XML Schema Documentation

Table of Contents

- Schema Document Properties
- Global Definitions
  - Complex Type: **Asian**
  - Complex Type: **AveragingPeriod**
  - Complex Type: **AveragingSchedule**
  - Complex Type: **Barrier**
  - Complex Type: **CalendarSpread**
  - Complex Type: **ClassifiedPayment**
  - Complex Type: **Composite**
  - Complex Type: **CreditEventNotice**
  - Complex Type: **CreditEvents**
  - Complex Type: **CreditEventsReference**
  - Complex Type: **FailureToPay**
  - Complex Type: **FeaturePayment**
  - Complex Type: **FrequencyType**
  - Complex Type: **FxFeature**
  - Complex Type: **GracePeriodExtension**
  - Complex Type: **Knock**
  - Complex Type: **MarketDisruption**
  - Complex Type: **NotifyingParty**
  - Complex Type: **OptionBase**
  - Complex Type: **OptionBaseExtended**
  - Complex Type: **OptionFeature**
  - Complex Type: **OptionNumericStrike**
  - Complex Type: **OptionStrike**
  - Complex Type: **PassThrough**
  - Complex Type: **PassThroughItem**
  - Complex Type: **Premium**
  - Complex Type: **PubliclyAvailableInformation**
  - Complex Type: **Quanto**
  - Complex Type: **Restructuring**
  - Complex Type: **RestructuringType**
  - Complex Type: **StrategyFeature**
  - Complex Type: **StrikeSpread**
  - Complex Type: **Trigger**
  - Complex Type: **TriggerEvent**
  - Model Group: **OptionBaseFeature.model**
  - Model Group: **OptionDenomination.model**
  - Model Group: **OptionFeature.model**
  - Model Group: **OptionSettlement.model**
- Legend
- Glossary

top

Schema Document Properties

Target Namespace	**http://www.fpml.org/2007/FpML-4-4**
Version	$Revision: 2590 $
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 includes components from the following schema document(s): ○ fpml-asset-4-4.xsd

Declared Namespaces

Prefix	Namespace
Default namespace	http://www.fpml.org/2007/FpML-4-4
ecore	http://www.eclipse.org/emf/2002/Ecore
xml	http://www.w3.org/XML/1998/namespace
xsd	http://www.w3.org/2001/XMLSchema
fpml	http://www.fpml.org/2007/FpML-4-4

Schema Component Representation

```
<xsd:schema nsPrefix="fpml" package="org.fpml" documentRoot="FpML" targetNamespace="http://
www.fpml.org/2007/FpML-4-4" version="$Revision: 2590 $"
attributeFormDefault="unqualified" elementFormDefault="qualified">
   <xsd:include schemaLocation="fpml-asset-4-4.xsd"/>
   ...
</xsd:schema>
```

top

Global Definitions

Complex Type: Asian

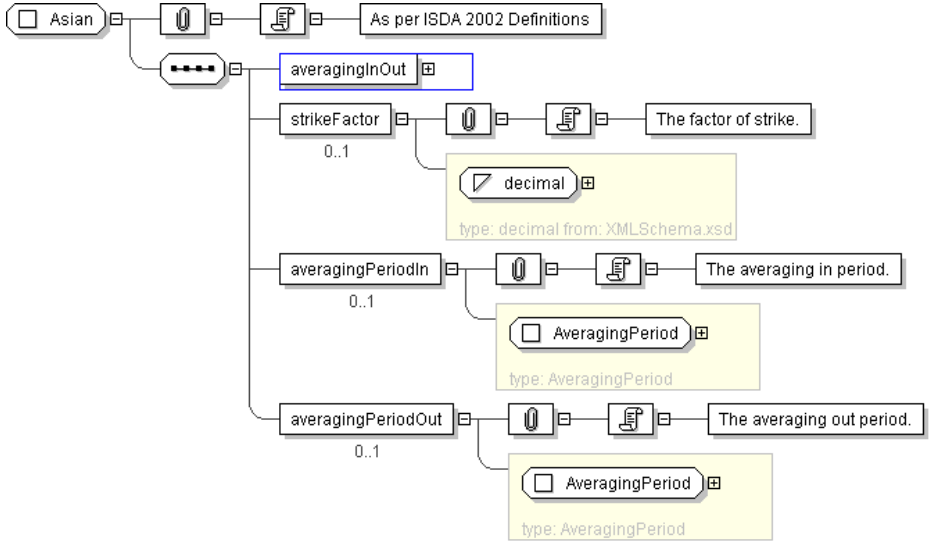
Super-types:	None
Sub-types:	None

Name	Asian
Used by (from the same schema document)	Model Group **OptionFeature.model**
Abstract	no
Documentation	As per ISDA 2002 Definitions

XML Instance Representation

```
<...>
   <averagingInOut> AveragingInOutEnum </averagingInOut> [1]
   <strikeFactor> xsd:decimal </strikeFactor> [0..1]
   'The factor of strike.'

   <averagingPeriodIn> AveragingPeriod </averagingPeriodIn> [0..1]
   'The averaging in period.'

   <averagingPeriodOut> AveragingPeriod </averagingPeriodOut> [0..1]
   'The averaging out period.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Asian">
   <xsd:sequence>
      <xsd:element name="averagingInOut" type=" AveragingInOutEnum "/>
      <xsd:element name="strikeFactor" type=" xsd:decimal " minOccurs="0"/>
      <xsd:element name="averagingPeriodIn" type=" AveragingPeriod " minOccurs="0"/>
      <xsd:element name="averagingPeriodOut" type=" AveragingPeriod " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: AveragingPeriod

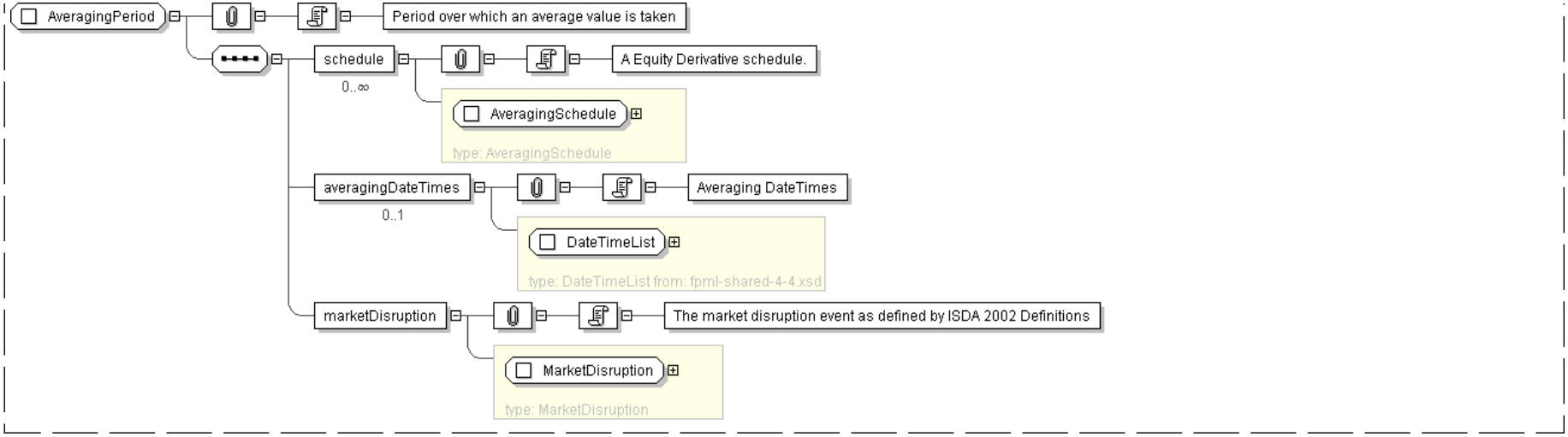
Super-types:	None
Sub-types:	None

Name	AveragingPeriod
Used by (from the same schema document)	Complex Type **Asian** , Complex Type **Asian**
Abstract	no
Documentation	Period over which an average value is taken

XML Instance Representation

```
<...>
   <schedule> AveragingSchedule </schedule> [0..*]
   'A Equity Derivative schedule.'

