WS-MediationPolicy 1.6

Latest version: 24 January 2012

Abstract

[WS-Policy] defines a framework for allowing Web services to express their policy constraints and IT operational requirements in a standardized structured document format. Such constraints and requirements are expressed as policy assertions.

This document defines a set of policy assertions for describing common service mediation requirements as it is being implemented in IBM policy enforcement products (such as WebSphere DataPower) and policy administration products (such as WebSphere Service Registry and Repository).

Status of this Document

This document is an IBM specification that has been adopted by IBM products for integration**.** Here are the document's Terms of use.

Table of Contents

1. Introduction

The adoption of SOA best practices has created an ideal environment for the introduction of several aspects of policy management inside an SOA solution. The use of policies in SOA has several benefits such as formalization, standardization, automation, reuse, management, etc.

This specification defines a set of policy assertions to be used with [*WS-Policy*] to define a Quality of Service (QoS), Routing, Message Validation, and Message Translation to control interactions between Web service entities. Such a policy can then be deployed to and enforced by SOA policy enforcement platform – such as a WebSphere DataPower appliance.

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 [RFC 2119].

This specification uses the following syntax to define outlines for messages:

- The syntax appears as an XML instance, but values in italics indicate data types instead of literal values.
- Characters are appended to elements and attributes to indicate cardinality:
  - "?" (0 or 1)
  - "*" (0 or more)
  - "+" (1 or more)
- The character "|" is used to indicate a choice between alternatives.
- The characters "(" and ")" are used to indicate that contained items are to be treated as a group with respect to cardinality or choice.
- The characters "[" and "]" are used to call out references and property names.
- Ellipses (i.e., "...") indicate points of extensibility.
- XML namespace prefixes [1.2 Namespaces] are used to indicate the namespace of the element being defined.

This specification can be used in terms of XML Information Set (Infoset) [XML Infoset], even though the specification uses XML 1.0 terminology. Valid Infoset for this specification is the one serializable in XML 1.0, hence the use of XML

1.0.

1.2. Namespaces

The XML namespace that MUST be used by implementers of this specification is:

http://www.ibm.com/xmlns/stdwip/2011/02/ws-mediation

The table below lists XML namespaces that are used in this specification. The choice of any namespace prefix is arbitrary and not semantically significant.

Prefix	XML Namespaces	Specification(s)
wsme	http://www.ibm.com/xmlns/stdwip/2011/02/ws-mediation	This specification
wsp	http://www.w3.org/ns/ws-policy	[WS-Policy]
xs	http://www.w3.org/2001/XMLSchema	[XMLSchema - Part 1]

2. Mediation Rule Assertion

Services indicate support for mediation policy through the use of the Web Services Policy - Framework [WS-Policy] and Web Services Policy - Attachment [WS-PolicyAttachment] specifications.

This specification defines a policy assertion (`wsme:Rule`). The normative outline of this assertion is:

```
<wsme:Rule ...>

  <wsme:Condition>
     ...
  </wsme:Condition>?

  <wsme:Action (IfCondition="xs:boolean")?>

     <wsme:QueueMessage/>?

    (<wsme:RejectMessage/>
     |
     <wsme:Notify/>
     |
     <wsme:RouteMessage>
        <wsme:EndPoint>...</wsme:EndPoint>
     </wsme:RouteMessage>
     |      <wsme:ValidateMessage>
        (<wsme:XSD>xs:anyURI</wsme:XSD> |
         <wsme:WSDL>xs:anyURI</wsme:WSDL>)
        <wsme:Scope>SOAPBody|SOAPBodyOrDetails|
                    SOAPEnvelope|SOAPIgnoreFaults</wsme:Scope>?
```

```
      </wsme:ValidateMessage>
      |
      <wsme:ExecuteXSL>
        <wsme:Parameter Name="xs:string"  Value="xs:string"/>*
        <wsme:Stylesheet>xs:anyURI</wsme:Stylesheet>
      </wsme:ExecuteXSL>
      |
      xs:any
     )*

  </wsme:Action>

  xs:any*

</wsme:Rule>
```

The following describes additional, normative constraints on the outline listed above:

/wsme:Rule

When present in a policy alternative this policy assertion indicates that the communications with the policy subject (for example, the endpoint) will have the rule defined by this assertion applied.

