Proposal for Issue
http://www.w3.org/Bugs/Public/show_bug.cgi?id=6430

WS-Eventing Appendix I — Service Metadata for Eventing defines an attribute @wse:EventSource="xs:boolean" to annotate WSDL port types to indicate the outbound operations in the port type are events.

The scope of this attribute is incomplete because some dynamic event sources are defined by XML elements instead of port types.

An example of dynamic event source is the sessions defined in ECMA-366, where a WS-Session service (an Event Source as defined by WS-Eventing) exposes multiple sessions created by clients. Each session is an "event source" because it generates an event and its ID is enclosed in a XML element in XSD.

Another example of dynamic event source is the monitors in ECMA-348, where a Switching Function Web service (Event Source) exposes multiple monitors created by Computing Function (Event Subscriber), each generating some events. Again, each monitor has a unique ID enclosed in a XML element in XSD.

The above use cases raise the need to identify not just port types but XML elements as event sources. Since there is already a W3C Recommendation "W3C Semantic Annotations for WSDL and XML Schema" (http://www.w3.org/TR/sawsdl/) designed for this purpose, we recommend using it instead of inventing our own. Using SAWSDL approach, we can still annotate a port type with one attribute as follows:

```
<wsdl:portType name="Warnings"
sawsdl:modelReference="http://schemas.xmlsoap.org/ws/2004/08/eventing/EventSource" >
…
</wsdl:portType>
```

And we can annotate ECMA-348 WSDL/ECMA-323 XSD with the same attribute as follows:

```
<xsd:element name="monitorCrossRefID"
sawsdl:modelReference="http://schemas.xmlsoap.org/ws/2004/08/eventing/EventSource" >
…
</xsd:element>
```

Proposed Changes

Rewrite Appendix I to use SAWSDL as follows.

In order to obtain the event-related metadata that describes a service, the mechanisms described in WS-MetadataExchange [WS-MetadataExchange] should be used. The GetMetadata operation defined there allows WSDL and policy information to be retrieved. The WSDL will contain annotations that identify a service as an event source and that

identify those messages that describe notification messages. The policy will specify the delivery modes and filter types supported by the event source.

To indicate that notification and solicit-response operations within a WSDL 1.1 portType are events exposed by an event source, or a resource identified by a XML element generates events, this specification uses W3C Semantic Annotations for WSDL and XML Schema attribute

@sawsdl:modelReference="http://schemas.xmlsoap.org/ws/2004/08/eventing/EventSource"

to annotate the portType and element for the event source. The normative outlines for this attribute are:

```
<wsdl:portType [sawsdl:modelReference="xs:anyURI"]? >
</wsdl:portType>


<xs:element [sawsdl:modelReference="xs:anyURI"]? >
</xs:element>
```

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

`wsdl:portType/`@sawsdl:modelReference

If omitted, implied value is that the port type does not contain events.

`wsdl:portType/@`
sawsdl:modelReference="http://schemas.xmlsoap.org/ws/2004/08/eventing/EventSource"

Indicates the portType supports the Subscribe operation and indicates that notification and solicit-response operations of the portType are events exposed by a service with a port bound to this portType.

`xs:element/`@sawsdl:modelReference

If omitted, implied value is that the element is not associated with events.

`xs:element/@`
sawsdl:modelReference="http://schemas.xmlsoap.org/ws/2004/08/eventing/EventSource"

Indicates the element supports the Subscribe operation and indicates that the resource identified by the element generates events as notification and solicit-response operations.

Other components of the outline above are not further constrained by this specification.

For example, here is the WSDL 1.1 for a hypothetical storm warning service that exposes a wind report event.

```
<wsdl:definitions
    targetNamespace="http://www.example.org/oceanwatch"
    xmlns:tns="http://www.example.org/oceanwatch"
    xmlns:wse="http://schemas.xmlsoap.org/ws/2004/08/eventing"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:xs="http://www.w3.org/2001/XMLSchema" >
  <wsdl:import
    namespace="http://schemas.xmlsoap.org/ws/2004/08/eventing"
    location=
    "http://schemas.xmlsoap.org/ws/2004/08/eventing/eventing.wsdl" />
  <wsdl:types>
    <xs:schema
      targetNamespace="http://www.example.org/oceanwatch"
      elementFormDefault="qualified"
      blockDefault="#all" >

      <xs:element name="WindID" type="xs:string” sawsdl:modelReference
="http://schemas.xmlsoap.org/ws/2004/08/eventing/EventSource" />

      <xs:element name="WindReport" >
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="tns:WindID" />
            <xs:element name="Date" type="xs:string" />
            <xs:element name="Time" type="xs:string" />
            <xs:element name="Speed" type="xs:string" />
            <xs:element name="Location" type="xs:string" />
            <xs:element name="County" type="xs:string" />
            <xs:element name="State" type="xs:string" />
            <xs:element name="Lat" type="xs:string" />
            <xs:element name="Long" type="xs:string" />
            <xs:element name="Comments" type="xs:string" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
  </wsdl:types>

  <wsdl:message name="WindMsg" >
    <wsdl:part name="body" element="tns:WindReport" />
  </wsdl:message>

  <wsdl:portType name="Warnings" sawsdl:modelReference
="http://schemas.xmlsoap.org/ws/2004/08/eventing/EventSource" >
    <wsdl:operation name="WindOp" >
      <wsdl:output message="tns:WindMsg" />
    </wsdl:operation>
  </wsdl:portType>
</wsdl:definitions>
```

As described here, this event source exposes a port type of events about initial report of winds and allows event subscriber to track a particular wind as it moves. To subscribe to

events exposed by an event source, a subscribing endpoint sends a Subscribe message to the endpoint reference for the event source. If the Subscribe does not include a filter, the event sink should expect to receive and respond to events defined by the outbound operations within the portType or associated with an element.

Editor’s Note: We anticipate that this WSDL extension may change in subsequent versions of this specification.