   <averagingDateTimes> DateTimeList </averagingDateTimes> [0..1]
   'Averaging DateTimes'

   <marketDisruption> MarketDisruption </marketDisruption> [1]
   'The market disruption event as defined by ISDA 2002 Definitions'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="AveragingPeriod">
   <xsd:sequence>
      <xsd:element name="schedule" type=" AveragingSchedule " minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="averagingDateTimes" type=" DateTimeList " minOccurs="0"/>
      <xsd:element name="marketDisruption" type=" MarketDisruption "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: AveragingSchedule

Super-types:	None
Sub-types:	None

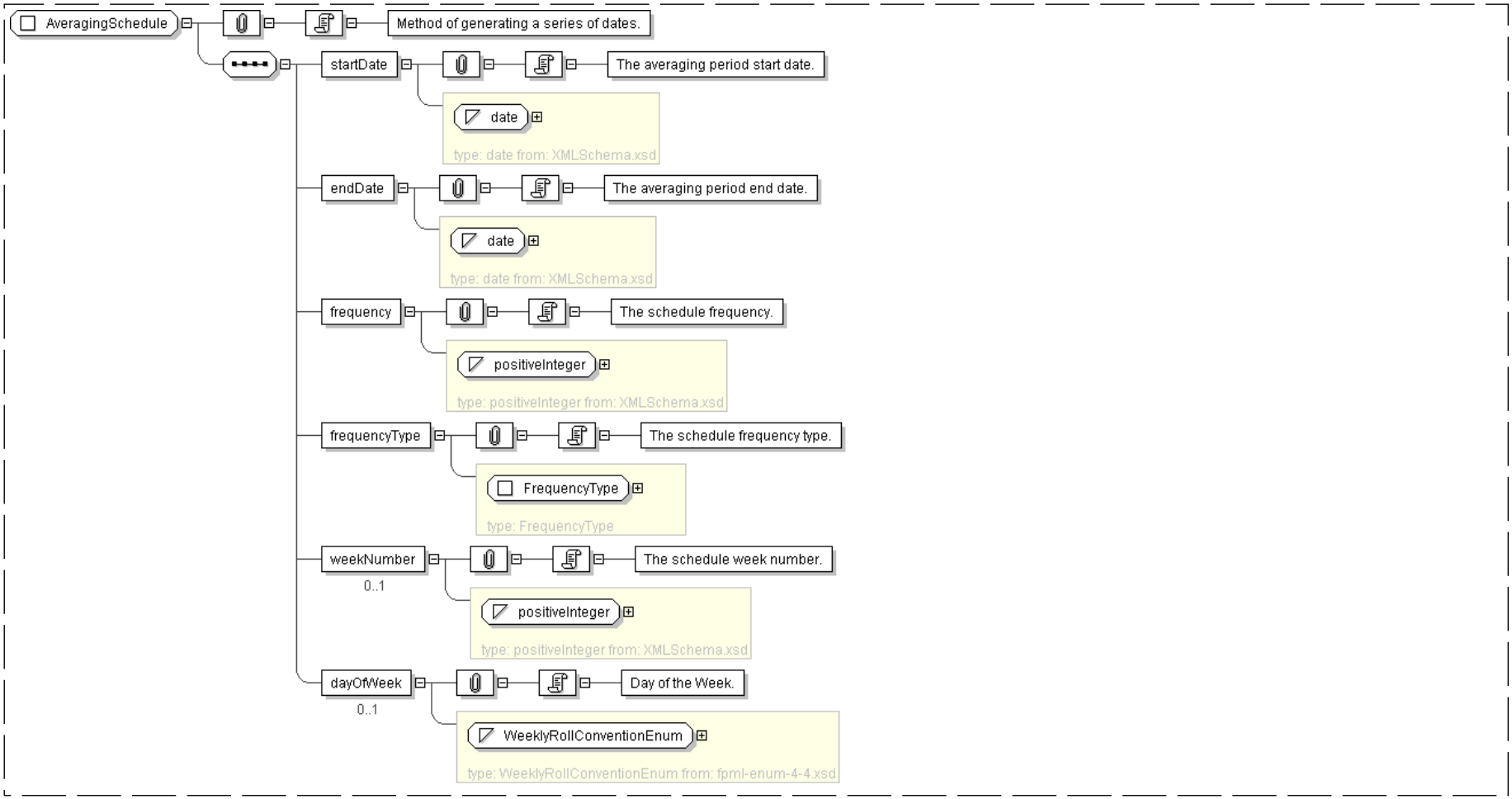
Name	AveragingSchedule
Used by (from the same schema document)	Complex Type **AveragingPeriod** , Complex Type **TriggerEvent**
Abstract	no
Documentation	Method of generating a series of dates.

XML Instance Representation

```
<...>
   <startDate> xsd:date </startDate> [1]
   'The averaging period start date.'

   <endDate> xsd:date </endDate> [1]
   'The averaging period end date.'

   <frequency> xsd:positiveInteger </frequency> [1]
   'The schedule frequency.'

   <frequencyType> FrequencyType </frequencyType> [1]
   'The schedule frequency type.'

   <weekNumber> xsd:positiveInteger </weekNumber> [0..1]
   'The schedule week number.'

   <dayOfWeek> WeeklyRollConventionEnum </dayOfWeek> [0..1]
```

```
    'Day of the Week.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="AveragingSchedule">
   <xsd:sequence>
      <xsd:element name="startDate" type=" xsd:date "/>
      <xsd:element name="endDate" type=" xsd:date "/>
      <xsd:element name="frequency" type=" xsd:positiveInteger "/>
      <xsd:element name="frequencyType" type=" FrequencyType "/>
      <xsd:element name="weekNumber" type=" xsd:positiveInteger " minOccurs="0"/>
      <xsd:element name="dayOfWeek" type=" WeeklyRollConventionEnum " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: Barrier

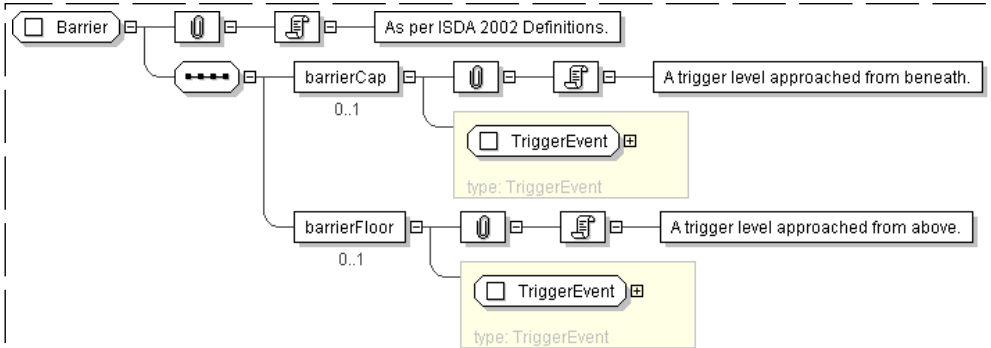
Super-types:	None
Sub-types:	None

Name	Barrier
Used by (from the same schema document)	Model Group **OptionFeature.model**
Abstract	no
Documentation	As per ISDA 2002 Definitions.

XML Instance Representation

```
<...>
   <barrierCap> TriggerEvent </barrierCap> [0..1]
   'A trigger level approached from beneath.'

   <barrierFloor> TriggerEvent </barrierFloor> [0..1]
   'A trigger level approached from above.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Barrier">
   <xsd:sequence>
      <xsd:element name="barrierCap" type=" TriggerEvent " minOccurs="0"/>
      <xsd:element name="barrierFloor" type=" TriggerEvent " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: CalendarSpread

Super-types:	None
Sub-types:	None

Name	CalendarSpread
Used by (from the same schema document)	Complex Type **StrategyFeature**
Abstract	no
Documentation	A type for defining a calendar spread feature

XML Instance Representation

```
<...>
   <expirationDateTwo> AdjustableOrRelativeDate </expirationDateTwo> [1]
</...>
```

Diagram

Schema Component Representation

```
<xsd:complexType name="CalendarSpread">
   <xsd:sequence>
      <xsd:element name="expirationDateTwo" type=" AdjustableOrRelativeDate "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: ClassifiedPayment

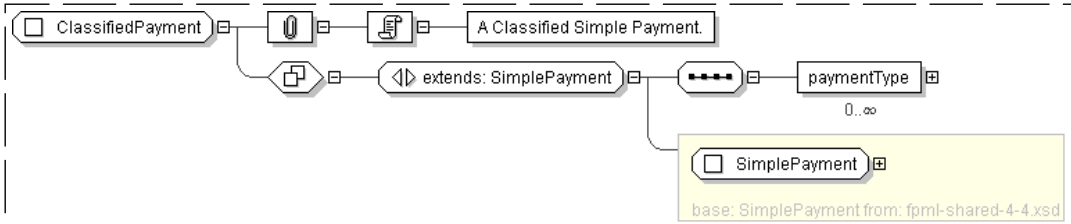
Super-types:	SimplePayment < **ClassifiedPayment** (by extension)
Sub-types:	None

Name	ClassifiedPayment
Abstract	no
Documentation	A Classified Simple Payment.

XML Instance Representation

```
<...>
   <payerPartyReference> PartyOrAccountReference </payerPartyReference> [1]
   'A reference to the party responsible for making the payments defined by this structure.'

   <receiverPartyReference> PartyOrAccountReference </receiverPartyReference> [1]
   'A reference to the party that receives the payments corresponding to this structure.'

   <paymentAmount> Money </paymentAmount> [1]
   <paymentDate> AdjustableOrRelativeAndAdjustedDate </paymentDate> [1]
   'The payment date. This date is subject to adjustment in accordance with any
   applicable business day convention.'