/wsme:Rule/wsme:Condition

This OPTIONAL, non-repeating element specifies the logical condition that needs to be evaluated to decide whether an action is to be performed.

If this element is not present, it is equivalent to the logical condition evaluating to `True` and actions SHALL be enforced accordingly.

Details of this element are described in the [Mediation Condition] section below.

/wsme:Rule/wsme:Action

This REQUIRED element specifies an action to perform when the `@IfCondition` logical test is met.

Further details of this element are described in the following section.

/wsme:Rule/wsme:Action@IfCondition

This OPTIONAL boolean attribute specifies whether the action is to be performed when the logical condition evaluates to `True` or `False`.

When this attribute is not present it is equivalent to the attribute being set to **"true".**

This attribute effectively allows for a classic if-then-else construct with a condition and a set of actions to be performed depending on how the condition evaluates. See [If-Then-Else example].

/wsme:Rule/xs:any

This is an extensibility point.

2.1. Mediation Action

The Mediation Action element specifies the actions to be taken. Although the syntax allows many combinations not all of them make sense and when conflicting actions are specified, such as asking for a message to be both queued and rejected, the behavior is implementation dependent.

/wsme:Rule/wsme:Action/wsme:QueueMessage

This OPTIONAL, non-repeating element specifies that messages SHALL be queued when the logical condition is met. Message processing SHALL NOT re-commence until when the logical condition is no longer met.

The queue methodology and any associated timeouts are implementation dependent.

When several actions are specified, within a single `wsme:Action` element or across several `wsme:Action` elements, `wsme:QueueMessage` SHOULD be the first action. Behavior when it is not the first `wsme:Action` element is implementation dependent.

/wsme:Rule/wsme:Action/wsme:RejectMessage

This OPTIONAL element specifies that messages SHALL be rejected when the logical condition is met. Message processing SHALL NOT re-commence until when the logical condition is no longer met.

/wsme:Rule/wsme:Action/wsme:Notify

This OPTIONAL element specifies that a notification SHALL be produced when the logical condition is met.

How the notification is performed and what information it contains is

implementation dependent and may be subject to implementation-dependent configuration. For instance, this may simply be implemented as an addition of a message entry in a system log.

/wsme:Rule/wsme:Action/wsme:RouteMessage

This OPTIONAL element specifies that messages SHALL be routed to specified endpoint destination when the logical condition is met. Messages SHALL be routed to specified endpoint until when the logical conditionis no longer met.

/wsme:Rule/wsme:Action/wsme:RouteMessage/wsme:EndPoint

This REQUIRED, non-repeating element specifies the endpoint to which messages SHALL be routed. The endpoint value supported can be an *IP address*, *hostname*, or *virtual host*; such as load balancer group.

/wsme:Rule/wsme:Action/wsme:ValidateMessage

This OPTIONAL element specifies that messages SHALL be validated against the specified grammars.

Messages SHALL be rejected when validation fails.

/wsme:Rule/wsme:Action/wsme:ValidateMessage/wsme:XSD

This OPTIONAL, non-repeating element specifies that messages SHALL be validated against the XML schema identified by the URI it contains.

/wsme:Rule/wsme:Action/wsme:ValidateMessage/wsme:WSDL

This OPTIONAL, non-repeating element specifies that messages SHALL be validated against the Web services description (WSDL) identified by the URI it contains.

/wsme:Rule/wsme:Action/wsme:ValidateMessage/wsme:Scope

This OPTIONAL, non-repeating element specifies what part of the message SHALL be validated.

The following table lists the possible values and what they mean:

Value	**Description**
SOAPBody	The contents of the SOAP Body element, without

	special processing for SOAP faults. (Default)
SOAPBodyOrDetails	The contents of the detail element for SOAP faults, and the contents of the Body otherwise.
SOAPEnvelope	The entire SOAP message, including the envelope.
SOAPIgnoreFaults	No validation if the message is a SOAP fault, the contents of the SOAP Body otherwise.

When `wsme:Scope` is not present validation is done against the SOAP Body.

/wsme:Rule/wsme:Action/wsme:ExecuteXSL

This OPTIONAL element specifies that a message translation (using XSL transform) SHALL be performed with the specified `Stylesheet` and `Parameters`.

Messages SHALL be rejected when the execution fails.

