XML Schema Documentation

Table of Contents

- Schema Document Properties
- Global Definitions
  - Complex Type: **AdditionalData**
  - Complex Type: **ConversationId**
  - Complex Type: **Message**
  - Complex Type: **MessageAddress**
  - Complex Type: **MessageHeader**
  - Complex Type: **MessageId**
  - Complex Type: **MessageRejected**
  - Complex Type: **NotificationMessage**
  - Complex Type: **NotificationMessageHeader**
  - Complex Type: **PartyMessageInformation**
  - Complex Type: **ProblemLocation**
  - Complex Type: **Reason**
  - Complex Type: **ReasonCode**
  - Complex Type: **RequestMessage**
  - Complex Type: **RequestMessageHeader**
  - Complex Type: **RequestTradeStatus**
  - Complex Type: **ResponseMessage**
  - Complex Type: **ResponseMessageHeader**
  - Complex Type: **TradeAlreadyCancelled**
  - Complex Type: **TradeAlreadySubmitted**
  - Complex Type: **TradeAlreadyTerminated**
  - Complex Type: **TradeErrorResponse**
  - Complex Type: **TradeNotFound**
  - Complex Type: **TradeStatus**
  - Complex Type: **TradeStatusItem**
  - Complex Type: **TradeStatusValue**
  - Model Group: **Exception.model**
  - Model Group: **MessageHeader.model**
- Legend
- Glossary

top

Schema Document Properties

Target Namespace	**http://www.fpml.org/2007/FpML-4-4**
Version	$Revision: 2350 $
Element and Attribute Namespaces	• Global element and attribute declarations belong to this schema's target namespace. • By default, local element declarations belong to this schema's target namespace. • By default, local attribute declarations have no namespace.

Schema Composition	• This schema imports schema(s) from the following namespace(s):
	  ○ *http://www.w3.org/2000/09/xmldsig#* (at xmldsig-core-schema.xsd)
	• This schema includes components from the following schema document(s):
	  ○ fpml-doc-4-4.xsd

Declared Namespaces

Prefix	Namespace
Default namespace	**http://www.fpml.org/2007/FpML-4-4**
xml	http://www.w3.org/XML/1998/namespace
ecore	http://www.eclipse.org/emf/2002/Ecore
fpml-annotation	http://www.fpml.org/annotation
fpml	**http://www.fpml.org/2007/FpML-4-4**
dsig	http://www.w3.org/2000/09/xmldsig#
xsd	http://www.w3.org/2001/XMLSchema

Schema Component Representation

```
<xsd:schema nsPrefix="fpml" package="org.fpml" documentRoot="FpML" targetNamespace="http://
www.fpml.org/2007/FpML-4-4" version="$Revision: 2350 $"
attributeFormDefault="unqualified" elementFormDefault="qualified">
   <xsd:import namespace="http://www.w3.org/2000/09/xmldsig#" schemaLocation="xmldsig-core-
   schema.xsd"/>
   <xsd:include schemaLocation="fpml-doc-4-4.xsd"/>
   ...
</xsd:schema>
```

top

Global Definitions

Complex Type: AdditionalData

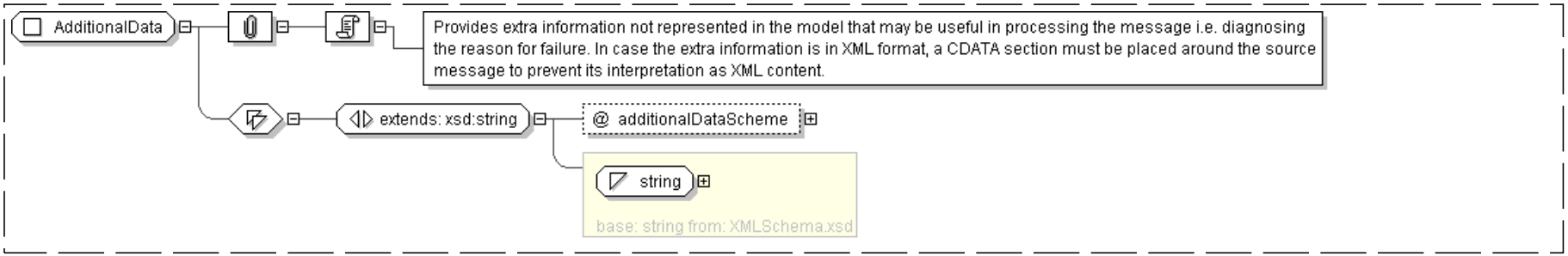
Super-types:	xsd:string < **AdditionalData** (by extension)
Sub-types:	None

Name	AdditionalData
Used by (from the same schema document)	Complex Type **Reason** , Model Group **Exception.model**
Abstract	no
Documentation	Provides extra information not represented in the model that may be useful in processing the message i.e. diagnosing the reason for failure. In case the extra information is in XML format, a CDATA section must be placed around the source message to prevent its interpretation as XML content.

XML Instance Representation

```
<...
additionalDataScheme=" xsd:anyURI [0..1]">
xsd:string
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="AdditionalData">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:string ">
         <xsd:attribute name="additionalDataScheme" type=" xsd:anyURI " use="optional"/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: ConversationId

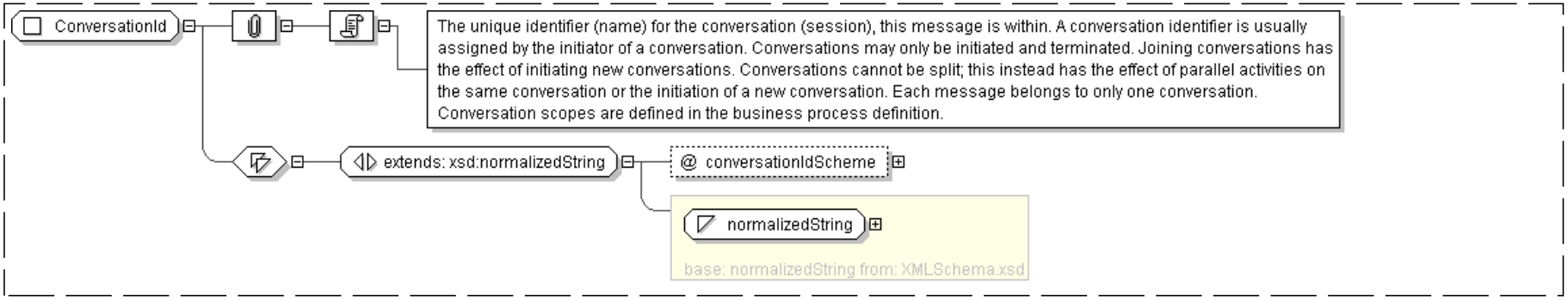
Super-types:	xsd:normalizedString < **ConversationId** (by extension)
Sub-types:	None

Name	ConversationId
Used by (from the same schema document)	Complex Type **MessageHeader**
Abstract	no
Documentation	The unique identifier (name) for the conversation (session), this message is within. A conversation identifier is usually assigned by the initiator of a conversation. Conversations may only be initiated and terminated. Joining conversations has the effect of initiating new conversations. Conversations cannot be split; this instead has the effect of parallel activities on the same conversation or the initiation of a new conversation. Each message belongs to only one conversation. Conversation scopes are defined in the business process definition.

XML Instance Representation

```
<...
conversationIdScheme=" xsd:anyURI [1]">
xsd:normalizedString
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="ConversationId">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="conversationIdScheme" type=" xsd:anyURI " use="required"/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: Message

Super-types:	Document < **Message** (by extension)

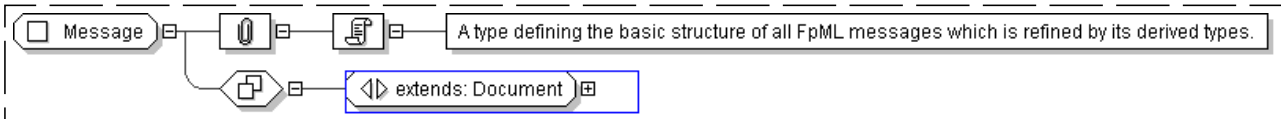
Sub-types:	• NotificationMessage (by extension)
	  ○ MessageRejected (by extension)
	• RequestMessage (by extension)
	  ○ RequestTradeStatus (by extension)
	• ResponseMessage (by extension)
	  ○ TradeNotFound (by extension)
	  ○ TradeStatus (by extension)
	  ○ TradeErrorResponse (by extension)
	  ▪ TradeAlreadyCancelled (by extension)
	  ▪ TradeAlreadyTerminated (by extension)
	  ○ TradeAlreadySubmitted (by extension)

Name	Message
Abstract	yes
Documentation	A type defining the basic structure of all FpML messages which is refined by its derived types.

XML Instance Representation

```
<...
version=" xsd:token (value comes from list: {'4-0'|'4-1'|'4-2'|'4-3'|'4-4'}) [1]
 'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
 'This optional attribute can be supplied by a message creator in an FpML instance to
 specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="2 [0..1]
 'The specific build number of this schema version. This attribute is not included in
 an instance document. Instead, it is supplied by the XML parser when the document is
 validated against the FpML schema and indicates the build number of the schema file. Every
 time FpML publishes a change to the schema, validation rules, or examples within a version
 (e.g., version 4.2) the actual build number is incremented. If no changes have been
 made between releases within a version (i.e. from Trial Recommendation to Recommendation)
 the actual build number stays the same.'

"/>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Message" abstract="true">
   <xsd:complexContent>
      <xsd:extension base=" Document "/>
   </xsd:complexContent>
```

```
</xsd:complexType>
```

top

Complex Type: MessageAddress

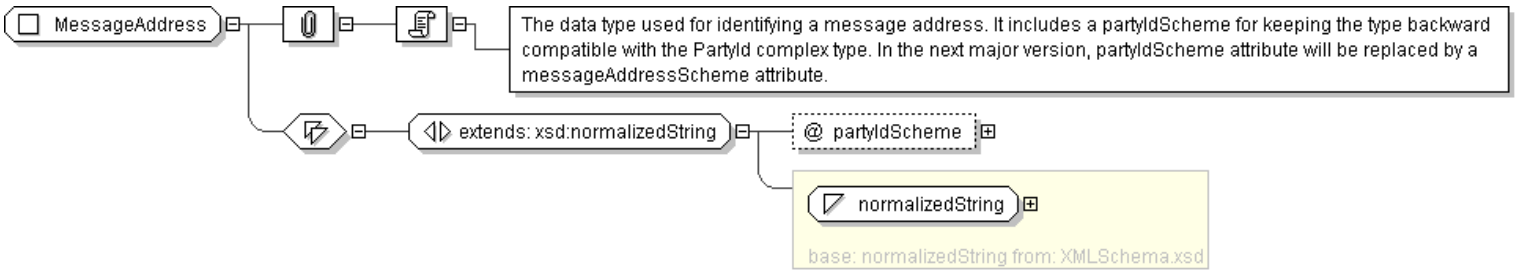
Super-types:	xsd:normalizedString < **MessageAddress** (by extension)
Sub-types:	None

Name	MessageAddress
Used by (from the same schema document)	Model Group **MessageHeader.model** , Model Group **MessageHeader.model** , Model Group **MessageHeader.model**
Abstract	no
Documentation	The data type used for identifying a message address. It includes a partyIdScheme for keeping the type backward compatible with the PartyId complex type. In the next major version, partyIdScheme attribute will be replaced by a messageAddressScheme attribute.