   <paymentType> PaymentType </paymentType> [0..*]
   'Classification of this Payment.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="ClassifiedPayment">
   <xsd:complexContent>
      <xsd:extension base=" SimplePayment ">
         <xsd:sequence>
            <xsd:element name="paymentType" type=" PaymentType " minOccurs="0" maxOccurs="unbounded"/>
```

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

top

Complex Type: Composite

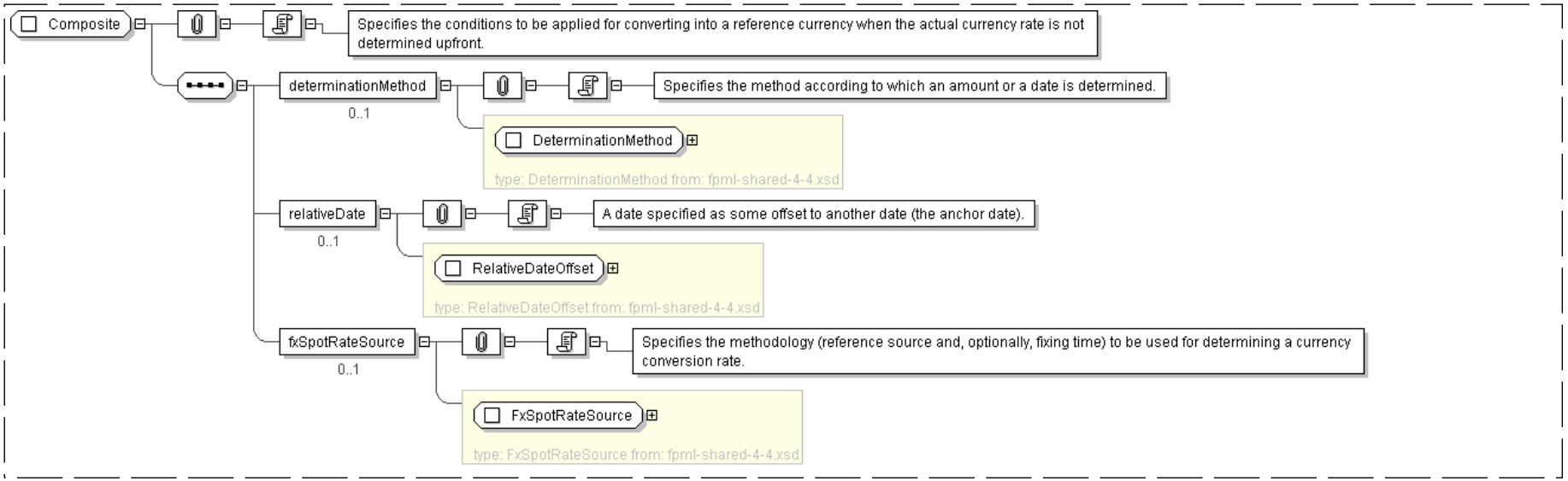
Super-types:	None
Sub-types:	None

Name	Composite
Used by (from the same schema document)	Complex Type **FxFeature** , Complex Type **FxFeature**
Abstract	no
Documentation	Specifies the conditions to be applied for converting into a reference currency when the actual currency rate is not determined upfront.

XML Instance Representation

```
<...>
   <determinationMethod> DeterminationMethod </determinationMethod> [0..1]
   'Specifies the method according to which an amount or a date is determined.'

   <relativeDate> RelativeDateOffset </relativeDate> [0..1]
   'A date specified as some offset to another date (the anchor date).'

   <fxSpotRateSource> FxSpotRateSource </fxSpotRateSource> [0..1]
   'Specifies the methodology (reference source and, optionally, fixing time) to be used
   for determining a currency conversion rate.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Composite">
   <xsd:sequence>
      <xsd:element name="determinationMethod" type=" DeterminationMethod " minOccurs="0"/>
      <xsd:element name="relativeDate" type=" RelativeDateOffset " minOccurs="0"/>
```

```
      <xsd:element name="fxSpotRateSource" type=" FxSpotRateSource " minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
```

top

Complex Type: CreditEventNotice

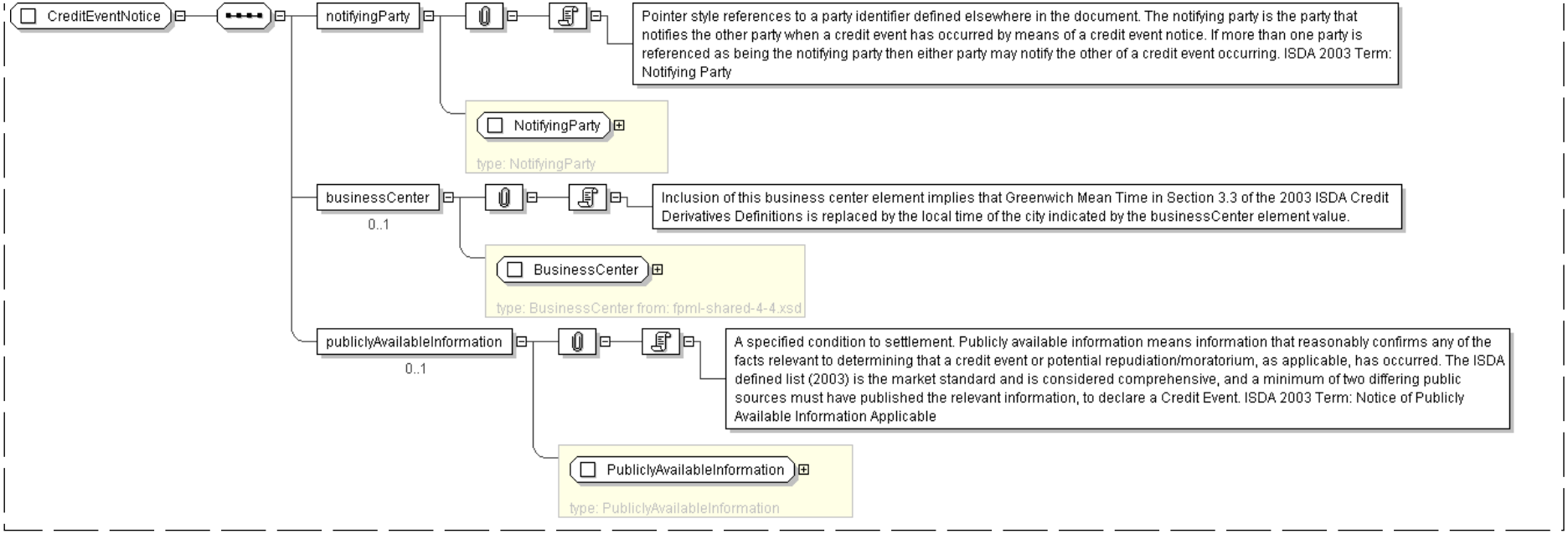
Super-types:	None
Sub-types:	None

Name	CreditEventNotice
Used by (from the same schema document)	Complex Type **CreditEvents**
Abstract	no

XML Instance Representation

```
<...>
   <notifyingParty> NotifyingParty </notifyingParty> [1]
   'Pointer style references to a party identifier defined elsewhere in the document.
   The notifying party is the party that notifies the other party when a credit event has
   occurred by means of a credit event notice. If more than one party is referenced as being
   the notifying party then either party may notify the other of a credit event occurring.
   ISDA 2003 Term: Notifying Party'

   <businessCenter> BusinessCenter </businessCenter> [0..1]
   'Inclusion of this business center element implies that Greenwich Mean Time in Section 3.3
   of the 2003 ISDA Credit Derivatives Definitions is replaced by the local time of the
   city indicated by the businessCenter element value.'

   <publiclyAvailableInformation> PubliclyAvailableInformation </
   publiclyAvailableInformation> [0..1]
   'A specified condition to settlement. Publicly available information means information
   that reasonably confirms any of the facts relevant to determining that a credit event
   or potential repudiation/moratorium, as applicable, has occurred. The ISDA defined list
   (2003) is the market standard and is considered comprehensive, and a minimum of two
   differing public sources must have published the relevant information, to declare a
   Credit Event. ISDA 2003 Term: Notice of Publicly Available Information Applicable'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="CreditEventNotice">
   <xsd:sequence>
      <xsd:element name="notifyingParty" type=" NotifyingParty "/>
      <xsd:element name="businessCenter" type=" BusinessCenter " minOccurs="0"/>
      <xsd:element name="publiclyAvailableInformation" type=" PubliclyAvailableInformation
      " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: CreditEvents

Super-types:	None
Sub-types:	None

Name	CreditEvents
Used by (from the same schema document)	Complex Type **Trigger**
Abstract	no

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
   <bankruptcy> Empty </bankruptcy> [0..1]
   'A credit event. The reference entity has been dissolved or has become insolvent. It
   also covers events that may be a precursor to insolvency such as instigation of bankruptcy
   or insolvency proceedings. Sovereign trades are not subject to Bankruptcy as \"technically\"
   a Sovereign cannot become bankrupt. ISDA 2003 Term: Bankruptcy'

