Web Services Description Language (WSDL) Version 2.0 Part 2: Message Patterns

W3C Working Draft 10 November 2003

This version:
    http://www.w3.org/TR/2003/WD-wsdl20-patterns-20031110
Latest version:
    http://www.w3.org/TR/wsdl20-patterns
Previous versions:
    http://www.w3.org/TR/2003/WD-wsdl12-patterns-20030611

Editors:
    Martin Gudgin, Microsoft
    Amy Lewis, TIBCO
    Jeffrey Schlimmer, Microsoft

This document is also available in these non-normative formats: postscript, PDF, XML, and plain text.

Abstract

This document describes Web Services Description Language (WSDL) Version 2.0 message patterns. These patterns are intended for use with the Web Services Description Language (WSDL).

Status of this Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at http://www.w3.org/TR/.

This is the first W3C Working Draft of the WSDL Version 2.0 Message Patterns specification for review by W3C members and other interested parties.

This document has been produced as part of the W3C Web Services Activity. The authors of this document are the Web Services Description Working Group members.

Publication as a Working Draft does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.

Comments on this document are invited and are to be sent to the public www-ws-desc@w3.org mailing list (public archive).

Patent disclosures relevant to this specification may be found on the Working Group's patent disclosure page.

Short Table of Contents

Table of Contents

Appendices

1. Introduction

Web Services Description Language (WSDL) message patterns define the sequence and cardinality of abstract messages listed in an operation. Message patterns also define which other nodes send messages to, and receive messages from, the service implementing the operation.

By design, WSDL message patterns abstract out specific message types. Patterns identify placeholders for messages, and placeholders are associated with specific message types by the operation using the pattern.

Unless explicitly stated otherwise, WSDL message patterns also abstract out binding-specific information like timing between messages, whether the pattern is synchronous or asynchronous, and whether the message are sent over a single or multiple channels.

Like interfaces and operations, WSDL message patterns do not exhaustively describe the set of messages exchanged between a service and other nodes; by some prior agreement, another node and/or the service may send other messages (to each other or to other nodes) that are not described by the pattern. For instance, even though a pattern may define a single message sent from a service to one other node, the Web Service may multicast that message to other nodes.

To maximize reuse, WSDL message patterns identify a minimal contract between other parties and Web Services, and contain only information that is relevant to both the Web Service and another party.

This specification defines several message patterns for use with *WSDL Version 2.0 Part 1: Core Language* [*WSDL 2.0 Core Language [p.8]*].

1.1 Notational Conventions

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [*IETF RFC 2119 [p.8]*].

2. Fault Generation Rules

WSDL patterns specify their fault generation model using standard rulesets to indicate where faults may occur. The two most common patterns for fault generation are defined here, and referenced by patterns later in the document.

Generation of a fault, regardless of ruleset, terminates the exchange.

2.1 Fault Replaces Message

Any message after the first in the pattern MAY be replaced with a fault message, which MUST have identical cardinality and direction. The fault message MUST be delivered to the same target node as the message it replaces.

2.2 Message Triggers Fault

Any message, including the first, MAY trigger a fault message in response. Each recipient MAY generate a fault message, and MUST generate no more than one fault for each triggering message. Each fault message has direction the reverse of its triggering message. The fault message MUST be delivered to the originator of the message which triggered it. If there is no path to this node, the fault MUST be discarded.

2.3 No Faults

No faults may be generated.

Editorial note: Introduction of No Faults ruleset	12 June 2003
The No Faults ruleset has been introduced primarily to clarify the confusion otherwise introduced by applying the Fault Replaces Message ruleset to single-message patterns (which implicitly disallows faults). Some concern has been expressed that a no-fault ruleset could easily be abused.	

3. Message Patterns

WSDL patterns are described in terms of the WSDL component model, specifically the Message Reference and Fault Reference components.

3.1 In-Only

This pattern consists of exactly one message as follows:

1. A message:

    - indicated by a Message Reference component whose {messageReference} is ’A’ and {direction} is ’in’
    - received from some node N

This pattern uses the rule **2.3 No Faults** [p.4] .

An operation using this message pattern has a {pattern} property with the value ’http://www.w3.org/2003/11/wsdl/in-only’.

3.2 Robust In-Only

This pattern consists of exactly one message as follows:

1. message:
    - indicated by a Message Reference component whose {messageReference} is ’A’ and {direction} is ’in’
    - received from some node N

This pattern uses the rule **2.2 Message Triggers Fault** [p.4] .

An operation using this message pattern has a {pattern} property with the value ’http://www.w3.org/2003/11/wsdl/robust-in-only’.

3.3 In-Out

This pattern consists of exactly two messages, in order, as follows:

1. A message:
    - indicated by a Message Reference component whose {messageReference} is ’A’ and {direction} is ’in’
    - received from some node N
2. A message:
    - indicated by a Message Reference component whose {messageReference} is ’B’ and {direction} is ’out’
    - sent to node N

This pattern uses the rule **2.1 Fault Replaces Message** [p.4] .