XML Instance Representation

```
<...
partyIdScheme=" xsd:anyURI [0..1]">
xsd:normalizedString
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="MessageAddress">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="partyIdScheme" type=" xsd:anyURI " default="http://www.fpml.org/
         ext/iso9362"/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: MessageHeader

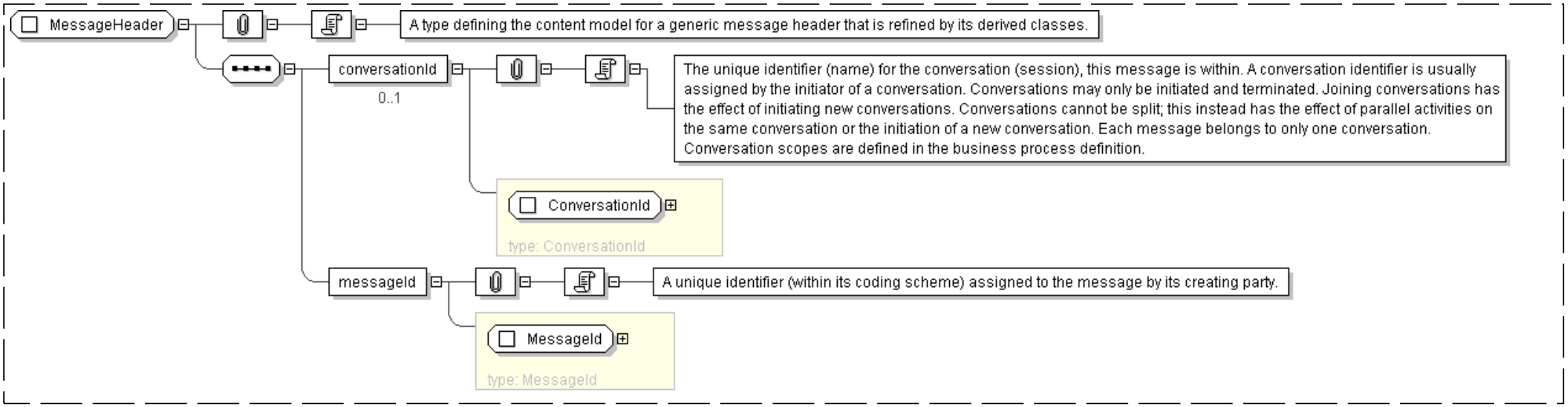
Super-types:	None
Sub-types:	• NotificationMessageHeader (by extension) • RequestMessageHeader (by extension) • ResponseMessageHeader (by extension)

Name	MessageHeader
Abstract	yes
Documentation	A type defining the content model for a generic message header that is refined by its derived classes.

XML Instance Representation

```
<...>
   <conversationId> ConversationId </conversationId> [0..1]
   'The unique identifier (name) for the conversation (session), this message is within.
   A conversation identifier is usually assigned by the initiator of a conversation.
   Conversations may only be initiated and terminated. Joining conversations has the effect
   of initiating new conversations. Conversations cannot be split; this instead has the effect
   of parallel activities on the same conversation or the initiation of a new conversation.
   Each message belongs to only one conversation. Conversation scopes are defined in the
   business process definition.'

   <messageId> MessageId </messageId> [1]
   'A unique identifier (within its coding scheme) assigned to the message by its creating party.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="MessageHeader" abstract="true">
   <xsd:sequence>
      <xsd:element name="conversationId" type=" ConversationId " minOccurs="0"/>
      <xsd:element name="messageId" type=" MessageId "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: MessageId

Super-types:	xsd:normalizedString < **MessageId** (by extension)

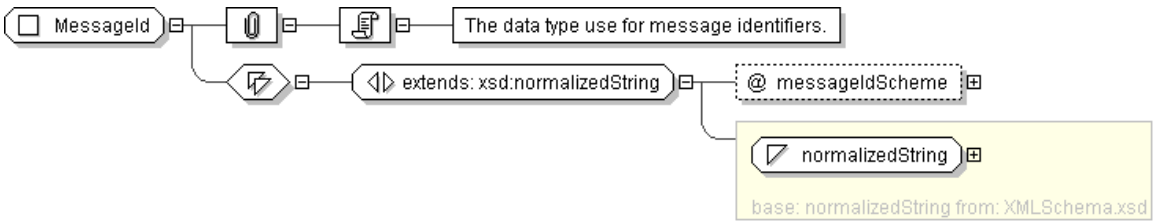
Sub-types:	None

Name	MessageId
Used by (from the same schema document)	Complex Type **MessageHeader** , Complex Type **NotificationMessageHeader** , Complex Type **ResponseMessageHeader**
Abstract	no
Documentation	The data type use for message identifiers.

XML Instance Representation

```
<...
messageIdScheme=" xsd:anyURI [1]">
xsd:normalizedString
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="MessageId">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="messageIdScheme" type=" xsd:anyURI " use="required"/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: MessageRejected

Super-types:	Document < Message (by extension) < NotificationMessage (by extension) < **MessageRejected** (by extension)
Sub-types:	None

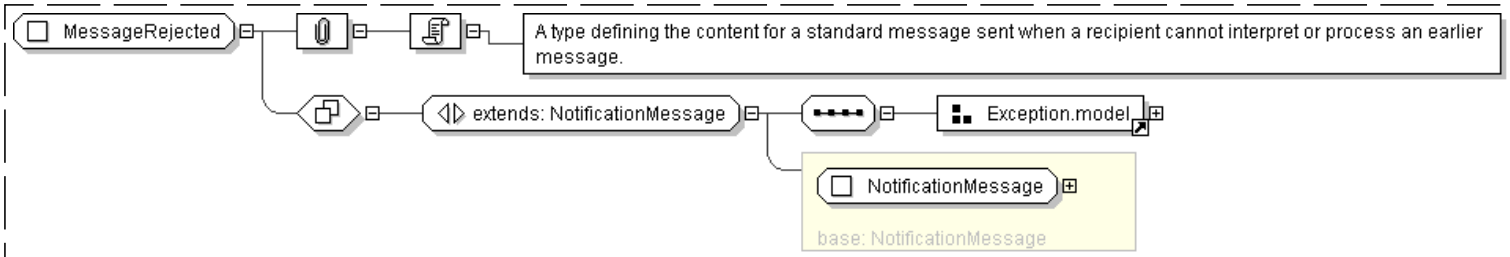
Name	MessageRejected
Abstract	no
Documentation	A type defining the content for a standard message sent when a recipient cannot interpret or process an earlier message.

XML Instance Representation

```
<...
version=" xsd:token (value comes from list: {'4-0'|'4-1'|'4-2'|'4-3'|'4-4'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'
```

```
"
actualBuild="2 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'

">
   <header> NotificationMessageHeader </header> [1]
   <validation> Validation </validation> [0..*]
   <reason> Reason </reason> [1..*]
   'An instance of the Reason type used to record the nature of any errors associated with
   a message.'

   <additionalData> AdditionalData </additionalData> [0..1]
   'Any string of additional data that may help the message processor, for example in a
   rejection message this might contain a code value or the text of the original request (within
   a CDATA section).'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="MessageRejected">
   <xsd:complexContent>
      <xsd:extension base=" NotificationMessage ">
         <xsd:sequence>
            <xsd:group ref=" Exception.model "/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: NotificationMessage

Super-types:	Document < Message (by extension) < **NotificationMessage** (by extension)
Sub-types:	• MessageRejected (by extension)

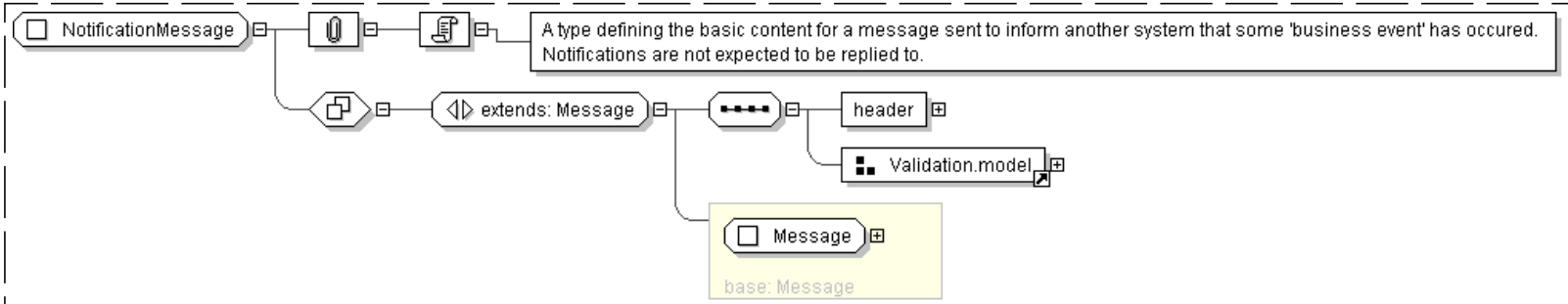
Name	NotificationMessage

Abstract	yes
Documentation	A type defining the basic content for a message sent to inform another system that some 'business event' has occured. Notifications are not expected to be replied to.