   <failureToPay> FailureToPay </failureToPay> [0..1]
   'A credit event. This credit event triggers, after the expiration of any applicable
   grace period, if the reference entity fails to make due payments in an aggregrate amount of
   not less than the payment requirement on one or more obligations (e.g. a missed
   coupon payment). ISDA 2003 Term: Failure to Pay'
```

<failureToPayPrincipal> Empty </failureToPayPrincipal> [0..1]

'A credit event. Corresponds to the failure by the Reference Entity to pay an expected principal amount or the payment of an actual principal amount that is less than the expected principal amount. ISDA 2003 Term: Failure to Pay Principal.'

<failureToPayInterest> Empty </failureToPayInterest> [0..1]

'A credit event. Corresponds to the failure by the Reference Entity to pay an expected interest amount or the payment of an actual interest amount that is less than the expected interest amount. ISDA 2003 Term: Failure to Pay Interest.'

<obligationDefault> Empty </obligationDefault> [0..1]

'A credit event. One or more of the obligations have become capable of being declared due and payable before they would otherwise have been due and payable as a result of, or on the basis of, the occurrence of a default, event of default or other similar condition or event other than failure to pay. ISDA 2003 Term: Obligation Default'

<obligationAcceleration> Empty </obligationAcceleration> [0..1]

'A credit event. One or more of the obligations have been declared due and payable before they would otherwise have been due and payable as a result of, or on the basis of, the occurrence of a default, event of default or other similar condition or event other than failure to pay (preferred by the market over Obligation Default, because more definitive and encompasses the definition of Obligation Default - this is more favorable to the Seller). Subject to the default requirement amount. ISDA 2003 Term: Obligation Acceleration'

<repudiationMoratorium> Empty </repudiationMoratorium> [0..1]

'A credit event. The reference entity, or a governmental authority, either refuses to recognise or challenges the validity of one or more obligations of the reference entity, or imposes a moratorium thereby postponing payments on one or more of the obligations of the reference entity. Subject to the default requirement amount. ISDA 2003 Term: Repudiation/Moratorium'

<restructuring> Restructuring </restructuring> [0..1]

'A credit event. A restructuring is an event that materially impacts the reference entity \'s obligations, such as an interest rate reduction, principal reduction, deferral of interest or principal, change in priority ranking, or change in currency or composition of payment. ISDA 2003 Term: Restructuring'

<distressedRatingsDowngrade> Empty </distressedRatingsDowngrade> [0..1]

'A credit event. Results from the fact that the rating of the reference obligation is downgraded to a distressed rating level. From a usage standpoint, this credit event is typically not applicable in case of RMBS trades.'

<maturityExtension> Empty </maturityExtension> [0..1]

'A credit event. Results from the fact that the underlier fails to make principal payments as expected.'

<writedown> Empty </writedown> [0..1]

'A credit event. Results from the fact that the underlier writes down its outstanding principal amount.'

<defaultRequirement> Money </defaultRequirement> [0..1]

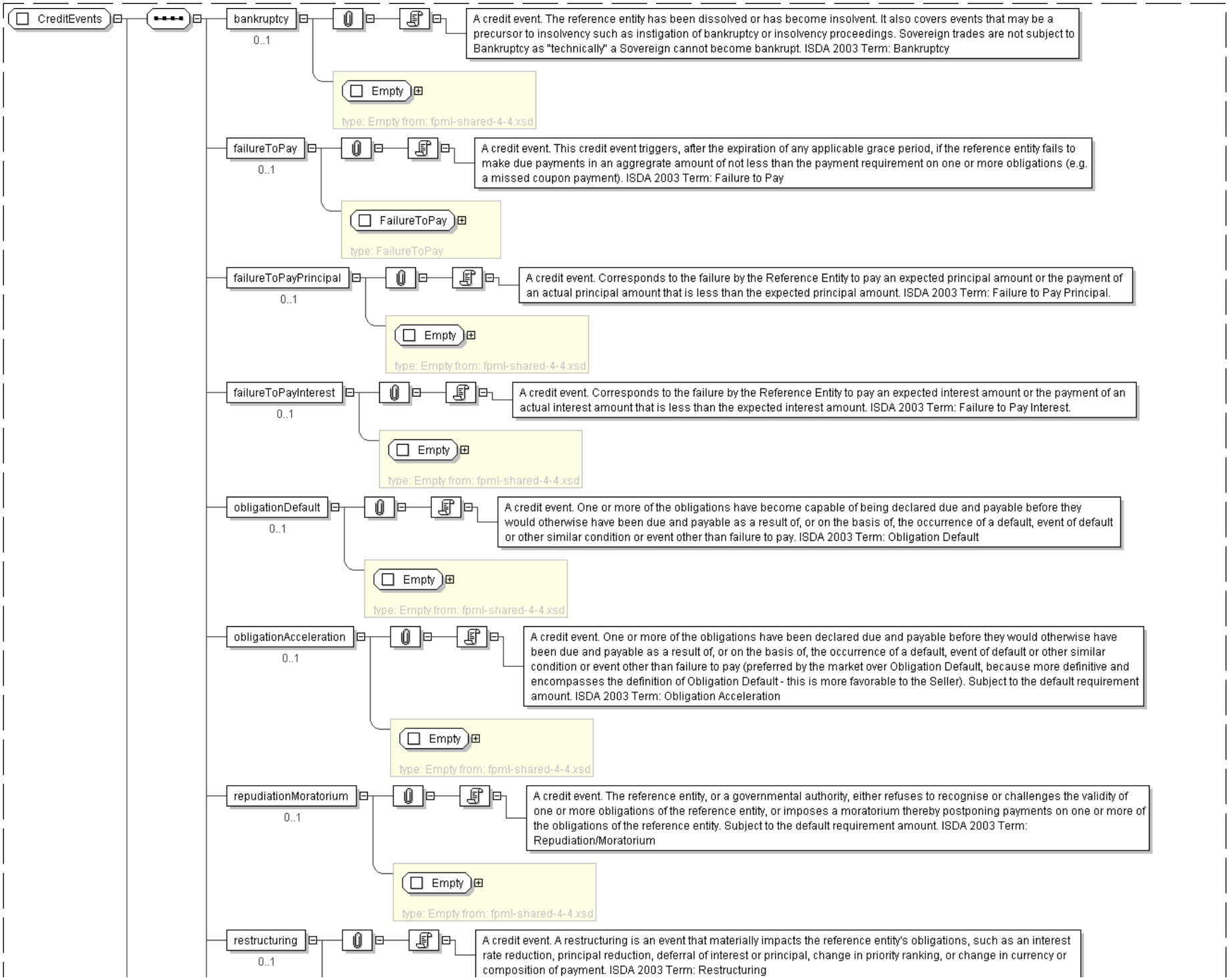
'In relation to certain credit events, serves as a threshold for Obligation Acceleration, Obligation Default, Repudiation/Moratorium and Restructuring. Market standard is USD 10,000,000 (JPY 1,000,000,000 for all Japanese Yen trades). This is applied on an aggregate or total basis across all Obligations of the Reference Entity. Used to prevent technical/operational errors from triggering credit events. ISDA 2003 Term: Default Requirement'

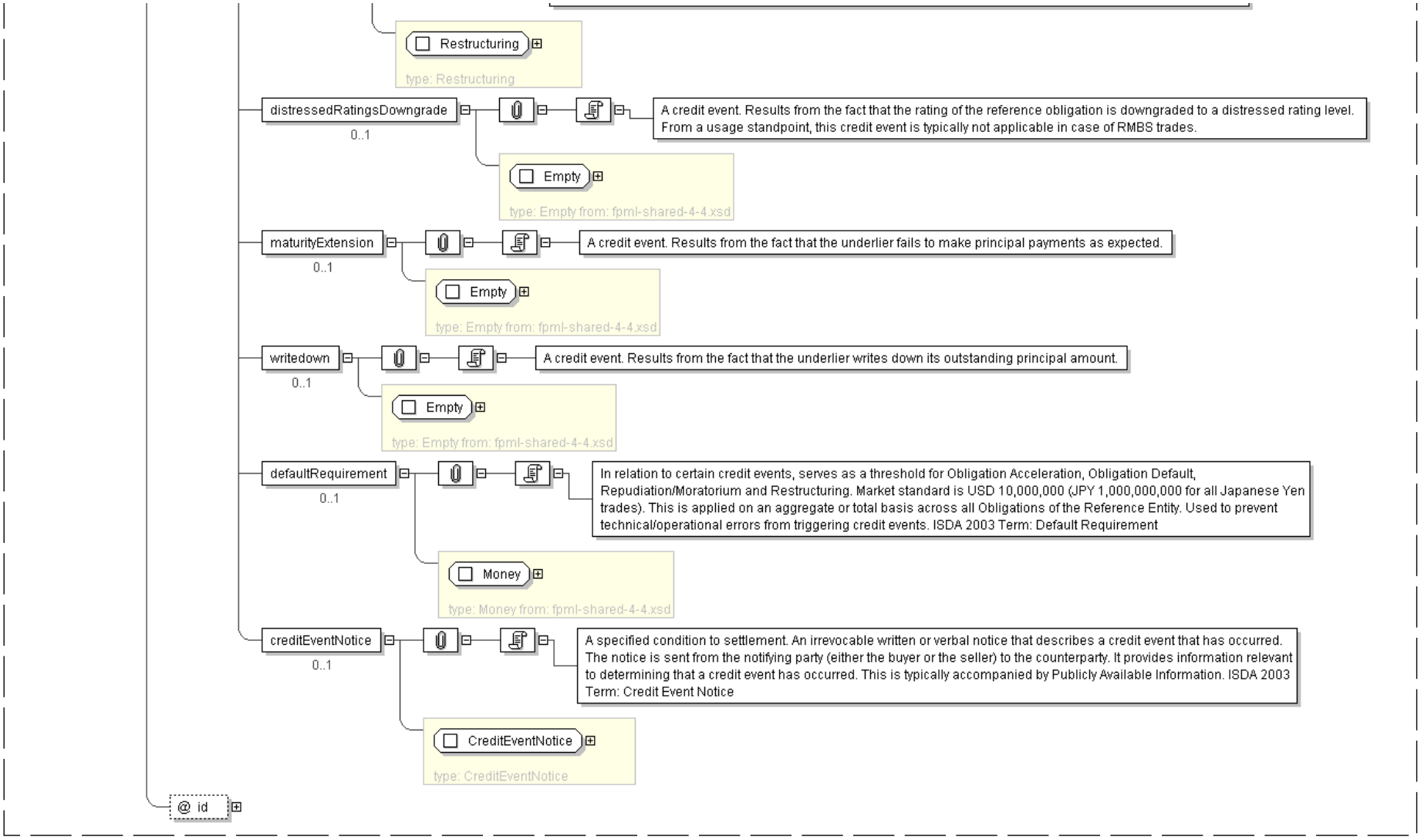
<creditEventNotice> CreditEventNotice </creditEventNotice> [0..1]

'A specified condition to settlement. An irrevocable written or verbal notice that describes a credit event that has occurred. The notice is sent from the notifying party (either the buyer or the seller) to the counterparty. It provides information relevant to determining that a credit event has occurred. This is typically accompanied by Publicly Available Information. ISDA 2003 Term: Credit Event Notice'