/wsme:Rule/wsme:Action/wsme:ExecuteXSL/wsme:Stylesheet

This REQUIRED, non-repeating element specifies the transform operation SHALL use the stylesheet specified by the contained URI.

The stylesheet MUST be an XSLT file. Each enforcement platform might have restrictions on the types of URI supported.

/wsme:Rule/wsme:Action/wsme:ExecuteXSL/wsme:Parameter

This OPTIONAL, repeating element specifies a stylesheet parameter to be used for the transform operation.

/wsme:Rule/wsme:Action/wsme:ExecuteXSL/wsme:Parameter@Name

This REQUIRED attribute specifies the name of the parameter.

/wsme:Rule/wsme:Action/wsme:ExecuteXSL/wsme:Parameter@Value

This REQUIRED attribute specifies the value of the parameter.

/wsme:Rule/xs:any

This is an extensibility point.

2.2. Mediation Condition

The Mediation Condition element defines the condition that triggers an action.

The outline for the Mediation Condition element is:

```
<wsme:Condition ...>

     <wsme:Expression>

       <wsme:Attribute>...</wsme:Attribute>
       <wsme:Operator>...</wsme:Operator>
       <wsme:Value>...</wsme:Value>
       <wsme:Interval>...</wsme:Interval>?
       <wsme:Limit>...</wsme:Limit>?

     </wsme:Expression> ?

     <wsme:Schedule StartDate="xs:date"? StopDate="xs:date"?>

     <wsme:Daily StartTime="xs:time" StopTime="xs:time" />?

     <wsme:WeekDays Days="Monday+Tuesday+Wednesday+Thursday+Friday+
                          Saturday+Sunday"/> ?

     </wsme:Schedule> ?

</wsme:Condition>
```

The following describes additional constraints on the outline listed above:

/wsme:Condition

This OPTIONAL element defines the condition whose evaluation is either `True` or `False`.

This element MUST contain at least one `wsme:Expression` element or one `wsme:Schedule` element.

/wsme:Condition/wsme:Expression

This OPTIONAL non repeating element specifies a Boolean expression.

The expression comprises a REQUIRED `wsme:Attribute`, `wsme:Operator`, and `wsme:Value`, plus an OPTIONAL `wsme:Interval`, and an OPTIONAL `wsme:Limit`. If the application of the `wsme:Operator` on the `wsme:Attribute` and the `wsme:Value`, plus the `wsme:Interval` and `wsme:Limit` when appropriate, evaluates to `True`, then the expression evaluates to `True`.

The `wsme:Limit` element is only used with the `HighLow` and `TokenBucket` operators. If not specified, the value of `wsme:Limit` is `0`.

If `wsme:Interval` is not specified, the default is implementation dependent. Implementations SHOULD choose a reasonable default which MAY be based on some configuration.

/wsme:Condition/wsme:Expression/wsme:Attribute

This REQUIRED element defines the `wsme:Attribute` the `wsme:`Operator is applied to.

The following table summarizes the defined attributes and their type.

Attribute	**Description and Type**
ErrorCount	The number of faults observed during this monitoring interval.
MessageCount	The number of actual messages intercepted during the monitoring interval.
InternalLatency	The internal latency (processing time) in milliseconds.
BackendLatency	The appliance-to-server latency in milliseconds.
TotalLatency	The sum of back-end and internal latency in milliseconds.

/wsme:Condition/wsme:Expression/wsme:Operator

This REQUIRED element defines the operator to apply.

The following table summarizes the available operators and their meaning:

Operator	Meaning
GreaterThan	A simple numeric algorithm that evaluates to True when the Attribute is greater than the defined Value.
LessThan	A simple numeric algorithm that evaluates to True when Attribute is less than the defined Value.
TokenBucket	A rate-based algorithm that allows bursting. The algorithm consists of a bucket with a maximum capacity of `Limit` tokens . The bucket refills at a constant rate of `Value` tokens per `Interval`, while for each unit of `Attribute` a token is removed. This algorithm evaluates to True when there are no tokens in the bucket, and evaluates to False otherwise. Here is an example to help explain the algorithm: – Assume `Limit`=100, `Value`=5, `Interval=1 second`, and the `Attribute=MessageCount`. – The bucket starts full with a maximum capacity of 100 tokens – When a message arrives the algorithm checks whether the bucket holds any tokens – If it does, the algorithm evaluates to False and one token is removed from the bucket – If it does not, the algorithm evaluates to True. – All the while, every second, the algorithm adds 5 tokens back to the bucket as room permits.
HighLow	An algorithm that evaluates to True when Attribute reaches the high threshold specified as the Value and then continues to evaluates to True until Attribute reaches the low threshold specified as the Limit.

