust establishment between the **Application Source** and the **Application Destination**. The **Application Source** creates and sends messages to the RM source (Figure 4). A **Message** consists of content and information about where it is supposed to be delivered. The RM Source transforms a message into a **Reliable Message** by adding new properties to the message. A **Sequence** (created by the RM destination at the request of the RM Source) acts like an envelope in which a Reliable message is placed before it is transmitted. The **RM Source** accepts messages and acknowledgements from the Application Source and RM Destination respectively, and transmits reliable messages to the RM destination. The **RM Destination** receives messages sent from the RM Source, sends a corresponding acknowledgement of receipt to the RM Source, and delivers the reliable message to the destination application. The **Application Destination** receives reliable messages from the RM Destination.

Dynamics

Use cases Send a message and Establish an agreement describe dynamic aspects but are not shown for lack of space.

3.5 Consequences

The WS- Reliable Messaging pattern presents the following advantages:

- Enterprises are able to obtain a higher degree of reliability for network communication because endpoints create and terminate message sequences. In addition a receipt of acknowledgement is sent every time a message is sent and retransmission of messages is done for messages that were not received.
- Quality of service defined by agreements can be maintained between businesses, thus increasing reliability and supporting the accountability of business partners.
- The WS-Policy standard is used to govern policies that can be attached to the agreements that govern the operations agreed to by communicating endpoints, therefore leveraging the use of other web service standards.
- WS-Addressing is utilized to achieve endpoint referencing. This specifies the endpoint reference to where the receipt of acknowledgement is to be sent in response to a message. In this way messages cannot be intercepted easily because the destination is known prior to their transmission.
- Terminate message sequence requests are sent to the RM destination to notify when no more messages will be sent using a given sequence. Therefore the system resources attached to a sequence can be freed and used to conduct other operations.

The pattern also has some possible liabilities:

- Introduces a high time overhead with the retransmission of messages and acknowledge-ments. The RM Source will retransmit messages for which no receipt of acknowledgments was received. This could result in high volume requests flooding the RM Destination depending on the retransmission and back-off interval set.
- There is a high demand on the resources used to track the state of each message transmitted as required by the RM Source.

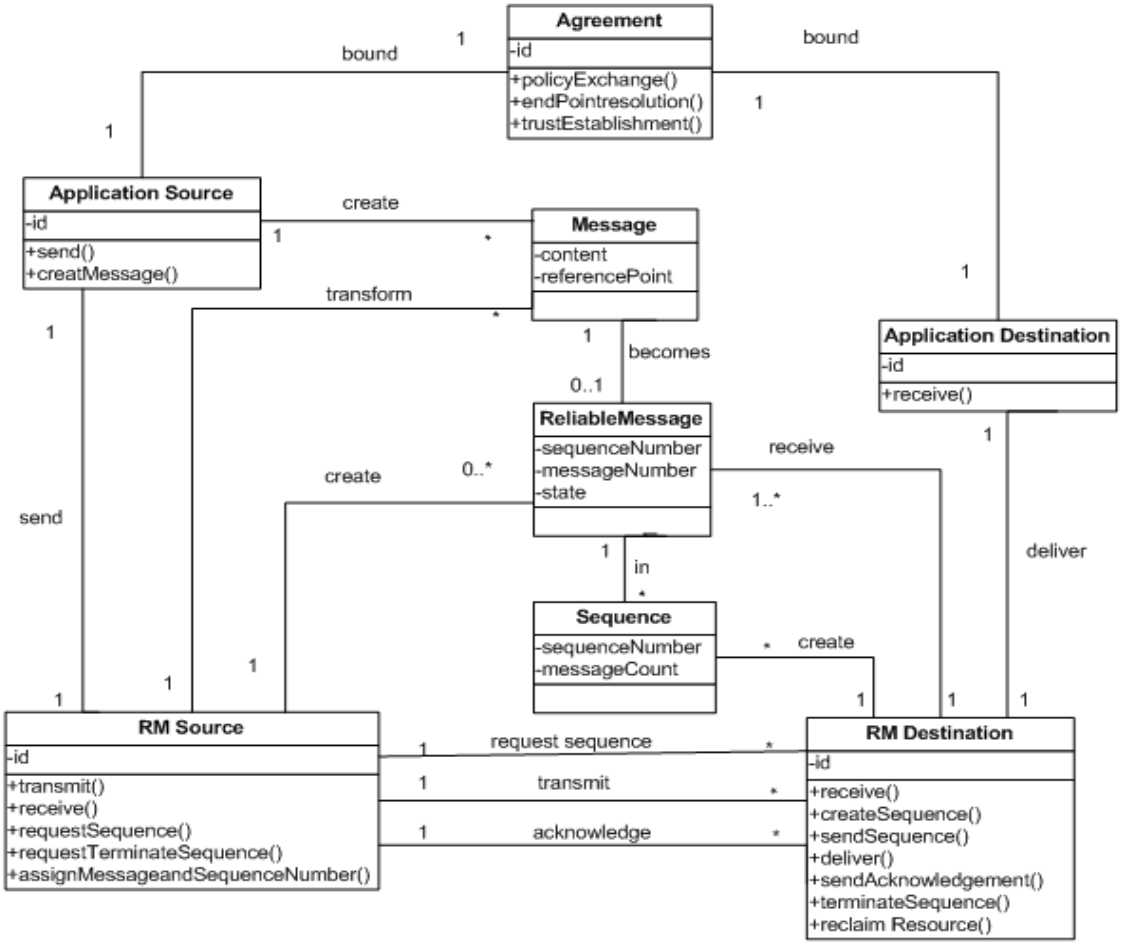

Figure 4: Class Diagram for the WS-Reliable Messaging pattern.