```
</...>
```

Diagram


CreditEvents
bankruptcy
0..1
A credit event. The reference entity has been dissolved or has become insolvent. It also covers events that may be a precursor to insolvency such as instigation of bankruptcy or insolvency proceedings. Sovereign trades are not subject to Bankruptcy as "technically" a Sovereign cannot become bankrupt. ISDA 2003 Term: Bankruptcy
Empty
type: Empty from: fpml-shared-4-4.xsd
failureToPay
0..1
A credit event. This credit event triggers, after the expiration of any applicable grace period, if the reference entity fails to make due payments in an aggregrate amount of not less than the payment requirement on one or more obligations (e.g. a missed coupon payment). ISDA 2003 Term: Failure to Pay
FailureToPay
type: FailureToPay
failureToPayPrincipal
0..1
A credit event. Corresponds to the failure by the Reference Entity to pay an expected principal amount or the payment of an actual principal amount that is less than the expected principal amount. ISDA 2003 Term: Failure to Pay Principal.
Empty
type: Empty from: fpml-shared-4-4.xsd
failureToPayInterest
0..1
A credit event. Corresponds to the failure by the Reference Entity to pay an expected interest amount or the payment of an actual interest amount that is less than the expected interest amount. ISDA 2003 Term: Failure to Pay Interest.
Empty
type: Empty from: fpml-shared-4-4.xsd
obligationDefault
0..1
A credit event. One or more of the obligations have become capable of being declared due and payable before they would otherwise have been due and payable as a result of, or on the basis of, the occurrence of a default, event of default or other similar condition or event other than failure to pay. ISDA 2003 Term: Obligation Default
Empty
type: Empty from: fpml-shared-4-4.xsd
obligationAcceleration
0..1
A credit event. One or more of the obligations have been declared due and payable before they would otherwise have been due and payable as a result of, or on the basis of, the occurrence of a default, event of default or other similar condition or event other than failure to pay (preferred by the market over Obligation Default, because more definitive and encompasses the definition of Obligation Default - this is more favorable to the Seller). Subject to the default requirement amount. ISDA 2003 Term: Obligation Acceleration
Empty
type: Empty from: fpml-shared-4-4.xsd
repudiationMoratorium
0..1
A credit event. The reference entity, or a governmental authority, either refuses to recognise or challenges the validity of one or more obligations of the reference entity, or imposes a moratorium thereby postponing payments on one or more of the obligations of the reference entity. Subject to the default requirement amount. ISDA 2003 Term: Repudiation/Moratorium
Empty
type: Empty from: fpml-shared-4-4.xsd
restructuring
0..1
A credit event. A restructuring is an event that materially impacts the reference entity's obligations, such as an interest rate reduction, principal reduction, deferral of interest or principal, change in priority ranking, or change in currency or composition of payment. ISDA 2003 Term: Restructuring



Schema Component Representation

```
<xsd:complexType name="CreditEvents">
   <xsd:sequence>
      <xsd:element name="bankruptcy" type=" Empty " minOccurs="0"/>
      <xsd:element name="failureToPay" type=" FailureToPay " minOccurs="0"/>
      <xsd:element name="failureToPayPrincipal" type=" Empty " minOccurs="0"/>
      <xsd:element name="failureToPayInterest" type=" Empty " minOccurs="0"/>
      <xsd:element name="obligationDefault" type=" Empty " minOccurs="0"/>
      <xsd:element name="obligationAcceleration" type=" Empty " minOccurs="0"/>
      <xsd:element name="repudiationMoratorium" type=" Empty " minOccurs="0"/>
      <xsd:element name="restructuring" type=" Restructuring " minOccurs="0"/>
      <xsd:element name="distressedRatingsDowngrade" type=" Empty " minOccurs="0"/>
      <xsd:element name="maturityExtension" type=" Empty " minOccurs="0"/>
      <xsd:element name="writedown" type=" Empty " minOccurs="0"/>
      <xsd:element name="defaultRequirement" type=" Money " minOccurs="0"/>
      <xsd:element name="creditEventNotice" type=" CreditEventNotice " minOccurs="0"/>
   </xsd:sequence>
   <xsd:attribute name="id" type=" xsd:ID " use="optional"/>
```

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

top

Complex Type: CreditEventsReference

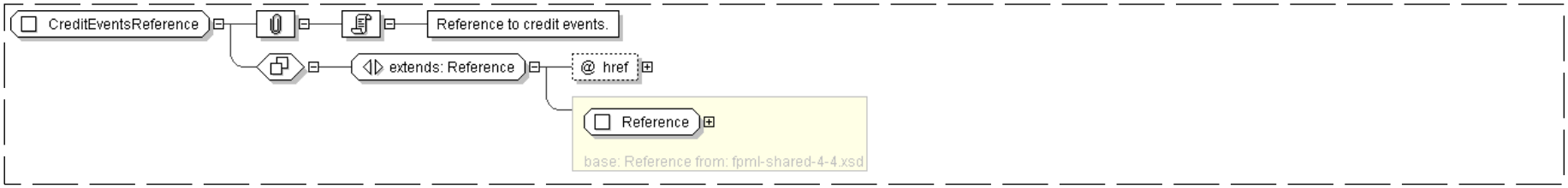
Super-types:	Reference < **CreditEventsReference** (by extension)
Sub-types:	None

Name	CreditEventsReference
Used by (from the same schema document)	Complex Type **Trigger**
Abstract	no
Documentation	Reference to credit events.

XML Instance Representation

```
<...
href=" xsd:IDREF [1]"/>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="CreditEventsReference">
   <xsd:complexContent>
      <xsd:extension base=" Reference ">
         <xsd:attribute name="href" type=" xsd:IDREF " use="required" reference="CreditEvents"/>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: FailureToPay

Super-types:	None
Sub-types:	None

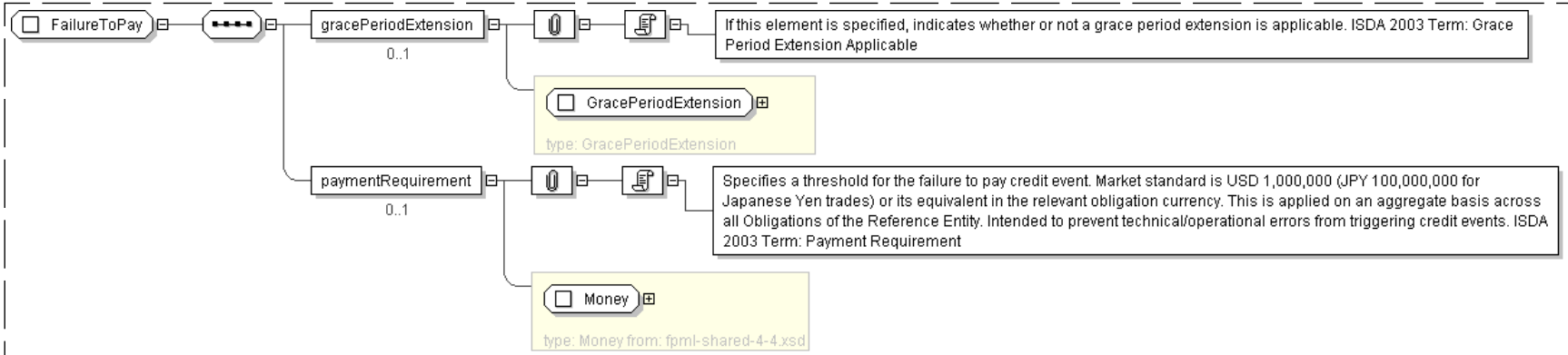
Name	FailureToPay
Used by (from the same schema document)	Complex Type **CreditEvents**
Abstract	no