/wsme:Condition/wsme:Expression/wsme:Value

This REQUIRED integer element defines the `wsme:Value` the `wsme:Operator` is applied to.

/wsme:Condition/wsme:Expression/wsme:Interval

This OPTIONAL element defines the time interval, used as a sliding window, to measure the `wsme:Attribute` when evaluating the expression, in `xs:duration` format.

If not specified, the interval used is implementation dependent. Implementations SHOULD choose a reasonable default which MAY be based on configured capabilities.

/wsme:Condition/wsme:Expression/wsme:Limit

This OPTIONAL integer element defines the additonal Limit argument required when `wsme:Operator` is `TokenBucket` or `HighLow`.

The unit depends on the `wsme:Operator` specified.

When `wsme:Operator` is `HighLow` this defines the low threshold while `wsme:Value` defines the high threshold. The specified threshold SHOULD be lower than that of `wsme:Value`. When not specified the default Limit is 0.

When `wsme:Operator` is `TokenBucket` this defines the maximum size of the burst, or maximum number of tokens in the bucket , while `Value` specifies the rate at which the bucket is refilled, in number of tokens per `Interval` . When not specified the default Limit is 0 and `TokenBucket` is then equivalent to a `GreaterThan` operation.

/wsme:Condition/wsme:Schedule

This OPTIONAL non-recurring element specifies a schedule defined with an optional start date and an optional stop date, an optional daily timeframe, and an optional list of weekdays.

For example, a schedule can be defined as being effective from October 1st 2011 to October 30th 2011, from 8am to 5pm on Wednesdays and Sundays.

When on the specified schedule this element evaluates to `True`.

If this element is missing, the schedule starts immediately and continues indefinitely.

wsme:Condition/wsme:Schedule@StartDate

This OPTIONAL attribute specifies the date at which the schedule becomes effective in `xs:date` format.

`StartDate` is inclusive and if this attribute is not present, the schedule becomes effective today.

/wsme:Condition/wsme:Schedule@StopDate

This OPTIONAL attribute specifies the date at which the schedule stops being effective in `xs:date` format.

`StopDate` is exclusive and the specified date SHOULD be after the start date. When the stop date is before or the same as the start date the schedule is never effective.

If this attribute is not present the schedule is effective indefinitely.

/wsme:Condition/wsme:Schedule/wsme:Daily

This OPTIONAL element specifies the daily timeframe during which the schedule is effective. If this element is not present, the schedule is effective all day.

/wsme:Condition/wsme:Schedule/wsme:Daily@StartTime

This REQUIRED attribute specifies the time at which the schedule starts daily in `xs:time` format.

`StartTime` is inclusive.

/wsme:Condition/wsme:Schedule/wsme:Daily@StopTime

This REQUIRED attribute specifies the time at which the schedule stops daily in `xs:time` format.

`StopTime` is exclusive and if the specified time is earlier than or the same as the daily start time the schedule stops at the specified stop time on the next day.

/wsme:Condition/wsme:Schedule/wsme:Weekdays

This OPTIONAL element specifies the days of the week included in the schedule. If this element is not present, every day of the week is included in the schedule.

This element only affects the start of the daily timeframe as schedules are allowed to run passed midnight. For example, if a schedule is set to start at 11pm and run for 2 hours on Wednesdays, the schedule will effectively end on Thursday at 1am.

/wsme:Condition/wsme:Schedule/wsme:Weekdays@Days

This REQUIRED attribute lists the weekdays included in the schedule, as a list of names separated with the plus sign ('+'), as in "Monday+Tuesday+Wednesday+Thursday+Friday+Saturday+Sunday".

3. Examples of Mediation Rules

In this section we provide a set of common mediation rules that exemplify the use of some of the syntax described in this specification.

Notice that some elements are only applicable when certain values are used.