An operation using this message pattern has a {pattern} property with the value ’http://www.w3.org/2003/11/wsdl/in-out’.

3.4 In-Multi-Out

Editorial note: Review of In-Multi-Out	22 September 2003
The WG has conditionally retained this pattern and expects to review a proof-of-concept binding that exercises this pattern at the next face-to-face meeting.	

This pattern consists of one or more messages, in order, as follows:

1. A message
   - indicated by a Message Reference component whose {messageReference} is ’A’ and {direction} is ’in’
   - received from some node N
2. Zero or more messages
   - indicated by a Message Reference component whose {messageReference} is ’B’ and {direction} is ’out’
   - sent to node N

This pattern uses the rule "Fault Replaces Message".

An operation using this message pattern has a {pattern} property with the value ’http://www.w3.org/2003/11/wsdl/in-multi-out’.

3.5 Out-Only

This pattern consists of exactly one message as follows:

1. A message:
   - indicated by a Message Reference component whose {messageReference} is ’A’ and {direction} is ’out’
   - sent to some node N

This pattern uses the rule **2.3 No Faults** [p.4] .

An operation using this message pattern has a {pattern} property with the value ’http://www.w3.org/2003/11/wsdl/out-only’.

3.6 Robust Out-Only

This pattern consists of exactly one message as follows:

1. message:
   - indicated by a Message Reference component whose {messageReference} is ’A’ and {direction} is ’out’
   - sent to some node N

This pattern uses the rule **2.2 Message Triggers Fault** [p.4] .

An operation using this message pattern has a {pattern} property with the value ’http://www.w3.org/2003/11/wsdl/robust-out-only’.

3.7 Out-In

This pattern consists of exactly two messages, in order, as follows:

1. A message:
    - indicated by a Message Reference component whose {messageReference} is ’A’ and {direction} is ’out’
    - sent to some node N
2. A message:
    - indicated by a Message Reference component whose {messageReference} is ’B’ and {direction} is ’in’
    - sent from node N

This pattern uses the rule **2.1 Fault Replaces Message** [p.4] .

An operation using this message pattern has a {pattern} property with the value ’http://www.w3.org/2003/11/wsdl/out-in’.

3.8 Asynchronous Out-In

This pattern consists of one or two messages, in order, as follows:

1. A message:
    - indicated by a Message Reference component whose {messageReference} is ’A’ and {direction} is ’out’
    - sent to some node N
2. An optional message:
    - indicated by a Message Reference component whose {messageReference} is ’B’ and {direction} is ’in’
    - sent from node N

This pattern uses the rule **2.2 Message Triggers Fault** [p.4] .

An operation using this message pattern has a {pattern} property with the value ’http://www.w3.org/2003/11/wsdl/asynch-out-in’.

3.9 Out-Multi-In

Editorial note: Review of Out-Multi-In	22 September 2003
The WG has conditionally retained this pattern and expects to review a proof-of-concept binding that exercises this pattern at the next face-to-face meeting.	

This pattern consists of one or more messages, in order, as follows:

1. A message:
   - indicated by a Message Reference component whose {messageReference} is ’A’ and {direction} is ’out’
   - sent to some node N
2. Zero or more messages:
   - indicated by a Message Reference component whose {messageReference} is ’B’ and {direction} is ’in’
   - sent from node N

This pattern uses the rule "Fault Replaces Message".

An operation using this message pattern has a {pattern} property with the value ’http://www.w3.org/2003/11/wsdl/out-multi-in’.

4. References

4.1 Normative References

[IETF RFC 2119]
Key words for use in RFCs to Indicate Requirement Levels, S. Bradner, Author. Internet Engineering Task Force, June 1999. Available at http://www.ietf.org/rfc/rfc2119.txt.

[WSDL 2.0 Core Language]
Web Services Description (WSDL) Version 2.0 Part 1: Core Language, R.Chinnici, M.Gudgin, J-J. Moreau, S.Weerawarana Editors. World Wide Web Consortium, 10 November 2003. This version of the "Web Services Description Version 2.0 Part 1: Core Language" Specification is available is available at http://www.w3.org/TR/2003/WD-wsdl20-20031110. The latest version of "Web Services Description Version 2.0 Part 1: Core Language" is available at http://www.w3.org/TR/wsdl20.

4.2 Informative References

[WSD Requirements]
Web Services Description Requirements, J. Schlimmer, Editor. World Wide Web Consortium, 28 October 2002. This version of the Web Services Description Requirements document is http://www.w3.org/TR/2002/WD-ws-desc-reqs-20021028. The latest version of Web Services Description Requirements is available at http://www.w3.org/TR/ws-desc-reqs.