XML Instance Representation

```
<...>
   <gracePeriodExtension> GracePeriodExtension </gracePeriodExtension> [0..1]
   'If this element is specified, indicates whether or not a grace period extension is
   applicable. ISDA 2003 Term: Grace Period Extension Applicable'

   <paymentRequirement> Money </paymentRequirement> [0..1]
   'Specifies a threshold for the failure to pay credit event. Market standard is USD
   1,000,000 (JPY 100,000,000 for Japanese Yen trades) or its equivalent in the
   relevant obligation currency. This is applied on an aggregate basis across all Obligations
```

```
   of the Reference Entity. Intended to prevent technical/operational errors from
   triggering credit events. ISDA 2003 Term: Payment Requirement'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="FailureToPay">
   <xsd:sequence>
      <xsd:element name="gracePeriodExtension" type=" GracePeriodExtension " minOccurs="0"/>
      <xsd:element name="paymentRequirement" type=" Money " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: FeaturePayment

Super-types:	None
Sub-types:	None

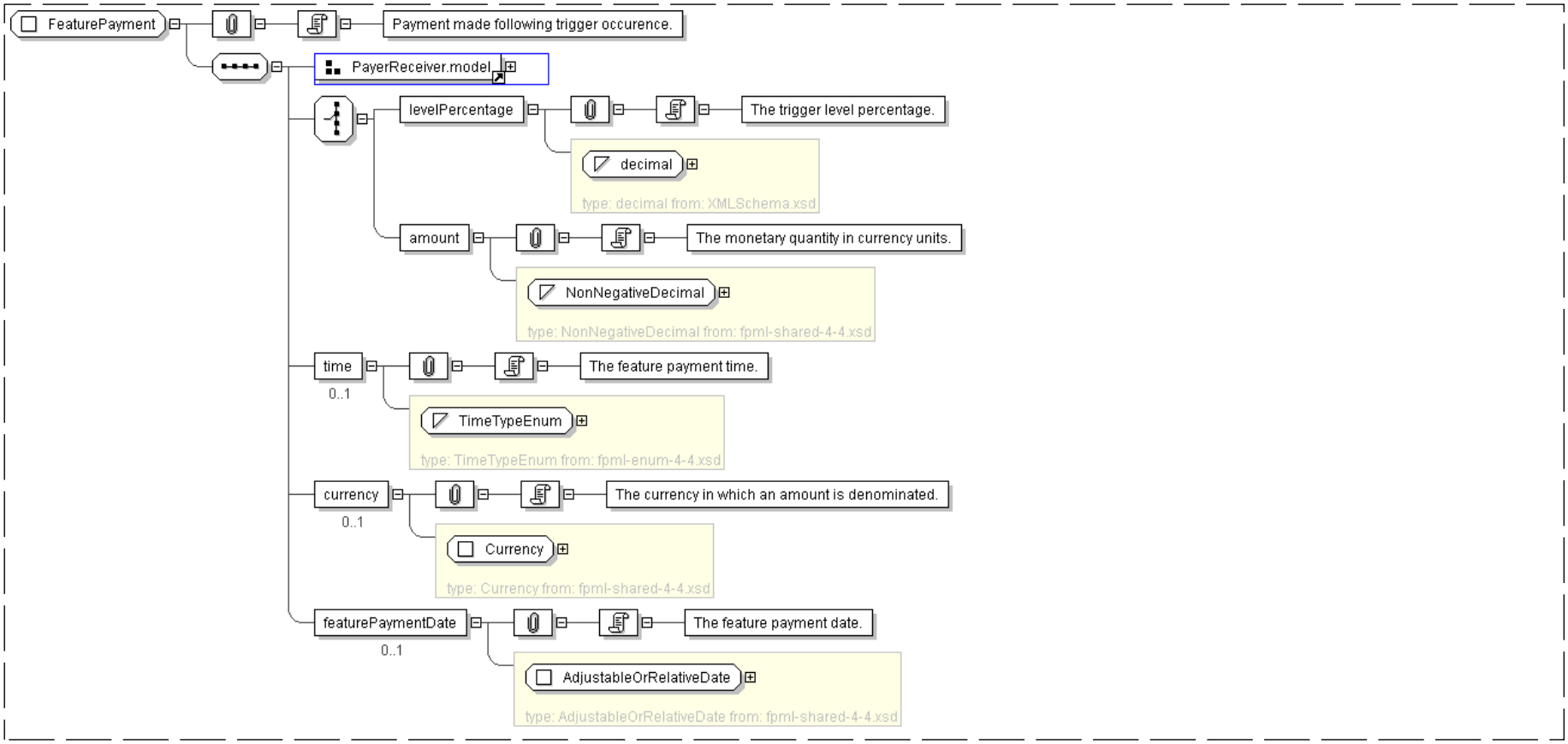
Name	FeaturePayment
Used by (from the same schema document)	Complex Type **TriggerEvent**
Abstract	no
Documentation	Payment made following trigger occurence.

XML Instance Representation

```
<...>
   <payerPartyReference> PartyOrAccountReference </payerPartyReference> [1]
   'A reference to the party responsible for making the payments defined by this structure.'

   <receiverPartyReference> PartyOrAccountReference </receiverPartyReference> [1]
   'A reference to the party that receives the payments corresponding to this structure.'

Start Choice [1]
      <levelPercentage> xsd:decimal </levelPercentage> [1]
      'The trigger level percentage.'

      <amount> NonNegativeDecimal </amount> [1]
      'The monetary quantity in currency units.'

End Choice
   <time> TimeTypeEnum </time> [0..1]
```

```
    'The feature payment time.'

    <currency> Currency </currency> [0..1]
    'The currency in which an amount is denominated.'

    <featurePaymentDate> AdjustableOrRelativeDate </featurePaymentDate> [0..1]
    'The feature payment date.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="FeaturePayment">
   <xsd:sequence>
      <xsd:group ref=" PayerReceiver.model "/>
      <xsd:choice>
         <xsd:element name="levelPercentage" type=" xsd:decimal "/>
         <xsd:element name="amount" type=" NonNegativeDecimal "/>
      </xsd:choice>
      <xsd:element name="time" type=" TimeTypeEnum " minOccurs="0"/>
      <xsd:element name="currency" type=" Currency " minOccurs="0"/>
      <xsd:element name="featurePaymentDate" type=" AdjustableOrRelativeDate " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: FrequencyType

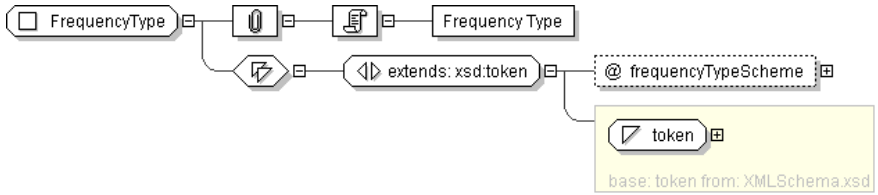
Super-types:	xsd:token < **FrequencyType** (by extension)
Sub-types:	None

Name	FrequencyType
Used by (from the same schema document)	Complex Type **AveragingSchedule**
Abstract	no
Documentation	Frequency Type

XML Instance Representation

```
<...
frequencyTypeScheme=" xsd:anyURI [0..1]">
xsd:token
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="FrequencyType">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:token ">
         <xsd:attribute name="frequencyTypeScheme" type=" xsd:anyURI "/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: FxFeature

Super-types:	None
Sub-types:	None

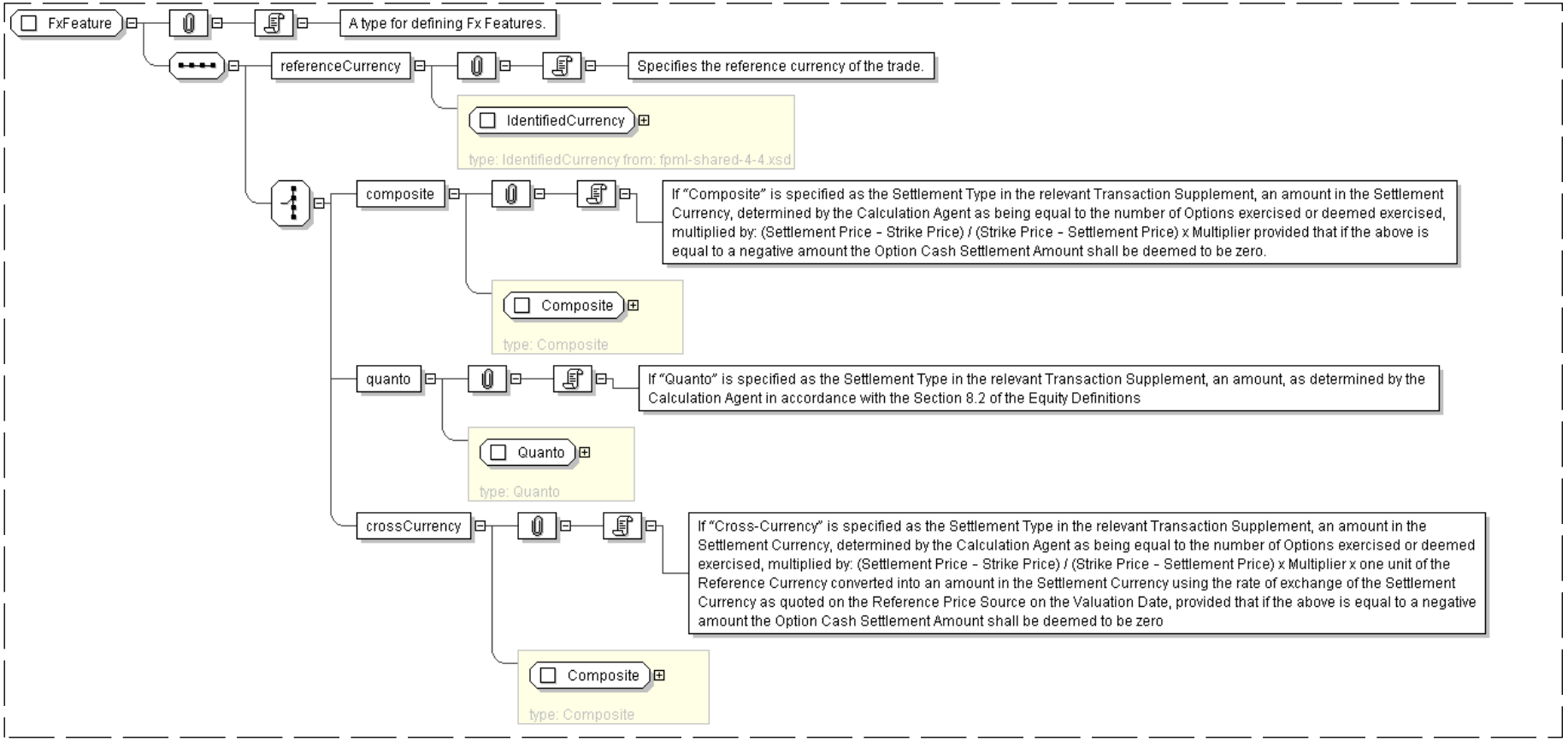
Name	FxFeature
Used by (from the same schema document)	Model Group **OptionBaseFeature.model**
Abstract	no
Documentation	A type for defining Fx Features.

XML Instance Representation

```
<...>
   <referenceCurrency> IdentifiedCurrency </referenceCurrency> [1]
   'Specifies the reference currency of the trade.'