3.1. Quality of Service (QoS) Mediation Rules

3.1.1. "Max5Errors_Reject" QoS rule

If backend errors exceeds 5 messages in a 10-second interval, then reject any new message until there is less than 5 backend errors per 10-second interval again.

```
<wsp:Policy Name="Max5Errors_Reject">
  <wsme:Rule>
    <wsme:Condition>
        <wsme:Expression>
            <wsme:Attribute>ErrorCount</wsme:Attribute>
            <wsme:Operator>GreaterThan</wsme:Operator>
            <wsme:Value>5</wsme:Value>
            <wsme:Interval>PT10S</wsme:Interval>
        </wsmp:Expression>
    </wsme:Condition>
    <wsme:Action>
        <wsme:RejectMessage/>
    </wsme:Action>
  </wsme:Rule>
</wsp:Policy>
```

3.1.2. “Max300Messages_Queue” QoS rule

If message traffic exceeds 300 messages per minute during 8am to 8pm on Mon, Wed, and Fri, then queue any new message until message traffic is below 300 messages per minute again.

```
<wsp:Policy Name="Max300Messages_Queue">
  <wsme:Rule>
    <wsme:Condition>
        <wsme:Expression>
            <wsme:Attribute>MessageCount</wsme:Attribute>
            <wsme:Operator>GreaterThan</wsme:Operator>
            <wsme:Value>300</wsme:Value>
            <wsme:Interval>PT60S</wsme:Interval>
        </wsme:Expression>
        <wsme:Schedule>
            <wsme:Daily StartTime="08:00:00" StopTime="20:00:00"/>
            <wsme:WeekDays Days="Monday+Wednesday+Friday"/>
        </wsme:Schedule>
    </wsme:Condition>
    <wsme:Action>
        <wsme:QueueMessage/>
    </wsme:Action>
  </wsme:Rule>
</wsp:Policy>
```

3.1.3. “Max100MessagesThenDampenTo50_Reject” QoS rule

From October 1st, 2011 to October 30th, 2011 if message traffic exceeds 100 messages per minute during 8am to 5pm, then reject any new message until message traffic is below 50 messages per minute.

```
<wsp:Policy Name="Max100MessagesThenDampenTo50_Reject">
  <wsme:Rule>
    <wsme:Condition>
        <wsme:Expression>
            <wsme:Attribute>MessageCount</wsme:Attribute>
            <wsme:Operator>HighLow</wsme:Operator>
            <wsme:Value>100</wsme:Value>
            <wsme:Limit>50</wsme:Limit>
            <wsme:Interval>PT60S</wsme:Interval>
        </wsme:Expression>
        <wsme:Schedule StartDate="2011-10-01" StopDate="2011-11-01">
            <wsme:Daily StartTime="08:00:00" StopTime=”17:00:00”/>
        </wsme:Schedule>
    </wsme:Condition>
    <wsme:Action>
        <wsme:RejectMessage/>
    </wsme:Action>
  </wsme:Rule>
</wsp:Policy>
```

3.1.4. “Max300MessagesWith500MessageBurst_Reject” QoS rule

Limit traffic to 300 messages per minute while allowing bursts of 500 messages per minute, rejecting messages in excess.

```
<wsp:Policy Name="Max300MessagesWith500MessageBurst_Reject">
  <wsme:Rule>
    <wsme:Condition>
        <wsme:Expression>
            <wsme:Attribute>MessageCount</wsme:Attribute>
            <wsme:Operator>TockenBucket</wsme:Operator>
            <wsme:Value>300</wsme:Value>
            <wsme:Limit>500</wsme:Limit>
            <wsme:Interval>PT60S</wsme:Interval>
        </wsme:Expression>
    </wsme:Condition>
    <wsme:Action>
        <wsme:RejectMessage/>
    </wsme:Action>
  </wsme:Rule>
</wsp:Policy>
```

3.2. Endpoint Routing Mediation Rules

3.2.1. “HostA_Route” rule

Route all message traffic to endpoint at ‘hostA.acme.com’.

```
<wsp:Policy Name="HostA_Route">
  <wsme:Rule>
    <wsme:Action>
        <wsme:RouteMessage>
           <wsme:EndPoint>hostA.acme.com</wsme:EndPoint>
        </wsme:RouteMessage>
    </wsme:Action>
  </wsme:Rule>
</wsp:Policy>
```