XML Instance Representation

```
<...
version=" xsd:token (value comes from list: {'4-0'|'4-1'|'4-2'|'4-3'|'4-4'}) [1]
 'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
 'This optional attribute can be supplied by a message creator in an FpML instance to
 specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="2 [0..1]
 'The specific build number of this schema version. This attribute is not included in
 an instance document. Instead, it is supplied by the XML parser when the document is
 validated against the FpML schema and indicates the build number of the schema file. Every
 time FpML publishes a change to the schema, validation rules, or examples within a version
 (e.g., version 4.2) the actual build number is incremented. If no changes have been
 made between releases within a version (i.e. from Trial Recommendation to Recommendation)
 the actual build number stays the same.'

">
   <header> NotificationMessageHeader </header> [1]
   <validation> Validation </validation> [0..*]
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="NotificationMessage" abstract="true">
   <xsd:complexContent>
      <xsd:extension base=" Message ">
         <xsd:sequence>
            <xsd:element name="header" type=" NotificationMessageHeader "/>
            <xsd:group ref=" Validation.model "/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: NotificationMessageHeader

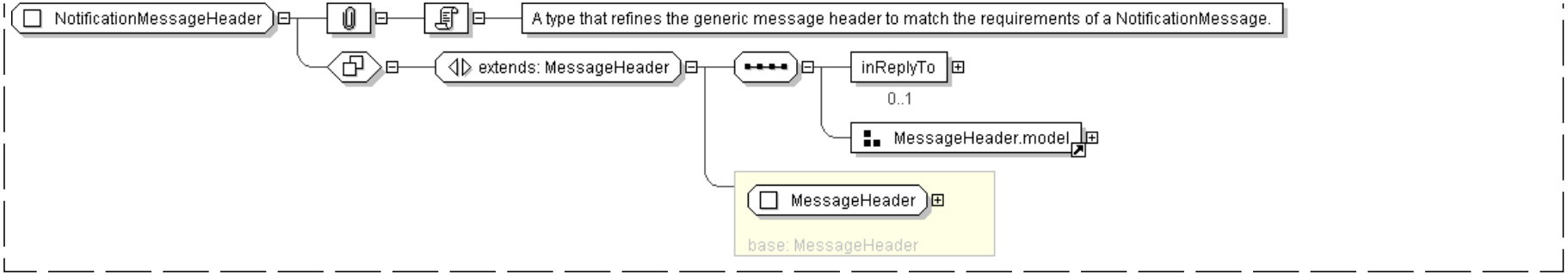
Super-types:	MessageHeader < **NotificationMessageHeader** (by extension)
Sub-types:	None

Name	NotificationMessageHeader
Used by (from the same schema document)	Complex Type **NotificationMessage**
Abstract	no
Documentation	A type that refines the generic message header to match the requirements of a NotificationMessage.

XML Instance Representation

```
<...>
   <conversationId> ConversationId </conversationId> [0..1]
   'The unique identifier (name) for the conversation (session), this message is within.
   A conversation identifier is usually assigned by the initiator of a conversation.
   Conversations may only be initiated and terminated. Joining conversations has the effect
   of initiating new conversations. Conversations cannot be split; this instead has the effect
   of parallel activities on the same conversation or the initiation of a new conversation.
   Each message belongs to only one conversation. Conversation scopes are defined in the
   business process definition.'

   <messageId> MessageId </messageId> [1]
   'A unique identifier (within its coding scheme) assigned to the message by its creating party.'

   <inReplyTo> MessageId </inReplyTo> [0..1]
   'A copy of the unique message identifier (within it own coding scheme) to which this message
   is responding.'

   <sentBy> MessageAddress </sentBy> [1]
   'The unique identifier (within its coding scheme) for the originator of a message instance.'

   <sendTo> MessageAddress </sendTo> [0..*]
   'A unique identifier (within its coding scheme) indicating an intended recipent of a message.'

   <copyTo> MessageAddress </copyTo> [0..*]
   'A unique identifier (within the specified coding scheme) giving the details of some party
   to whom a copy of this message will be sent for reference.'

   <creationTimestamp> xsd:dateTime </creationTimestamp> [1]
   'The date and time (on the source system) when this message instance was created.'

   <expiryTimestamp> xsd:dateTime </expiryTimestamp> [0..1]
   'The date and time (on the source system) when this message instance will be
   considered expired.'

   <partyMessageInformation> PartyMessageInformation </partyMessageInformation> [0..*]
   'Additional message information that may be provided by each involved party.'

   <dsig:Signature> ... </dsig:Signature> [0..*]
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="NotificationMessageHeader">
   <xsd:complexContent>
      <xsd:extension base=" MessageHeader ">
         <xsd:sequence>
            <xsd:element name="inReplyTo" type=" MessageId " minOccurs="0"/>
            <xsd:group ref=" MessageHeader.model "/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: PartyMessageInformation

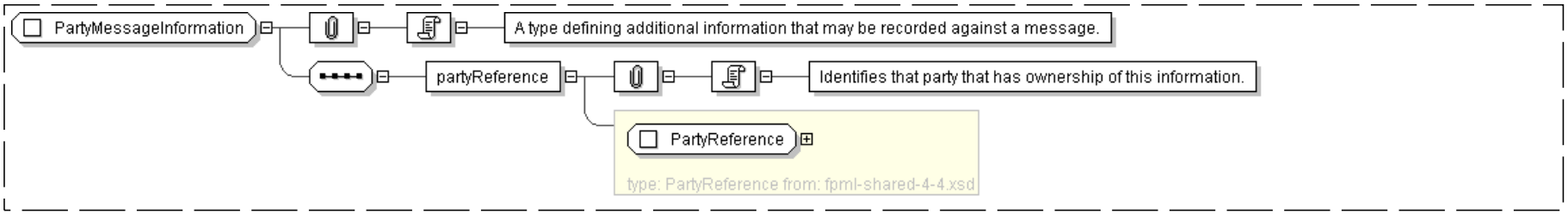
Super-types:	None
Sub-types:	None

Name	PartyMessageInformation
Used by (from the same schema document)	Model Group **MessageHeader.model**
Abstract	no
Documentation	A type defining additional information that may be recorded against a message.

XML Instance Representation

```
<...>
   <partyReference> PartyReference </partyReference> [1]
   'Identifies that party that has ownership of this information.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="PartyMessageInformation">
   <xsd:sequence>
      <xsd:element name="partyReference" type=" PartyReference "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: ProblemLocation

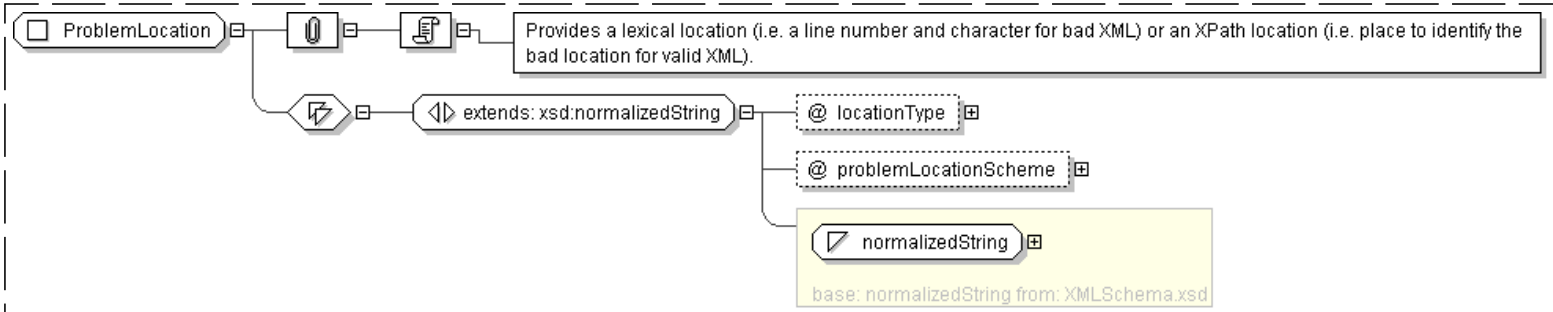
Super-types:	xsd:normalizedString < **ProblemLocation** (by extension)
Sub-types:	None

Name	ProblemLocation
Used by (from the same schema document)	Complex Type **Reason**
Abstract	no
Documentation	Provides a lexical location (i.e. a line number and character for bad XML) or an XPath location (i.e. place to identify the bad location for valid XML).

XML Instance Representation

```
<...
locationType=" xsd:token [0..1]
'The value of the locationType attribute defines which type of location has been given. It
may take the values \'lexical\' or \'xpath\'.'

"
problemLocationScheme=" xsd:anyURI [0..1]
'DEPRECATED. It will be removed in FpML 5.0. New implementations are encouraged to use
the locationType attribute.'

">
xsd:normalizedString
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="ProblemLocation">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="locationType" type=" xsd:token "/>
```

```
        <xsd:attribute name="problemLocationScheme" type=" xsd:anyURI "
        deprecated="true" deprecatedReason="This attribute was introduced by mistake in FpML
        4.2. instead of keeping the locationType attribute."/>
      </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: Reason

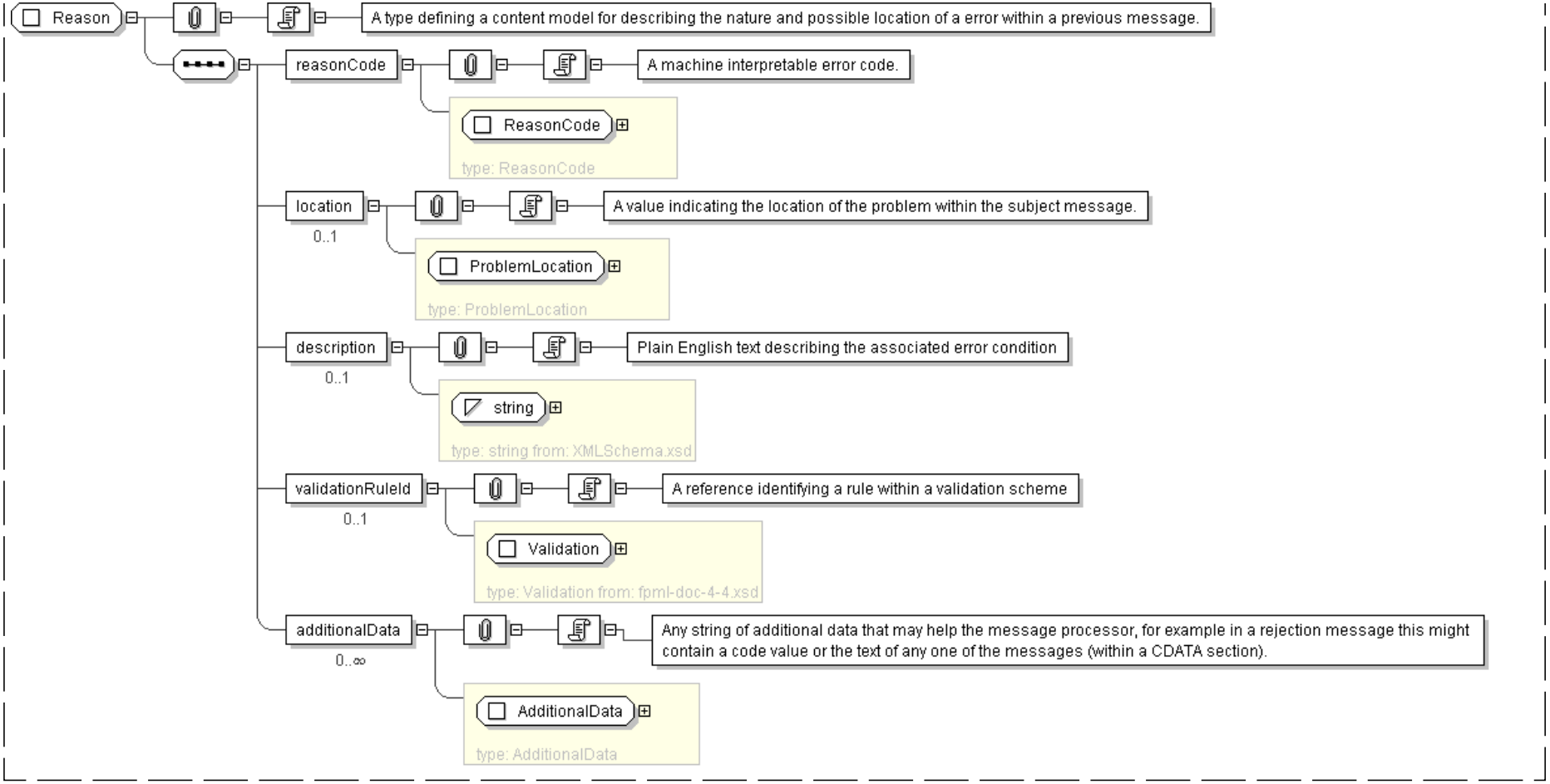
Super-types:	None
Sub-types:	None

Name	Reason
Used by (from the same schema document)	Model Group **Exception.model**
Abstract	no
Documentation	A type defining a content model for describing the nature and possible location of a error within a previous message.