Start Choice [1]
      <composite> Composite </composite> [1]
      'If "Composite" is specified as the Settlement Type in the relevant Transaction Supplement,
      an amount in the Settlement Currency, determined by the Calculation Agent as being equal to
      the number of Options exercised or deemed exercised, multiplied by: (Settlement Price –
      Strike Price) / (Strike Price – Settlement Price) x Multiplier provided that if the above
```

```
is equal to a negative amount the Option Cash Settlement Amount shall be deemed to be zero.'

<quanto> Quanto </quanto> [1]
'If "Quanto" is specified as the Settlement Type in the relevant Transaction Supplement,
an amount, as determined by the Calculation Agent in accordance with the Section 8.2 of
the Equity Definitions'

<crossCurrency> Composite </crossCurrency> [1]
'If "Cross-Currency" is specified as the Settlement Type in the relevant
Transaction Supplement, an amount in the Settlement Currency, determined by the
Calculation Agent as being equal to the number of Options exercised or deemed
exercised, multiplied by: (Settlement Price – Strike Price) / (Strike Price – Settlement
Price) x Multiplier x one unit of the Reference Currency converted into an amount in
the Settlement Currency using the rate of exchange of the Settlement Currency as quoted on
the Reference Price Source on the Valuation Date, provided that if the above is equal to
a negative amount the Option Cash Settlement Amount shall be deemed to be zero'

End Choice
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="FxFeature">
   <xsd:sequence>
      <xsd:element name="referenceCurrency" type=" IdentifiedCurrency "/>
      <xsd:choice>
         <xsd:element name="composite" type=" Composite "/>
         <xsd:element name="quanto" type=" Quanto "/>
```

```
        <xsd:element name="crossCurrency" type=" Composite "/>
      </xsd:choice>
    </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: GracePeriodExtension

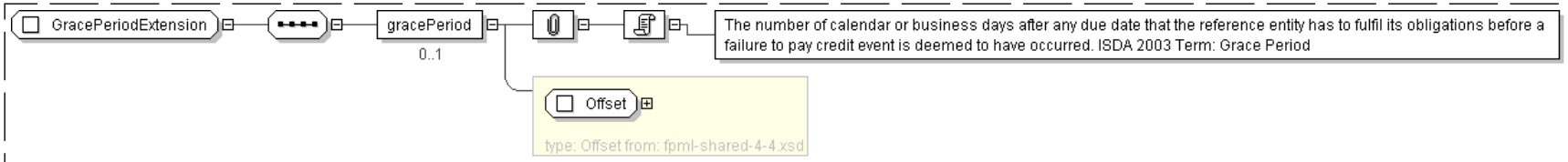
Super-types:	None
Sub-types:	None

Name	GracePeriodExtension
Used by (from the same schema document)	Complex Type **FailureToPay**
Abstract	no

XML Instance Representation

```
<...>
   <gracePeriod> Offset </gracePeriod> [0..1]
   'The number of calendar or business days after any due date that the reference entity has
   to fulfil its obligations before a failure to pay credit event is deemed to have occurred.
   ISDA 2003 Term: Grace Period'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="GracePeriodExtension">
   <xsd:sequence>
      <xsd:element name="gracePeriod" type=" Offset " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: Knock

Super-types:	None
Sub-types:	None

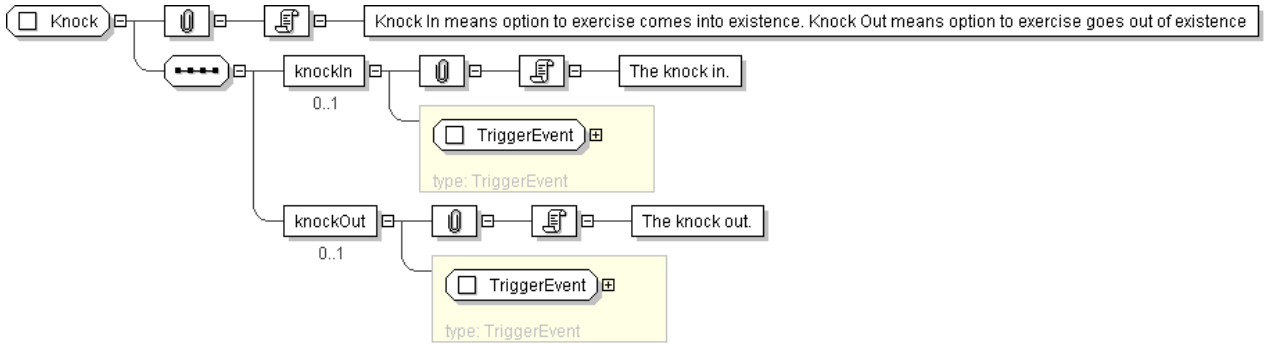
Name	Knock
Used by (from the same schema document)	Model Group **OptionFeature.model**
Abstract	no
Documentation	Knock In means option to exercise comes into existence. Knock Out means option to exercise goes out of existence

XML Instance Representation

```
<...>
   <knockIn> TriggerEvent </knockIn> [0..1]
   'The knock in.'
```

```
<knockOut> TriggerEvent </knockOut> [0..1]
'The knock out.'
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Knock">
   <xsd:sequence>
      <xsd:element name="knockIn" type=" TriggerEvent " minOccurs="0"/>
      <xsd:element name="knockOut" type=" TriggerEvent " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: MarketDisruption

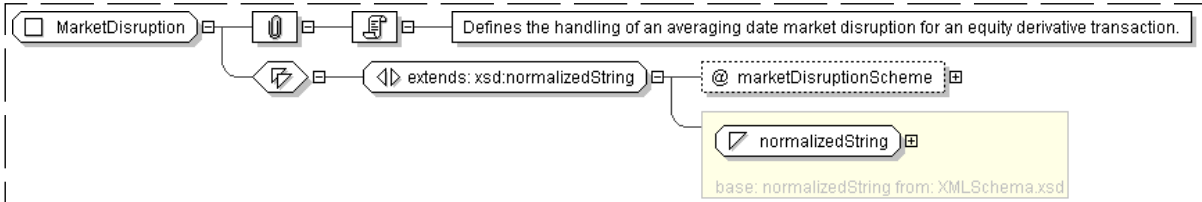
Super-types:	xsd:normalizedString < **MarketDisruption** (by extension)
Sub-types:	None

Name	MarketDisruption
Used by (from the same schema document)	Complex Type **AveragingPeriod**
Abstract	no
Documentation	Defines the handling of an averaging date market disruption for an equity derivative transaction.

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="MarketDisruption">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="marketDisruptionScheme" type=" xsd:anyURI " default="http://www.fpml.
         org/coding-scheme/market-disruption-1-0"/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: NotifyingParty

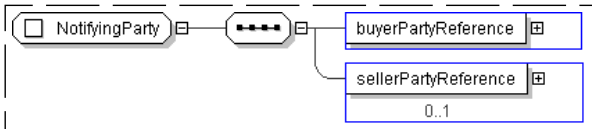
Super-types:	None
Sub-types:	None

Name	NotifyingParty
Used by (from the same schema document)	Complex Type **CreditEventNotice**
Abstract	no

XML Instance Representation

```
<...>
   <buyerPartyReference> PartyReference </buyerPartyReference> [1]
   <sellerPartyReference> PartyReference </sellerPartyReference> [0..1]
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="NotifyingParty">
   <xsd:sequence>
      <xsd:element name="buyerPartyReference" type=" PartyReference "/>
      <xsd:element name="sellerPartyReference" type=" PartyReference " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: OptionBase

Super-types:	Product < **OptionBase** (by extension)
Sub-types:	• OptionBaseExtended (by extension)

Name	OptionBase
Abstract	yes
Documentation	A type for defining the common features of options

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
   <productType> ProductType </productType> [0..*]
```

'A classification of the type of product. FpML defines a simple product categorization using a coding scheme.'

```
<productId> ProductId </productId> [0..*]
```