3.2.2. “HACluster_Route” rule

Route all message traffic to endpoint at ‘HACluster’ Load Balancer.

```
<wsp:Policy Name="HACluster_Route">
  <wsme:Rule>
    <wsme:Action>
        <wsme:RouteMessage>
           <wsme:EndPoint>HAClusterLBG</wsme:EndPoint>
        </wsme:RouteMessage>
    </wsme:Action>
  </wsme:Rule>
</wsp:Policy>
```

3.2.3. “9.22.67.115_Route” rule

Route all message traffic to endpoint at ‘9.22.67.115'.

```
<wsp:Policy Name="9.22.67.115_Route">
  <wsme:Rule>
    <wsme:Action>
        <wsme:RouteMessage>
           <wsme:EndPoint>9.22.67.115</wsme:EndPoint>
        </wsme:RouteMessage>
    </wsme:Action>
  </wsme:Rule>
</wsp:Policy>
```

3.3. Message Validation Rules

3.3.1. XML Schema validation

Check that input message validates against a specified XML schema (XSD).

```
<wsp:Policy Name="ValidateMsgXSD">
  <wsme:Rule>
    <wsme:Action>
        <wsme:ValidateMessage>
          <wsme:XSD>
            http://www.example.com/AccountService/message.xsd
          </wsme:XSD>
        </wsme:ValidateMessage>
    </wsme:Action>
  </wsme:Rule>
</wsp:Policy>
```

3.3.2. Web service description validation

Check that input message validates against a specified web service description (WSDL).

```
<wsp:Policy Name="ValidateMsgWSDL">
  <wsme:Rule>
    <wsme:Action>
        <wsme:ValidateMessage>
          <wsme:WSDL>
            http://www.example.com/AccountService/accountservice.wsdl
          </wsme:WSDL>
        </wsme:ValidateMessage>
    </wsme:Action>
  </wsme:Rule>
</wsp:Policy>
```

3.4. Message Translation Rules

Execute a message normalization transform (XSLT) with provided stylesheet parameters.

```
<wsp:Policy Name="NormalizeMessageUsingXSLT">
  <wsme:Rule>
    <wsme:Action>
      <wsme:ExecuteXSL>
         <wsme:Parameter Name="convert-data-from" Value="soap"/>
         <wsme:Parameter Name="convert-data-to" Value="legacy-
format"/>
         <wsme:Stylesheet>
             http://www.example.com/normalize-msg.xsl
         </wsme:Stylesheet>
      </wsme:ExecuteXSL>
    </wsme:Action>
  </wsme:Rule>
</wsp:Policy>
```

3.5. If-Then-Else Rule

If message traffic is less than 100 messages per minute then route message traffic to EndPoint 'A', otherwise route traffic to EndPoint 'B'.

```
<wsp:Policy Name="Max100Messages_RerouteA">
  <wsme:Rule>
    <!-- If... -->
    <wsme:Condition>
        <wsme:Expression>
            <wsme:Attribute>MessageCount</wsme:Attribute>
            <wsme:Operator>LessThan</wsme:Operator>
            <wsme:Value>100</wsme:Value>
        </wsme:Expression>
    </wsme:Condition>
    <!-- Then... -->
    <wsme:Action>
        <wsme:RouteMessage>
           <wsme:EndPoint>A</wsme:EndPoint>
        </wsme:RouteMessage>
    </wsme:Action>
    <!-- Else... -->
    <wsme:Action IfCondition="false">
        <wsme:RouteMessage>
           <wsme:EndPoint>B</wsme:EndPoint>
        </wsme:RouteMessage>
    </wsme:Action>
  </wsme:Rule>
</wsp:Policy>
```

Appendices

A. Normative References

WS-Policy

W3C Recommendation, "Web Services Policy (WS-Policy) 1.5 - Framework" , A. Vedamuthu, et al., Editors. World Wide Web Consortium (W3C), 4 September 2007. Available at http://www.w3.org/TR/ws-policy/.

WS-PolicyAttachment

W3C Recommendation, "Web Services Policy (WS-Policy) 1.5 - Attachment" , A. Vedamuthu, et al., Editors. World Wide Web Consortium (W3C), 4 September 2007. Available at http://www.w3.org/TR/ws-policy-attach/.

XMLSchema - Part 1

W3C Recommendation, "XML Schema Part 1: Structures (Second Edition)" , H. Thompson, et al., Editors. World Wide Web Consortium (W3C), 28 October 2004. Available at http://www.w3.org/TR/xmlschema-1/.