XML Instance Representation

```
<...>
    <reasonCode> ReasonCode </reasonCode> [1]
    'A machine interpretable error code.'

    <location> ProblemLocation </location> [0..1]
    'A value indicating the location of the problem within the subject message.'

    <description> xsd:string </description> [0..1]
    'Plain English text describing the associated error condition'

    <validationRuleId> Validation </validationRuleId> [0..1]
    'A reference identifying a rule within a validation scheme'

    <additionalData> AdditionalData </additionalData> [0..*]
    'Any string of additional data that may help the message processor, for example in a
    rejection message this might contain a code value or the text of any one of the
    messages (within a CDATA section).'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Reason">
   <xsd:sequence>
      <xsd:element name="reasonCode" type=" ReasonCode "/>
      <xsd:element name="location" type=" ProblemLocation " minOccurs="0"/>
      <xsd:element name="description" type=" xsd:string " minOccurs="0"/>
      <xsd:element name="validationRuleId" type=" Validation " minOccurs="0"/>
      <xsd:element name="additionalData" type=" AdditionalData " minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: ReasonCode

Super-types:	xsd:normalizedString < **ReasonCode** (by extension)
Sub-types:	None

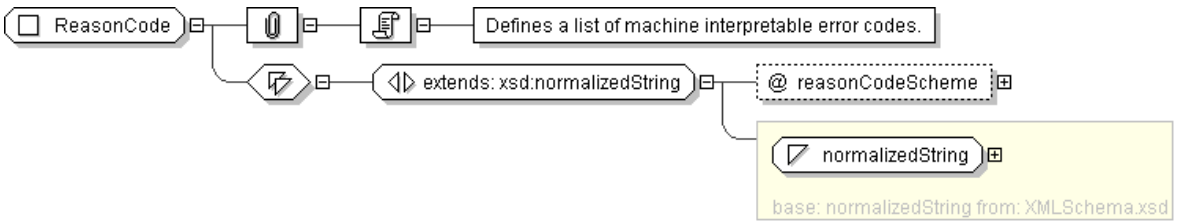
Name	ReasonCode

Used by (from the same schema document)	Complex Type **Reason**
Abstract	no
Documentation	Defines a list of machine interpretable error codes.

XML Instance Representation

```
<...
reasonCodeScheme=" xsd:anyURI [0..1]">
xsd:normalizedString
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="ReasonCode">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="reasonCodeScheme" type=" xsd:anyURI " default="http://www.fpml.org/
         coding-scheme/reason-code-1-0"/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: RequestMessage

Super-types:	Document < Message (by extension) < **RequestMessage** (by extension)
Sub-types:	• RequestTradeStatus (by extension)

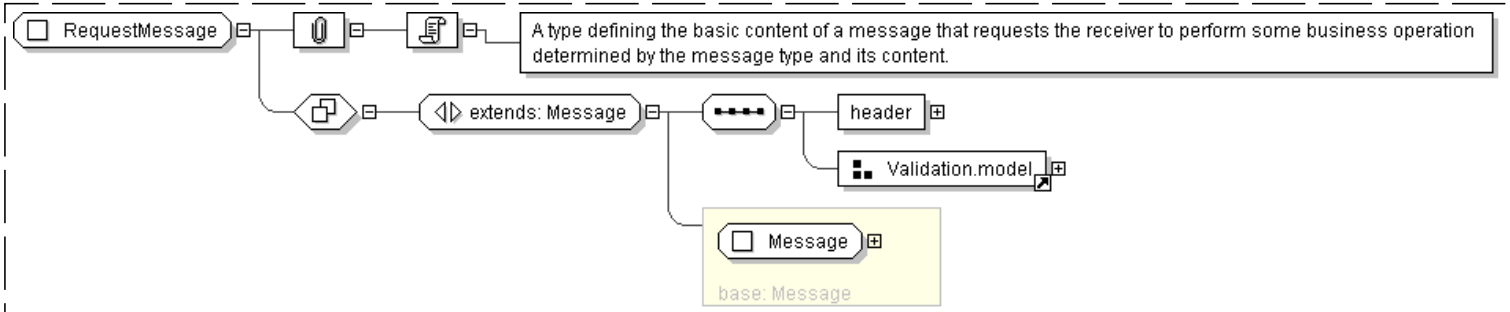
Name	RequestMessage
Abstract	yes
Documentation	A type defining the basic content of a message that requests the receiver to perform some business operation determined by the message type and its content.

XML Instance Representation

```
<...
version=" xsd:token (value comes from list: {'4-0'|'4-1'|'4-2'|'4-3'|'4-4'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'
```

```
"
actualBuild="2 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'

">
   <header> RequestMessageHeader </header> [1]
   <validation> Validation </validation> [0..*]
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="RequestMessage" abstract="true">
   <xsd:complexContent>
      <xsd:extension base=" Message ">
         <xsd:sequence>
            <xsd:element name="header" type=" RequestMessageHeader "/>
            <xsd:group ref=" Validation.model "/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: RequestMessageHeader

Super-types:	MessageHeader < **RequestMessageHeader** (by extension)
Sub-types:	None

Name	RequestMessageHeader
Used by (from the same schema document)	Complex Type **RequestMessage**
Abstract	no
Documentation	A type refining the generic message header content to make it specific to request messages.

XML Instance Representation

```
<...>
```

<conversationId> ConversationId </conversationId> [0..1]

'The unique identifier (name) for the conversation (session), this message is within. A conversation identifier is usually assigned by the initiator of a conversation. Conversations may only be initiated and terminated. Joining conversations has the effect of initiating new conversations. Conversations cannot be split; this instead has the effect of parallel activities on the same conversation or the initiation of a new conversation. Each message belongs to only one conversation. Conversation scopes are defined in the business process definition.'

<messageId> MessageId </messageId> [1]

'A unique identifier (within its coding scheme) assigned to the message by its creating party.'

<sentBy> MessageAddress </sentBy> [1]

'The unique identifier (within its coding scheme) for the originator of a message instance.'

<sendTo> MessageAddress </sendTo> [0..*]

'A unique identifier (within its coding scheme) indicating an intended recipent of a message.'

<copyTo> MessageAddress </copyTo> [0..*]

'A unique identifier (within the specified coding scheme) giving the details of some party to whom a copy of this message will be sent for reference.'

<creationTimestamp> xsd:dateTime </creationTimestamp> [1]

'The date and time (on the source system) when this message instance was created.'

<expiryTimestamp> xsd:dateTime </expiryTimestamp> [0..1]

'The date and time (on the source system) when this message instance will be considered expired.'

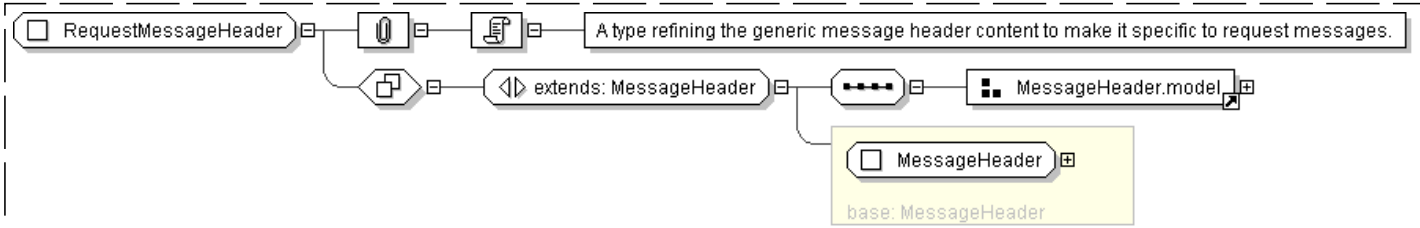
<partyMessageInformation> PartyMessageInformation </partyMessageInformation> [0..*]

'Additional message information that may be provided by each involved party.'

<dsig:Signature> ... </dsig:Signature> [0..*]

</...>

Diagram



Schema Component Representation

```
<xsd:complexType name="RequestMessageHeader">
   <xsd:complexContent>
      <xsd:extension base=" MessageHeader ">
         <xsd:sequence>
            <xsd:group ref=" MessageHeader.model "/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: RequestTradeStatus

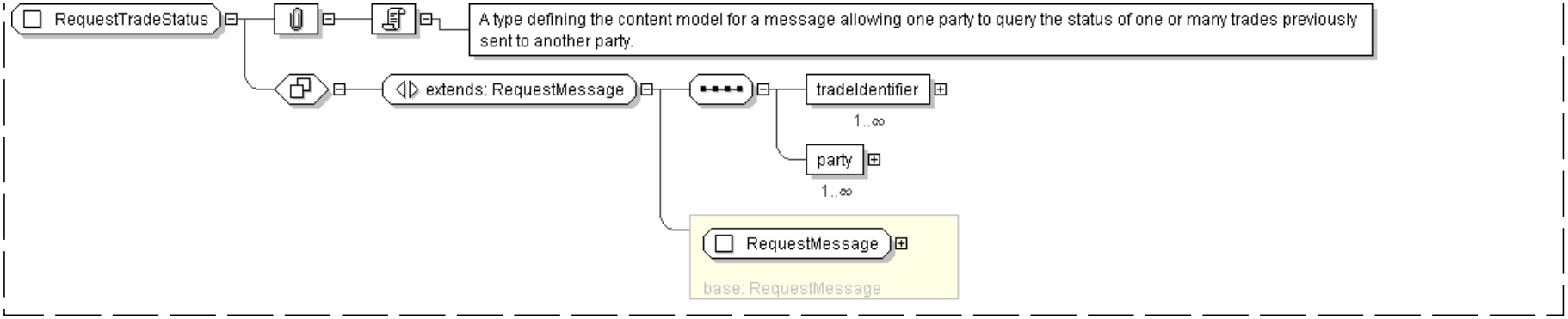
Super-types:	Document < Message (by extension) < RequestMessage (by extension) < **RequestTradeStatus** (by extension)
Sub-types:	None

Name	RequestTradeStatus
Abstract	no
Documentation	A type defining the content model for a message allowing one party to query the status of one or many trades previously sent to another party.