4. Comparing the WS-Reliability and WS-Reliable Messaging Patterns

WS-Reliability and WS-Reliable Messaging specifications offer the same basic service, which is sending messages in a reliable manner. However, the two protocols utilize different means of performing this service. WS-Reliability has a binding to HTTP whereas WS-Reliable Messaging is transport independent allowing it to be implemented using different network technologies. In order to support interoperable web services, a SOAP binding is defined within both patterns. The specifications mandate that an agreement be made before communication can be done between endpoints. However the WS-Reliable Messaging explicitly states that endpoint referencing, establishment of trust and policy exchange are to be included in the agreement. Endpoint reference explicitly states the address where a reliable message should be sent. Establishment of trust is achieved with an enforced agreement and policy exchange facilitates the updating of quality of service terms and conditions. WS-Reliability does not explicitly dictate the terms of the contract.

WS-Reliability engages the producer and consumer of a message in the entire cycle of sending a reliable message; due to the fact that WS-Reliability ensures that a guaranteed acknowledgement be sent to the producer of a reliable message. The producer specifies the mode of response that is required from the consumer and waits until an acknowledgement is received, this acknowledgement ends the cycle. In contrast, WS-Reliable Messaging ensures guaranteed receipt; the RM Source and Destination components control the execution of a reliable message between each other. Once the initial message is obtained, a guaranteed receipt is sent between these two components, not directly to the initial sender. In other words, WS-Reliable Messaging does not require that the sender listens for a guaranteed receipt, this is dealt with by the RM Source.

Additionally WS-Reliable Messaging must use a sequence to transmit all messages (individual and series), while in WS-Reliability the Sending RMP and Receiving RMP send messages either individually or in groups. Another contrast between the two specifications is that WS-Reliable Messaging mandates that all sequences be ended when no further messages will be sent using that sequence. This allows resources that are attached to each sequence to be reclaimed. WS-Reliability uses a GroupExpiryTime to terminate group messages and an ExpiryTime to terminate an individual message.

Another difference between the two specifications is that WS-Reliability uses SOAP message exchange patterns, which specify the mode of response to be used by the recipient of a reliable message. The message exchange patterns used are poll, respond and callback. However, WS-Reliable Messaging does not explicitly ask for a particular response mode from the recipient of a reliable message. In fact WS-Reliable Messaging does not require a response from the recipient of a reliable message, because the RM Destination sends a receipt of acknowledgement to the RM Source directly. Additionally WS-Reliable Messaging allows a receipt of acknowledgment to be sent with or without using the SOAP body.

In summary, WS-Reliability will only send an acknowledgement when a reliable message is delivered to the recipient; this supports real time communication using messaging. However WS-Reliable Messaging sends a receipt of acknowledgment once the RM destination receives a reliable message, which can be done before, after or simultaneously to delivering the reliable message to its destination. In the case of group messages WS-Reliable Messaging can hold messages at the RM Destination until all messages are received and send them all at once to the recipient. Therefore the concept of guaranteed acknowledgment and guaranteed receipt is different between the two specifications.

WS-Reliable Messaging is dependent on WS-Policy and WS-Addressing for policy and identification of endpoints respectively. WS-Reliability and WS-Reliable Messaging do not explicitly state how to achieve ordering and duplicate elimination of messages. Instead, they state that both features can be implemented in different ways. WS-Reliable Messaging includes message state disposition, the RM Source tracks each reliable message until a receipt is received from the RM Destination. WS-Reliability uses a message number to keep track of each message sent.

WS-Reliability requires a contract while WS-Reliability requires an agreement before communication can begin between endpoints. The patterns are similar in this regard however, the agreement includes establishment of trust, policy exchange and endpoint resolution. The structure of the key components in the WS-Reliability and WS-Reliable Message patterns are similar, see Figure 2 and 4. However the WS-Reliability pattern uses the Sending and Receiving RMP to provide acknowledgement, ordering, duplicate elimination, and guaranteed delivery of messages, while, the WS-Reliable Messaging pattern uses the RM Source and RM Destination to provide similar functions.

5. Conclusions

WS-Reliability and WS-Reliable Messaging are two standards intended to specify the reliable delivery of message between web services. We have provided patterns for these standards. The original standards are verbose and complex; we hope to have clarified their structure and behavior. Since both standards apply to the same problem we provided a comparison of their features.

Future work will include the development of further patterns so as to provide the designer with a catalog of patterns that can be used when developing web-services-based systems.

Acknowledgements

This work was supported in part by IBM, the National Science Foundation (grant OISE-0730065), and DoD's DISA through Pragmatics, Inc.

References

[Buc08] I. Buckley and E.B Fernandez, "Web Services Reliability Patterns", Department of Computer Science and Engineering, Florida Atlantic University, 2007.

[Dob06] G. Dobson, "Using WS-BPEL to implement software fault-tolerance for Web Services", Proc. 32nd EUROMICRO Conference on software, 2006.

[Gam94] E. Gamma, R. Helm, R. Johnson, J. Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, Boston, Mass., 1994.

[Oas04] OASIS, "Web Services Reliable Messaging TC WS-Reliability 1.1", http://docs.oasis-open.org/wsrm/ws-reliability/v1.1/wsrm-ws_reliability-1.1-spec-os.pdf, 2004.

[Oas07] OASIS, "Web Services Reliable Messaging (WS-Reliable Messaging) Version1.1", http://docs.oasis-open.org/ws-rx/wsrm/200702/wsrm-1.1-spec-os-01-e1.pdf, 2007.

[W3c07] W3C, "SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)" http://www.w3.org/TR/soap12-part1/, 2007