A. Acknowledgements (Non-Normative)

This document is the work of the W3C Web Service Description Working Group.

Members of the Working Group are (at the time of writing, and by alphabetical order): Adi Sakala (IONA Technologies), Alan Davies (SeeBeyond), Allen Brookes (Rogue Wave Softwave), Amelia Lewis (TIBCO/Extensibility), Arthur Ryman (IBM), Bijan Parsia (University of Maryland), Dale Moberg (Cyclone Commerce), Dan Kulp (IONA Technologies), David Booth (W3C), Dietmar Gaertner (Software AG), Don Mullen (TIBCO Software), Erik Ackerman (Lexmark), Glen Daniels (Macromedia), Igor Sedukhin (Computer Associates), Ingo Melzer (DaimlerChrysler Research and Technology), Jacek Kopecky (Systinet), Jean-Jacques Moreau (Canon), Jeff Mischkinsky (Oracle Corporation), Jeffrey Schlimmer (Microsoft Corporation), Jerry Thrasher (Lexmark), Jim Hendler (University of Maryland), Jonathan Marsh (*Chair*, Microsoft Corporation), Kevin Canyang Liu (SAP), Lily Liu (webMethods, Inc.), Martin Gudgin (Microsoft Corporation), Michael Champion (Software AG), Michael Mahan (Nokia), Michael Mealling (Verisign), Mike Ballantyne (Electronic Data Systems), Mike Davoren (W. W. Grainger), Mike McHugh (W. W. Grainger), Paul Downey (BT), Philippe Le Hégaret (W3C), Prasad Yendluri (webMethods, Inc.), Roberto Chinnici (Sun Microsystems), Sandeep Kumar (Cisco Systems), Sanjiva Weerawarana (IBM), Steve Graham (Global Grid Forum), Steve Tuecke (Global Grid Forum), Tom Jordahl (Macromedia), Umit Yalcinalp (Oracle Corporation), Waqar Sadiq (Electronic Data Systems), William Vambenepe (Hewlett-Packard Company), Youenn Fablet (Canon)

Previous members were: Aaron Skonnard (DevelopMentor), Jochen Ruetschlin (DaimlerChrysler Research and Technology), Keith Ballinger (Microsoft), Krishna Sankar (Cisco Systems), Mario Jeckle (DaimlerChrysler Research and Technology), Pallavi Malu (Intel Corporation), Don Wright (Lexmark), Joyce Yang (Oracle Corporation), Daniel Schutzer (Citigroup), Dave Solo (Citigroup), Stefano Pogliani (Sun Microsystems), William Stumbo (Xerox), Stephen White (SeeBeyond), Barbara Zengler (DaimlerChrysler Research and Technology), Tim Finin (University of Maryland) Laurent De Teneuille (L’Échangeur), Johan Pauhlsson (L’Échangeur), Mark Jones (AT&T), Steve Lind (AT&T), Sandra Swearingen (U.S. Department of Defense, U.S. Air Force)

The people who have contributed to discussions on www-ws-desc@w3.org are also gratefully acknowledged.

B. Change Log (Non-Normative)

B.1 Changes

Date	Author	Description
20031105	aal	Fix titles of added patterns. Move them to be in conjunction with similar patterns.
20031022	aal	Per action item from October 16 teleconference, added the three patterns using message-triggers-fault as published on the mailing list (robust-in-only, robust-out-only, asynch-out-in).
20031022	aal	Added internal linkage (using specref) from patterns to the fault rulesets which they use.
20031022	aal	Per 9 and 16 Oct 2003 teleconferences, marked in-multi-out and out-multi-in patterns deleted.
20031022	aal	Per 16 Oct 2003 teleconference, added a paragraph/sentence stating that generation of a fault terminates an exchange.
20031007	JCS	Per 2 Oct 2003 teleconference, changed "broadcast" to "multicast" in the introduction.
20030922	JCS	Per 22 Sep 2003 meeting in Palo Alto, CA, removed "Pattern Review" editorial note; added specific editorial notes for In-Multi-Out and Out-Multi-In.
20030911	RRC	Changed the "name" property of the message reference component to "messageReference".
20030904	JCS	Incorporated clarifications suggested by W3C\David Booth.
20030801	JCS	Per 30 July meeting, added recommendations from patterns task force.
20030612	AAL	Added fault generation rulesets and references to them from patterns.
20030313	MJG	Changed to Part 2 (from Part 3)
20030306	JCS	Proposed name for MEP7.
20030305	JCS	Per 4 Mar 03 meeting, renamed ’message exchange pattern’ to ’message pattern’ or ’pattern’, added pattern for request-response, added ednote about review of patterns.
20030217	MJG	Fixed some issues with entities and validity errors WRT ulists
20030212	JCS	Initial draft