XML Instance Representation

```
<...
version=" xsd:token (value comes from list: {'4-0'|'4-1'|'4-2'|'4-3'|'4-4'}) [1]
 'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
 'This optional attribute can be supplied by a message creator in an FpML instance to
 specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="2 [0..1]
 'The specific build number of this schema version. This attribute is not included in
 an instance document. Instead, it is supplied by the XML parser when the document is
 validated against the FpML schema and indicates the build number of the schema file. Every
 time FpML publishes a change to the schema, validation rules, or examples within a version
 (e.g., version 4.2) the actual build number is incremented. If no changes have been
 made between releases within a version (i.e. from Trial Recommendation to Recommendation)
 the actual build number stays the same.'

">
   <header> RequestMessageHeader </header> [1]
   <validation> Validation </validation> [0..*]
   <tradeIdentifier> TradeIdentifier </tradeIdentifier> [1..*]
    'An instance of a unique trade identifier.'

   <party> Party </party> [1..*]
    'A legal entity or a subdivision of a legal entity.','Parties can perform multiple roles in
    a trade lifecycle. For example, the principal parties obligated to make payments from time
    to time during the term of the trade, but may include other parties involved in, or
    incidental to, the trade, such as parties acting in the role of novation transferor/
    transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
    within a document.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="RequestTradeStatus">
   <xsd:complexContent>
      <xsd:extension base=" RequestMessage ">
         <xsd:sequence>
            <xsd:element name="tradeIdentifier" type=" TradeIdentifier " maxOccurs="unbounded"/>
            <xsd:element name="party" type=" Party " maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: ResponseMessage

Super-types:	Document < Message (by extension) < **ResponseMessage** (by extension)

Sub-types:	• TradeNotFound (by extension)
	• TradeStatus (by extension)
	• TradeErrorResponse (by extension)
	  ○ TradeAlreadyCancelled (by extension)
	  ○ TradeAlreadyTerminated (by extension)
	• TradeAlreadySubmitted (by extension)

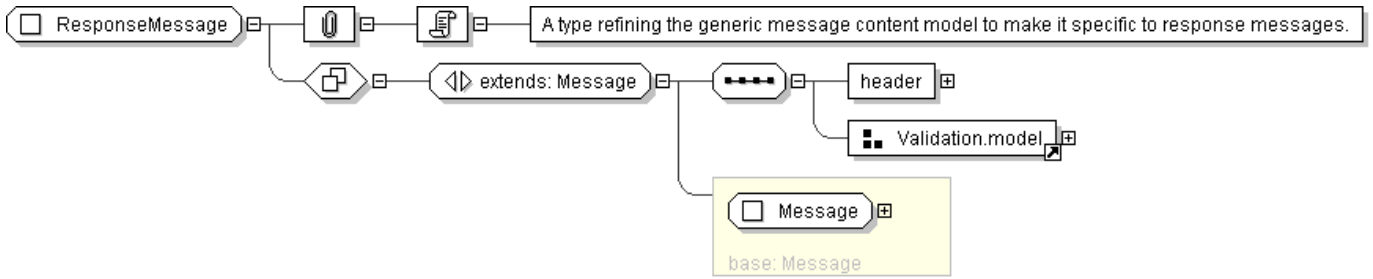
Name	ResponseMessage
Abstract	yes
Documentation	A type refining the generic message content model to make it specific to response messages.

XML Instance Representation

```
<...
version=" xsd:token (value comes from list: {'4-0'|'4-1'|'4-2'|'4-3'|'4-4'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="2 [0..1]
```

```
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'

">
   <header> ResponseMessageHeader </header> [1]
   <validation> Validation </validation> [0..*]
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="ResponseMessage" abstract="true">
   <xsd:complexContent>
      <xsd:extension base=" Message ">
         <xsd:sequence>
            <xsd:element name="header" type=" ResponseMessageHeader "/>
            <xsd:group ref=" Validation.model "/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: ResponseMessageHeader

Super-types:	MessageHeader < **ResponseMessageHeader** (by extension)
Sub-types:	None

Name	ResponseMessageHeader
Used by (from the same schema document)	Complex Type **ResponseMessage**
Abstract	no
Documentation	A type refining the generic message header to make it specific to response messages.

XML Instance Representation

```
<...>
   <conversationId> ConversationId </conversationId> [0..1]
   'The unique identifier (name) for the conversation (session), this message is within.
   A conversation identifier is usually assigned by the initiator of a conversation.
```

Conversations may only be initiated and terminated. Joining conversations has the effect of initiating new conversations. Conversations cannot be split; this instead has the effect of parallel activities on the same conversation or the initiation of a new conversation. Each message belongs to only one conversation. Conversation scopes are defined in the business process definition.'

<messageId> MessageId </messageId> [1]

'A unique identifier (within its coding scheme) assigned to the message by its creating party.'

<inReplyTo> MessageId </inReplyTo> [1]

'A copy of the unique message identifier (within it own coding scheme) to which this message is responding.'

<sentBy> MessageAddress </sentBy> [1]

'The unique identifier (within its coding scheme) for the originator of a message instance.'

<sendTo> MessageAddress </sendTo> [0..*]

'A unique identifier (within its coding scheme) indicating an intended recipent of a message.'

<copyTo> MessageAddress </copyTo> [0..*]

'A unique identifier (within the specified coding scheme) giving the details of some party to whom a copy of this message will be sent for reference.'

<creationTimestamp> xsd:dateTime </creationTimestamp> [1]

'The date and time (on the source system) when this message instance was created.'

<expiryTimestamp> xsd:dateTime </expiryTimestamp> [0..1]

'The date and time (on the source system) when this message instance will be considered expired.'

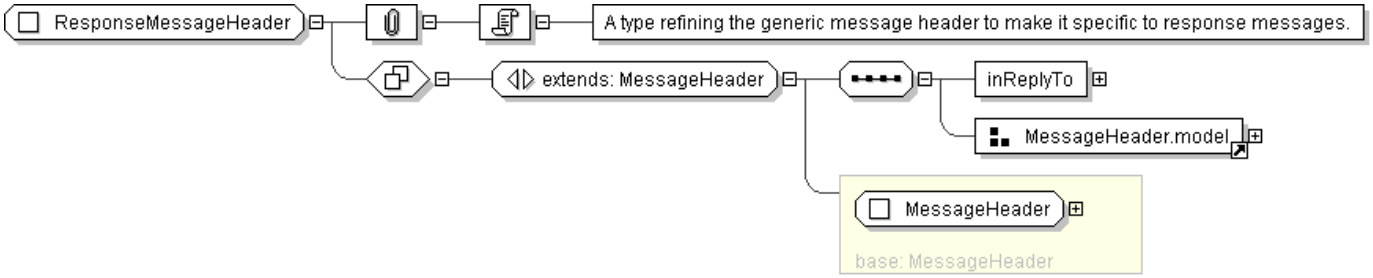
<partyMessageInformation> PartyMessageInformation </partyMessageInformation> [0..*]

'Additional message information that may be provided by each involved party.'

<dsig:Signature> ... </dsig:Signature> [0..*]

</...>

Diagram



Schema Component Representation

```
<xsd:complexType name="ResponseMessageHeader">
   <xsd:complexContent>
      <xsd:extension base=" MessageHeader ">
         <xsd:sequence>
            <xsd:element name="inReplyTo" type=" MessageId "/>
            <xsd:group ref=" MessageHeader.model "/>
```

```
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
```

top

Complex Type: TradeAlreadyCancelled

Super-types:	Document < Message (by extension) < ResponseMessage (by extension) < TradeErrorResponse (by extension) < **TradeAlreadyCancelled** (by extension)
Sub-types:	None

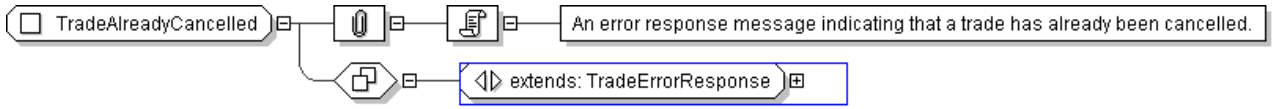
Name	TradeAlreadyCancelled
Abstract	no
Documentation	An error response message indicating that a trade has already been cancelled.

XML Instance Representation

```
<...
version=" xsd:token (value comes from list: {'4-0'|'4-1'|'4-2'|'4-3'|'4-4'}) [1]
 'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
 'This optional attribute can be supplied by a message creator in an FpML instance to
 specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="2 [0..1]
 'The specific build number of this schema version. This attribute is not included in
 an instance document. Instead, it is supplied by the XML parser when the document is
 validated against the FpML schema and indicates the build number of the schema file. Every
 time FpML publishes a change to the schema, validation rules, or examples within a version
 (e.g., version 4.2) the actual build number is incremented. If no changes have been
 made between releases within a version (i.e. from Trial Recommendation to Recommendation)
 the actual build number stays the same.'

">
   <header> ResponseMessageHeader </header> [1]
   <validation> Validation </validation> [0..*]
Start Choice [1]
      <trade> Trade </trade> [1]
       'An element that allows the full details of the trade to be used as a mechanism for
       identifying the trade for which the post-trade event pertains'

      <tradeReference> PartyTradeIdentifiers </tradeReference> [1]
       'A container since an individual trade can be referenced by two or more
       different partyTradeIdentifier elements - each allocated by a different party.'

End Choice
   <party> Party </party> [1..*]
    'A legal entity or a subdivision of a legal entity.','Parties can perform multiple roles in
    a trade lifecycle. For example, the principal parties obligated to make payments from time
    to time during the term of the trade, but may include other parties involved in, or
    incidental to, the trade, such as parties acting in the role of novation transferor/
    transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
```

```
   within a document.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="TradeAlreadyCancelled">
   <xsd:complexContent>
      <xsd:extension base=" TradeErrorResponse "/>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: TradeAlreadySubmitted

Super-types:	Document < Message (by extension) < ResponseMessage (by extension) < **TradeAlreadySubmitted** (by extension)
Sub-types:	None

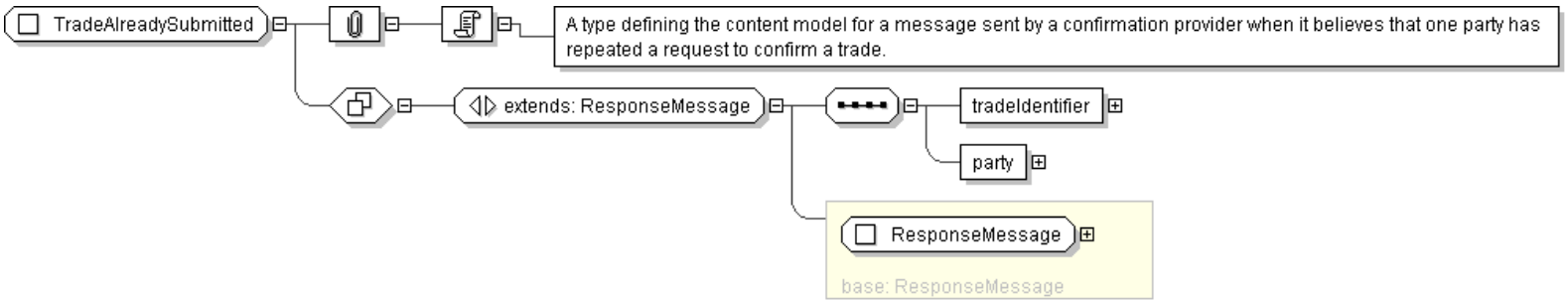
Name	TradeAlreadySubmitted
Abstract	no
Documentation	A type defining the content model for a message sent by a confirmation provider when it believes that one party has repeated a request to confirm a trade.

XML Instance Representation

```
<...
version=" xsd:token (value comes from list: {'4-0'|'4-1'|'4-2'|'4-3'|'4-4'}) [1]
 'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
 'This optional attribute can be supplied by a message creator in an FpML instance to
 specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="2 [0..1]
 'The specific build number of this schema version. This attribute is not included in
 an instance document. Instead, it is supplied by the XML parser when the document is
 validated against the FpML schema and indicates the build number of the schema file. Every
 time FpML publishes a change to the schema, validation rules, or examples within a version
 (e.g., version 4.2) the actual build number is incremented. If no changes have been
 made between releases within a version (i.e. from Trial Recommendation to Recommendation)
 the actual build number stays the same.'