XMLSchema - Part 2

W3C Recommendation, "XML Schema Part 2: Datatypes (Second Edition)" , P. Biron, A. Malhotra, Editors. World Wide Web Consortium (W3C), 28 October 2004. Available at http://www.w3.org/TR/xmlschema-2/.

B. XML Schema

A non-normative copy of the XML schema is listed below for convenience.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="http://www.ibm.com/xmlns/stdwip/2011/02/ws-mediation"
  elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:wsme="http://www.ibm.com/xmlns/stdwip/2011/02/ws-mediation">

  <xs:element name="Rule">
    <xs:complexType>
      <xs:sequence>
        <!-- Left out to avoid non-deterministic content model
             and provide for extensibility
        <xs:element ref="wsme:Condition" maxOccurs="1" minOccurs="0"/>
        <xs:element ref="wsme:Action"
                        maxOccurs="unbounded" minOccurs="1"/> -->
        <xs:any maxOccurs="unbounded" minOccurs="0"
                processContents="lax"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="Condition">
```

```
    <xs:complexType>
      <xs:sequence maxOccurs="1" minOccurs="0">
        <xs:element ref="wsme:Expression"
                    maxOccurs="1" minOccurs="0" />
        <xs:element ref="wsme:Schedule" maxOccurs="1" minOccurs="0" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="Action">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="QueueMessage" maxOccurs="1" minOccurs="0">
          <xs:complexType></xs:complexType>
        </xs:element>

        <xs:choice maxOccurs="unbounded" minOccurs="0">

          <xs:element name="RejectMessage">
            <xs:complexType></xs:complexType>
          </xs:element>

          <xs:element name="Notify">
            <xs:complexType></xs:complexType>
          </xs:element>

          <xs:element name="RouteMessage">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="EndPoint" type="xs:string" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>

          <xs:element name="ValidateMessage">
            <xs:complexType>
              <xs:sequence>
                <xs:choice>
                  <xs:element name="XSD" type="xs:anyURI" />
                  <xs:element name="WSDL" type="xs:anyURI" />
                </xs:choice>
                  <xs:element name="Scope" type="xs:string"
                              maxOccurs="1" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>

          <xs:element name="ExecuteXSL">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Parameter"
                              maxOccurs="unbounded" minOccurs="0">
                  <xs:complexType>
                    <xs:attribute name="Name" type="xs:string" />
                    <xs:attribute name="Value" type="xs:string" />
                  </xs:complexType>
                </xs:element>
                <xs:element name="Stylesheet" type="xs:anyURI" />
              </xs:sequence>
```

```
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:sequence>
      <xs:attribute name="IfCondition" type="xs:boolean"
                                  default="true" use="optional" />
    </xs:complexType>
  </xs:element>

  <xs:element name="Expression">
    <xs:complexType>
      <xs:all>
        <xs:element name="Attribute" type="xs:string" />

        <xs:element name="Operator" type="xs:string" />

        <xs:element name="Value" type="xs:string" />

        <xs:element name="Interval" type="xs:duration" maxOccurs="1"
          minOccurs="0" />

        <xs:element name="Limit" type="xs:string" maxOccurs="1"
          minOccurs="0" />
      </xs:all>
    </xs:complexType>
  </xs:element>

  <xs:element name="Schedule">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Daily" maxOccurs="1" minOccurs="0">
            <xs:complexType>
              <xs:attribute name="StartTime" type="xs:time" />
              <xs:attribute name="StopTime" type="xs:time" />
            </xs:complexType>
        </xs:element>

        <xs:element name="WeekDays" maxOccurs="1" minOccurs="0">
          <xs:complexType>
            <xs:attribute name="Days" type="xs:string" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>

      <xs:attribute name="StartDate" type="xs:date" use="optional"/>
      <xs:attribute name="StopDate" type="xs:date" use="optional"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

C. Acknowledgments (Non-Normative)

This specification was developed with input from several people over a period of time. This includes Robert Laird, Arnaud J Le Hors, Heather Kreger, Katy Warr, Mario De Armas, Mark Weatherill, Steve Groeger, and Thomas Burke.

D. Change Log

Date	Author	Description
2012/01/24	IBM	Document finalized.