">
   <header> ResponseMessageHeader </header> [1]
   <validation> Validation </validation> [0..*]
   <tradeIdentifier> TradeIdentifier </tradeIdentifier> [1]
    'An instance of a unique trade identifier.'
```

```
<party> Party </party> [1]
'A legal entity or a subdivision of a legal entity.','Parties can perform multiple roles in
a trade lifecycle. For example, the principal parties obligated to make payments from time
to time during the term of the trade, but may include other parties involved in, or
incidental to, the trade, such as parties acting in the role of novation transferor/
transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
within a document.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="TradeAlreadySubmitted">
   <xsd:complexContent>
      <xsd:extension base=" ResponseMessage ">
         <xsd:sequence>
            <xsd:element name="tradeIdentifier" type=" TradeIdentifier "/>
            <xsd:element name="party" type=" Party "/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: TradeAlreadyTerminated

Super-types:	Document < Message (by extension) < ResponseMessage (by extension) < TradeErrorResponse (by extension) < **TradeAlreadyTerminated** (by extension)
Sub-types:	None

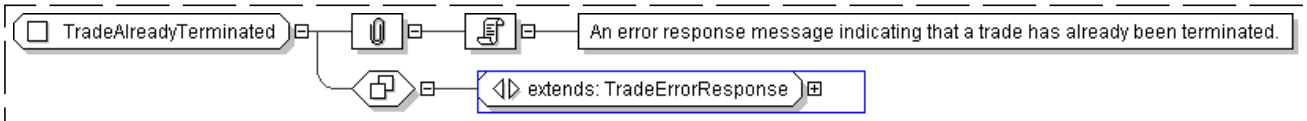
Name	TradeAlreadyTerminated
Abstract	no
Documentation	An error response message indicating that a trade has already been terminated.

XML Instance Representation

```
<...
version=" xsd:token (value comes from list: {'4-0'|'4-1'|'4-2'|'4-3'|'4-4'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
```

```
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="2 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'

">
   <header> ResponseMessageHeader </header> [1]
   <validation> Validation </validation> [0..*]
Start Choice [1]
      <trade> Trade </trade> [1]
      'An element that allows the full details of the trade to be used as a mechanism for
      identifying the trade for which the post-trade event pertains'

      <tradeReference> PartyTradeIdentifiers </tradeReference> [1]
      'A container since an individual trade can be referenced by two or more
      different partyTradeIdentifier elements - each allocated by a different party.'

End Choice
   <party> Party </party> [1..*]
   'A legal entity or a subdivision of a legal entity.','Parties can perform multiple roles in
   a trade lifecycle. For example, the principal parties obligated to make payments from time
   to time during the term of the trade, but may include other parties involved in, or
   incidental to, the trade, such as parties acting in the role of novation transferor/
   transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
   within a document.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="TradeAlreadyTerminated">
   <xsd:complexContent>
      <xsd:extension base=" TradeErrorResponse "/>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: TradeErrorResponse

Super-types:	Document < Message (by extension) < ResponseMessage (by extension) < **TradeErrorResponse** (by extension)

Sub-types:

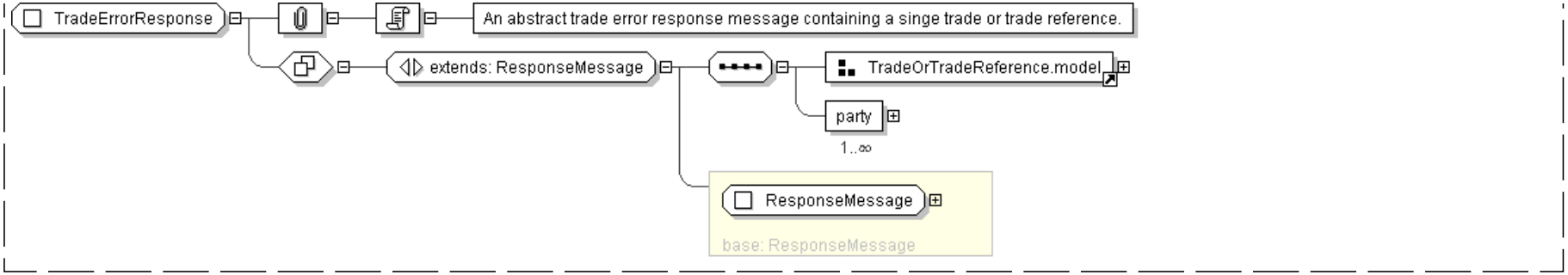
- TradeAlreadyCancelled (by extension)
- TradeAlreadyTerminated (by extension)

Name	TradeErrorResponse
Abstract	yes
Documentation	An abstract trade error response message containing a singe trade or trade reference.

XML Instance Representation

```
<...
version=" xsd:token (value comes from list: {'4-0'|'4-1'|'4-2'|'4-3'|'4-4'}) [1]
 'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
 'This optional attribute can be supplied by a message creator in an FpML instance to
 specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="2 [0..1]
 'The specific build number of this schema version. This attribute is not included in
 an instance document. Instead, it is supplied by the XML parser when the document is
 validated against the FpML schema and indicates the build number of the schema file. Every
 time FpML publishes a change to the schema, validation rules, or examples within a version
 (e.g., version 4.2) the actual build number is incremented. If no changes have been
 made between releases within a version (i.e. from Trial Recommendation to Recommendation)
 the actual build number stays the same.'

">
   <header> ResponseMessageHeader </header> [1]
   <validation> Validation </validation> [0..*]
Start Choice [1]
      <trade> Trade </trade> [1]
       'An element that allows the full details of the trade to be used as a mechanism for
       identifying the trade for which the post-trade event pertains'

      <tradeReference> PartyTradeIdentifiers </tradeReference> [1]
       'A container since an individual trade can be referenced by two or more
       different partyTradeIdentifier elements - each allocated by a different party.'

End Choice
   <party> Party </party> [1..*]
    'A legal entity or a subdivision of a legal entity.','Parties can perform multiple roles in
    a trade lifecycle. For example, the principal parties obligated to make payments from time
    to time during the term of the trade, but may include other parties involved in, or
    incidental to, the trade, such as parties acting in the role of novation transferor/
    transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
    within a document.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="TradeErrorResponse" abstract="true">
   <xsd:complexContent>
      <xsd:extension base=" ResponseMessage ">
         <xsd:sequence>
            <xsd:group ref=" TradeOrTradeReference.model "/>
            <xsd:element name="party" type=" Party " maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: TradeNotFound

Super-types:	Document < Message (by extension) < ResponseMessage (by extension) < **TradeNotFound** (by extension)
Sub-types:	None

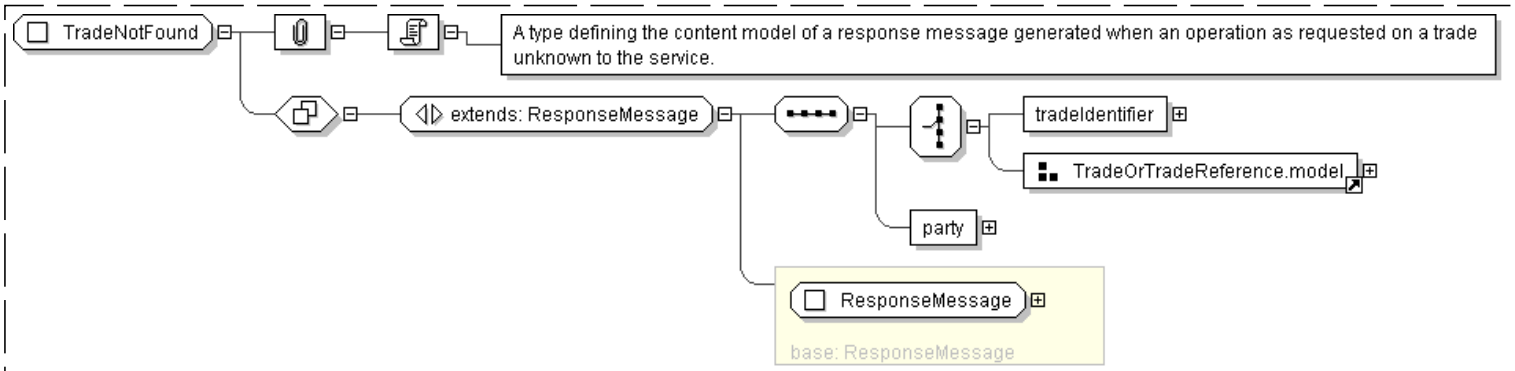
Name	TradeNotFound
Abstract	no
Documentation	A type defining the content model of a response message generated when an operation as requested on a trade unknown to the service.

XML Instance Representation

```
<...
version=" xsd:token (value comes from list: {'4-0'|'4-1'|'4-2'|'4-3'|'4-4'}) [1]
 'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
 'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="2 [0..1]
 'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'
```

```
">
   <header> ResponseMessageHeader </header> [1]
   <validation> Validation </validation> [0..*]
Start Choice [1]
      <tradeIdentifier> TradeIdentifier </tradeIdentifier> [1]
       'An instance of a unique trade identifier.'

Start Choice [1]
         <trade> Trade </trade> [1]
          'An element that allows the full details of the trade to be used as a mechanism for
          identifying the trade for which the post-trade event pertains'

         <tradeReference> PartyTradeIdentifiers </tradeReference> [1]
          'A container since an individual trade can be referenced by two or more
          different partyTradeIdentifier elements - each allocated by a different party.'

End Choice
End Choice
   <party> Party </party> [1]
   'A legal entity or a subdivision of a legal entity.','Parties can perform multiple roles in
   a trade lifecycle. For example, the principal parties obligated to make payments from time
   to time during the term of the trade, but may include other parties involved in, or
   incidental to, the trade, such as parties acting in the role of novation transferor/
   transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
   within a document.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="TradeNotFound">
   <xsd:complexContent>
      <xsd:extension base=" ResponseMessage ">
         <xsd:sequence>
            <xsd:choice>
               <xsd:element name="tradeIdentifier" type=" TradeIdentifier "/>
               <xsd:group ref=" TradeOrTradeReference.model "/>
            </xsd:choice>
            <xsd:element name="party" type=" Party "/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
```

```
</xsd:complexType>
```

top

Complex Type: TradeStatus

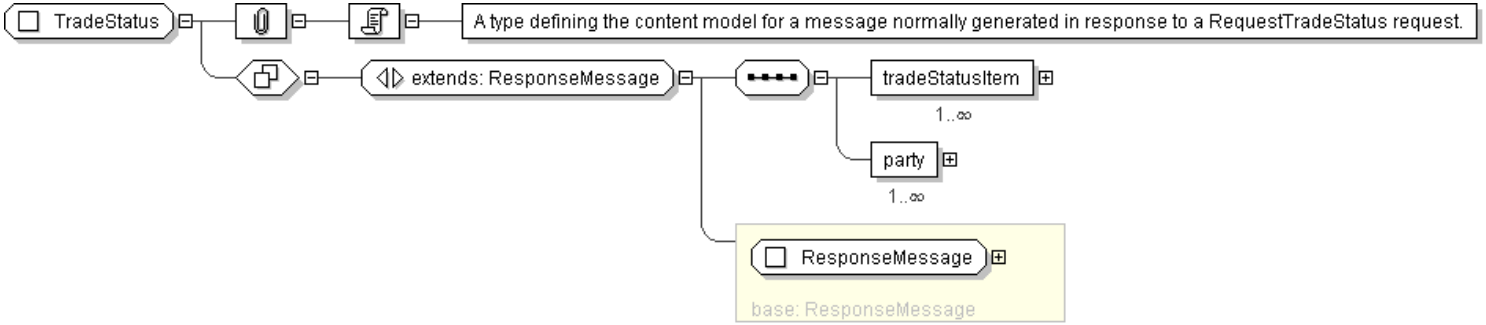
Super-types:	Document < Message (by extension) < ResponseMessage (by extension) < **TradeStatus** (by extension)
Sub-types:	None

Name	TradeStatus
Abstract	no
Documentation	A type defining the content model for a message normally generated in response to a RequestTradeStatus request.

XML Instance Representation

```
<...
version=" xsd:token (value comes from list: {'4-0'|'4-1'|'4-2'|'4-3'|'4-4'}) [1]
'Indicate which version of the FpML Schema an FpML message adheres to.'

"
expectedBuild=" xsd:positiveInteger [0..1]
'This optional attribute can be supplied by a message creator in an FpML instance to
specify which build number of the schema was used to define the message when it was generated.'

"
actualBuild="2 [0..1]
'The specific build number of this schema version. This attribute is not included in
an instance document. Instead, it is supplied by the XML parser when the document is
validated against the FpML schema and indicates the build number of the schema file. Every
time FpML publishes a change to the schema, validation rules, or examples within a version
(e.g., version 4.2) the actual build number is incremented. If no changes have been
made between releases within a version (i.e. from Trial Recommendation to Recommendation)
the actual build number stays the same.'

">
   <header> ResponseMessageHeader </header> [1]
   <validation> Validation </validation> [0..*]
   <tradeStatusItem> TradeStatusItem </tradeStatusItem> [1..*]
   'A collection of data values describing the state of the given trade.'

   <party> Party </party> [1..*]
   'A legal entity or a subdivision of a legal entity.','Parties can perform multiple roles in
   a trade lifecycle. For example, the principal parties obligated to make payments from time
   to time during the term of the trade, but may include other parties involved in, or
   incidental to, the trade, such as parties acting in the role of novation transferor/
   transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places
   within a document.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="TradeStatus">
   <xsd:complexContent>
      <xsd:extension base=" ResponseMessage ">
         <xsd:sequence>
            <xsd:element name="tradeStatusItem" type=" TradeStatusItem " maxOccurs="unbounded"/>
            <xsd:element name="party" type=" Party " maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: TradeStatusItem

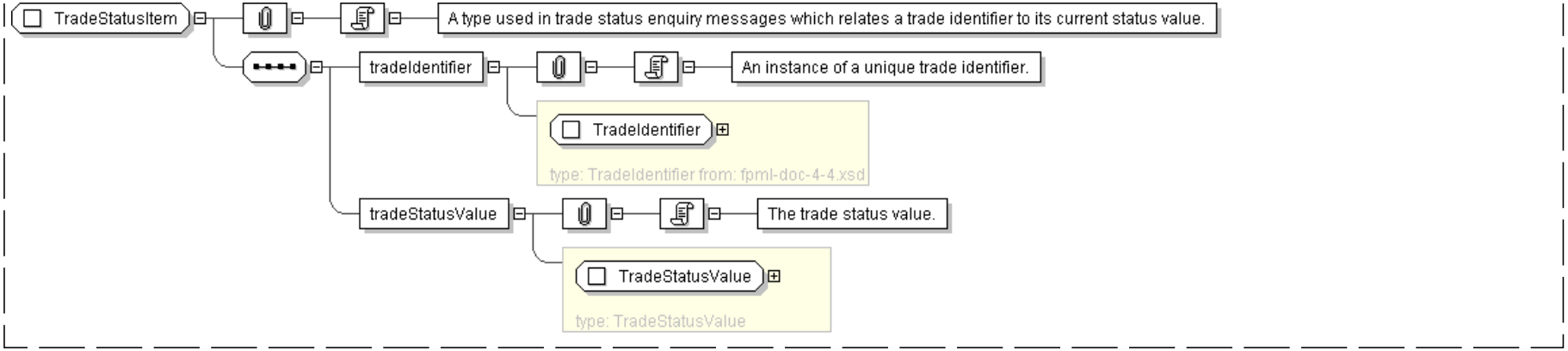
Super-types:	None
Sub-types:	None

Name	TradeStatusItem
Used by (from the same schema document)	Complex Type **TradeStatus**
Abstract	no
Documentation	A type used in trade status enquiry messages which relates a trade identifier to its current status value.

XML Instance Representation

```
<...>
   <tradeIdentifier> TradeIdentifier </tradeIdentifier> [1]
   'An instance of a unique trade identifier.'

   <tradeStatusValue> TradeStatusValue </tradeStatusValue> [1]
   'The trade status value.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="TradeStatusItem">
   <xsd:sequence>
      <xsd:element name="tradeIdentifier" type=" TradeIdentifier "/>
      <xsd:element name="tradeStatusValue" type=" TradeStatusValue "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: TradeStatusValue

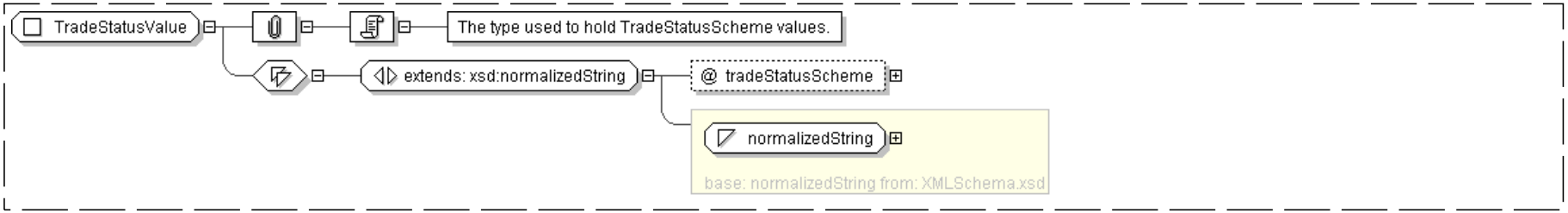
Super-types:	xsd:normalizedString < **TradeStatusValue** (by extension)
Sub-types:	None

Name	TradeStatusValue
Used by (from the same schema document)	Complex Type **TradeStatusItem**
Abstract	no
Documentation	The type used to hold TradeStatusScheme values.

XML Instance Representation

```
<...
tradeStatusScheme=" xsd:anyURI [0..1]">
xsd:normalizedString
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="TradeStatusValue">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="tradeStatusScheme" type=" xsd:anyURI "/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

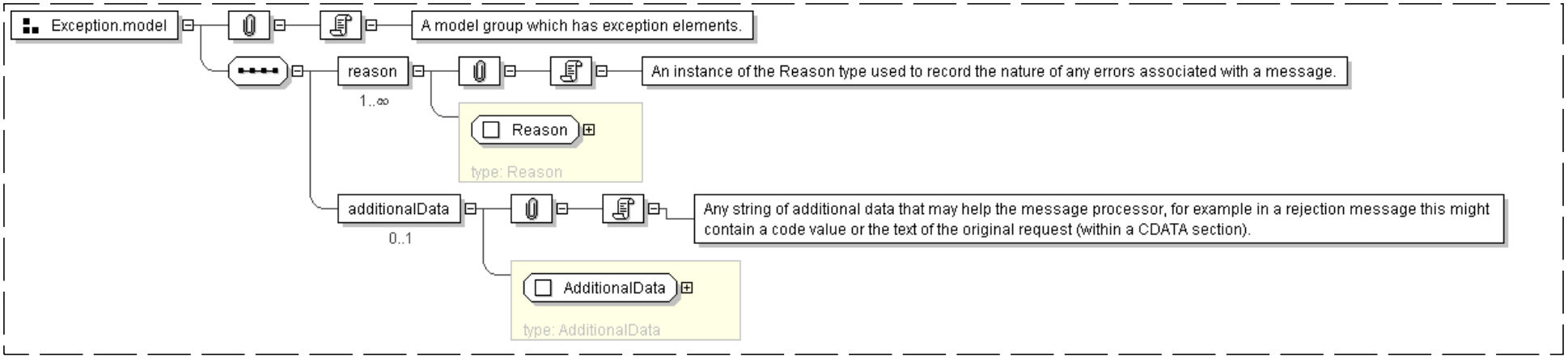
Model Group: Exception.model

Name	Exception.model
Used by (from the same schema document)	Complex Type **MessageRejected**
Documentation	A model group which has exception elements.

XML Instance Representation

```
<reason> Reason </reason> [1..*]
'An instance of the Reason type used to record the nature of any errors associated with
a message.'

<additionalData> AdditionalData </additionalData> [0..1]
'Any string of additional data that may help the message processor, for example in a
rejection message this might contain a code value or the text of the original request (within
a CDATA section).'
```

Diagram



Schema Component Representation

```
<xsd:group name="Exception.model">
   <xsd:sequence>
      <xsd:element name="reason" type=" Reason " maxOccurs="unbounded"/>
      <xsd:element name="additionalData" type=" AdditionalData " minOccurs="0"/>
   </xsd:sequence>
</xsd:group>
```

top

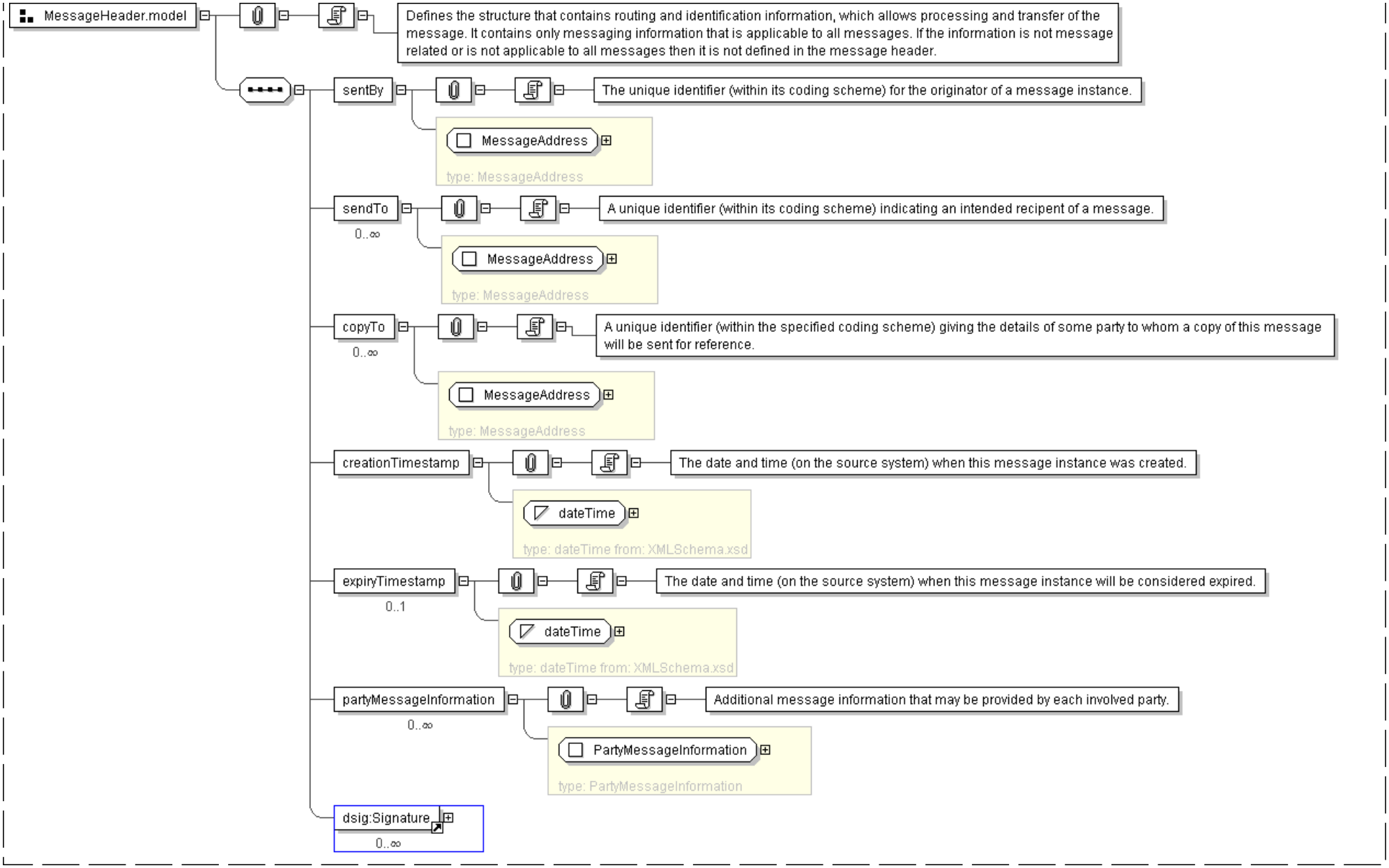
Model Group: MessageHeader.model

Name	MessageHeader.model
Used by (from the same schema document)	Complex Type **NotificationMessageHeader** , Complex Type **RequestMessageHeader** , Complex Type **ResponseMessageHeader**
Documentation	Defines the structure that contains routing and identification information, which allows processing and transfer of the message. It contains only messaging information that is applicable to all messages. If the information is not message related or is not applicable to all messages then it is not defined in the message header.

XML Instance Representation

```
<sentBy> MessageAddress </sentBy> [1]
 'The unique identifier (within its coding scheme) for the originator of a message instance.'

<sendTo> MessageAddress </sendTo> [0..*]
 'A unique identifier (within its coding scheme) indicating an intended recipent of a message.'

<copyTo> MessageAddress </copyTo> [0..*]
 'A unique identifier (within the specified coding scheme) giving the details of some party
 to whom a copy of this message will be sent for reference.'

<creationTimestamp> xsd:dateTime </creationTimestamp> [1]
 'The date and time (on the source system) when this message instance was created.'

<expiryTimestamp> xsd:dateTime </expiryTimestamp> [0..1]
 'The date and time (on the source system) when this message instance will be
 considered expired.'

<partyMessageInformation> PartyMessageInformation </partyMessageInformation> [0..*]
 'Additional message information that may be provided by each involved party.'

<dsig:Signature> ... </dsig:Signature> [0..*]
```

Diagram



Schema Component Representation

```
<xsd:group name="MessageHeader.model">
   <xsd:sequence>
      <xsd:element name="sentBy" type=" MessageAddress "/>
      <xsd:element name="sendTo" type=" MessageAddress " minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="copyTo" type=" MessageAddress " minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="creationTimestamp" type=" xsd:dateTime "/>
      <xsd:element name="expiryTimestamp" type=" xsd:dateTime " minOccurs="0"/>
      <xsd:element name="partyMessageInformation" type=" PartyMessageInformation "
      minOccurs="0" maxOccurs="unbounded"/>
```

```
      <xsd:element ref=" dsig:Signature " minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:group>
```

top

Legend

Complex Type: Schema Component Type

AusAddress Schema Component Name

Super-types:	Address < AusAddress (by extension)
Sub-types:	• QLDAddress (by restriction)

If this schema component is a type definition, its type hierarchy is shown in a gray-bordered box.

Name	AusAddress
Abstract	no

The table above displays the properties of this schema component.

XML Instance Representation

```
<... country="Australia" >
<unitNo> string </unitNo> [0..1]
<houseNo> string </houseNo> [1]
<street> string </street> [1]
Start Choice [1]
<city> string </city> [1]
<town> string </town> [1]
End Choice
<state> AusStates </state> [1]
<postcode> string <<pattern = [1-9][0-9]{3}>> </postcode> [1]
</...>
```

The XML Instance Representation table above shows the schema component's content as an XML instance.

- The minimum and maximum occurrence of elements and attributes are provided in square brackets, e.g. [0..1].
- Model group information are shown in gray, e.g. Start Choice ... End Choice.
- For type derivations, the elements and attributes that have been added to or changed from the base type's content are shown in **bold**.
- If an element/attribute has a fixed value, the fixed value is shown in green, e.g. country="Australia".
- Otherwise, the type of the element/attribute is displayed.
  - If the element/attribute's type is in the schema, a link is provided to it.
  - For local simple type definitions, the constraints are displayed in angle brackets, e.g. <<*pattern* = [1-9][0-9]{3}>>.

Schema Component Representation

```
<complexType name="AusAddress">
<complexContent>
<extension base=" Address ">
<sequence>
<element name="state" type=" AusStates "/>
<element name="postcode">
<simpleType>
<restriction base=" string ">
<pattern value="[1-9][0-9]{3}"/>
```

```
</restriction>
</simpleType>
</element>
</sequence>
<attribute name="country" type=" string " fixed="Australia"/>
</extension>
</complexContent>
</complexType>
```

The Schema Component Representation table above displays the underlying XML representation of the schema component. (Annotations are not shown.)

top

Glossary

Abstract (Applies to complex type definitions and element declarations). An abstract element or complex type cannot used to validate an element instance. If there is a reference to an abstract element, only element declarations that can substitute the abstract element can be used to validate the instance. For references to abstract type definitions, only derived types can be used.

All Model Group Child elements can be provided *in any order* in instances. See: http://www.w3.org/TR/xmlschema-1/#element-all.

Choice Model Group *Only one* from the list of child elements and model groups can be provided in instances. See: http://www.w3.org/TR/xmlschema-1/#element-choice.

Collapse Whitespace Policy Replace tab, line feed, and carriage return characters with space character (Unicode character 32). Then, collapse contiguous sequences of space characters into single space character, and remove leading and trailing space characters.

Disallowed Substitutions (Applies to element declarations). If *substitution* is specified, then substitution group members cannot be used in place of the given element declaration to validate element instances. If *derivation methods*, e.g. extension, restriction, are specified, then the given element declaration will not validate element instances that have types derived from the element declaration's type using the specified derivation methods. Normally, element instances can override their declaration's type by specifying an `xsi:type` attribute.

Key Constraint Like Uniqueness Constraint, but additionally requires that the specified value(s) must be provided. See: http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint_Definitions.

Key Reference Constraint Ensures that the specified value(s) must match value(s) from a Key Constraint or Uniqueness Constraint. See: http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint_Definitions.

Model Group Groups together element content, specifying the order in which the element content can occur and the number of times the group of element content may be repeated. See: http://www.w3.org/TR/xmlschema-1/#Model_Groups.

Nillable (Applies to element declarations). If an element declaration is nillable, instances can use the `xsi:nil` attribute. The `xsi:nil` attribute is the boolean attribute, *nil*, from the *http://www.w3.org/2001/XMLSchema-instance* namespace. If an element instance has an `xsi:nil` attribute set to true, it can be left empty, even though its element declaration may have required content.

Notation A notation is used to identify the format of a piece of data. Values of elements and attributes that are of type, NOTATION, must come from the names of declared notations. See: http://www.w3.org/TR/xmlschema-1/#cNotation_Declarations.

Preserve Whitespace Policy Preserve whitespaces exactly as they appear in instances.

Prohibited Derivations (Applies to type definitions). Derivation methods that cannot be used to create sub-types from a given type definition.

Prohibited Substitutions (Applies to complex type definitions). Prevents sub-types that have been derived using the specified derivation methods from validating element instances in place of the given type definition.

Replace Whitespace Policy Replace tab, line feed, and carriage return characters with space character (Unicode character 32).

Sequence Model Group Child elements and model groups must be provided *in the specified order* in instances. See: http://www.w3.org/TR/xmlschema-1/#element-sequence.

Substitution Group Elements that are *members* of a substitution group can be used wherever the *head* element of the substitution group is referenced.

Substitution Group Exclusions (Applies to element declarations). Prohibits element declarations from nominating themselves as being able to substitute a given element declaration, if they have types that are derived from the original element's type using the specified derivation methods.

Target Namespace The target namespace identifies the namespace that components in this schema belongs to. If no target namespace is provided, then the schema components do not belong to any namespace.

Uniqueness Constraint Ensures uniqueness of an element/attribute value, or a combination of values, within a specified scope. See: http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint_Definitions.

top

Generated by <oXygen/> XML Editor using a modified version of xs3p that adds schema diagrams and chunking support.