'A product reference identifier allocated by a party. FpML does not define the domain values associated with this element. Note that the domain values for this element are not strictly an enumerated list.'

```
<buyerPartyReference> PartyOrTradeSideReference </buyerPartyReference> [1]
```

'A reference to the party that buys this instrument, ie. pays for this instrument and receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case of FRAs this the fixed rate payer.'

```
<sellerPartyReference> PartyOrTradeSideReference </sellerPartyReference> [1]
```

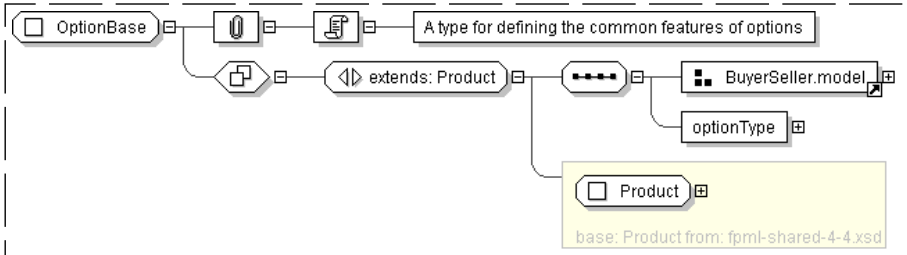
'A reference to the party that sells (\"writes\") this instrument, i.e. that grants the rights defined by this instrument and in return receives a payment for it. See 2000 ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'

```
<optionType> OptionTypeEnum </optionType> [1]
```

'The type of option transaction. From a usage standpoint, put/call is the default option type, while payer/receiver indicator is used for options index credit default swaps, consistently with the industry practice. Straddle is used for the case of straddle strategy, that combine a call and a put with the same strike.'

```
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="OptionBase" abstract="true">
   <xsd:complexContent>
      <xsd:extension base=" Product ">
         <xsd:sequence>
            <xsd:group ref=" BuyerSeller.model "/>
            <xsd:element name="optionType" type=" OptionTypeEnum "/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: OptionBaseExtended

Super-types:	Product < OptionBase (by extension) < **OptionBaseExtended** (by extension)
Sub-types:	None

Name	OptionBaseExtended
Abstract	yes
Documentation	Base type for options starting with the 4-3 release, until we refactor the schema as part of the 5-0 release series

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
```

<productType> ProductType </productType> [0..*]

'A classification of the type of product. FpML defines a simple product categorization using a coding scheme.'

<productId> ProductId </productId> [0..*]

'A product reference identifier allocated by a party. FpML does not define the domain values associated with this element. Note that the domain values for this element are not strictly an enumerated list.'

<buyerPartyReference> PartyOrTradeSideReference </buyerPartyReference> [1]

'A reference to the party that buys this instrument, ie. pays for this instrument and receives the rights defined by it. See 2000 ISDA definitions Article 11.1 (b). In the case of FRAs this the fixed rate payer.'

<sellerPartyReference> PartyOrTradeSideReference </sellerPartyReference> [1]

'A reference to the party that sells (\"writes\") this instrument, i.e. that grants the rights defined by this instrument and in return receives a payment for it. See 2000 ISDA definitions Article 11.1 (a). In the case of FRAs this is the floating rate payer.'

<optionType> OptionTypeEnum </optionType> [1]

'The type of option transaction. From a usage standpoint, put/call is the default option type, while payer/receiver indicator is used for options index credit default swaps, consistently with the industry practice. Straddle is used for the case of straddle strategy, that combine a call and a put with the same strike.'

<premium> Premium </premium> [0..1]

'The option premium payable by the buyer to the seller'

<exercise> ... </exercise> [1]
<exerciseProcedure> ExerciseProcedure </exerciseProcedure> [1]

'A set of parameters defining procedures associated with the exercise.'

<feature> OptionFeature </feature> [0..1]

'An Option feature such as quanto, asian, barrier, knock'

Start Choice [0..1]

'A choice between an explicit representation of the notional amount, or a reference to a notional amount defined elsewhere in this document'

<notionalReference> NotionalAmountReference </notionalReference> [1]
<notionalAmount> Money </notionalAmount> [1]

End Choice
Start Group: OptionDenomination.model [0..1]

<optionEntitlement> PositiveDecimal </optionEntitlement> [1]

'The number of units of underlyer per option comprised in the option transaction.'

<entitlementCurrency> Currency </entitlementCurrency> [0..1]

'TODO'

<numberOfOptions> PositiveDecimal </numberOfOptions> [0..1]

'The number of options comprised in the option transaction.'

End Group: OptionDenomination.model

<settlementType> SettlementTypeEnum </settlementType> [1]
<settlementDate> AdjustableOrRelativeDate </settlementDate> [0..1]

Start Group: SettlementAmountOrCurrency.model [0..1]
Start Choice [1]

<settlementAmount> Money </settlementAmount> [1]

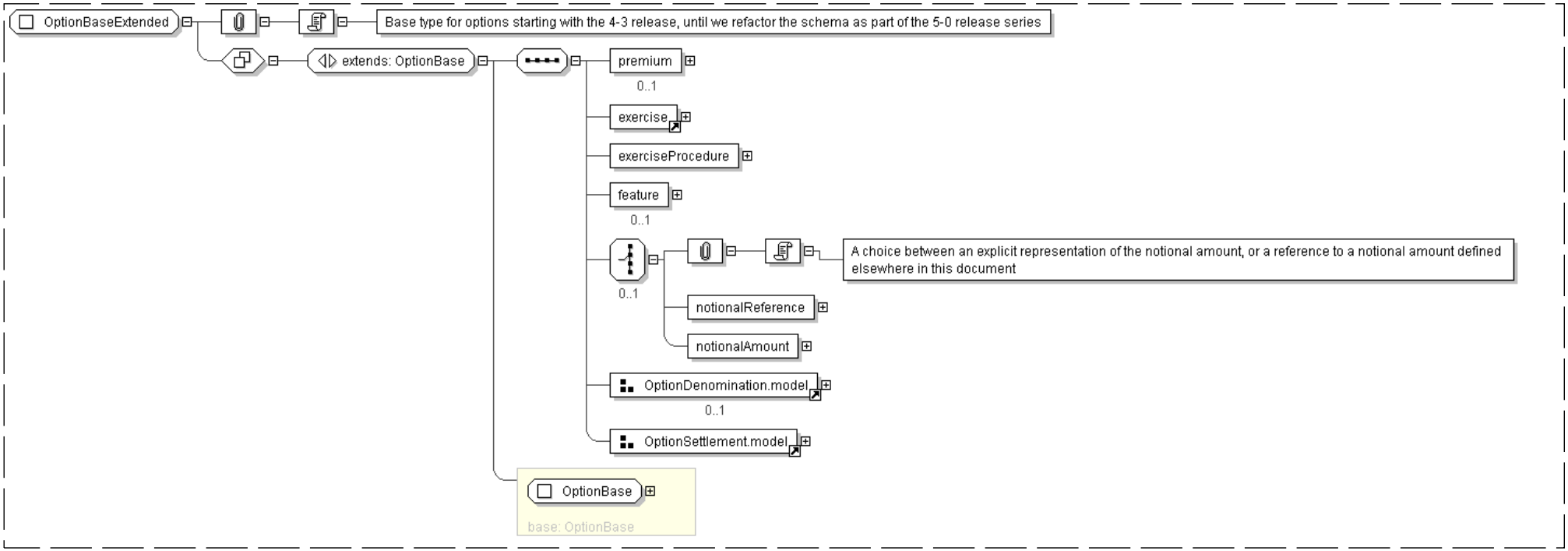
'Settlement Amount'

<settlementCurrency> Currency </settlementCurrency> [1]

'Settlement Currency for use where the Settlement Amount cannot be known in advance'

```
End Choice
End Group: SettlementAmountOrCurrency.model
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="OptionBaseExtended" abstract="true">
   <xsd:complexContent>
      <xsd:extension base=" OptionBase ">
         <xsd:sequence>
            <xsd:element name="premium" type=" Premium " minOccurs="0"/>
            <xsd:element ref=" exercise "/>
            <xsd:element name="exerciseProcedure" type=" ExerciseProcedure "/>
            <xsd:element name="feature" type=" OptionFeature " minOccurs="0"/>
            <xsd:choice minOccurs="0">
               <xsd:element name="notionalReference" type=" NotionalAmountReference "/>
               <xsd:element name="notionalAmount" type=" Money "/>
            </xsd:choice>
            <xsd:group ref=" OptionDenomination.model " minOccurs="0"/>
            <xsd:group ref=" OptionSettlement.model "/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: OptionFeature

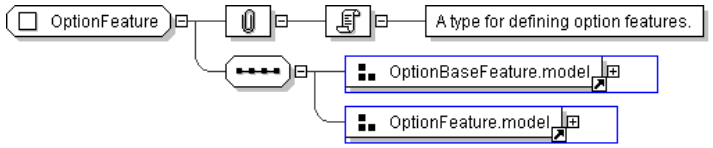
Super-types:	None
Sub-types:	None

Name	OptionFeature
Used by (from the same schema document)	Complex Type **OptionBaseExtended**
Abstract	no
Documentation	A type for defining option features.

XML Instance Representation

```
<...>
   <fxFeature> FxFeature </fxFeature> [0..1]
   'A quanto or composite FX feature.'

   <strategyFeature> StrategyFeature </strategyFeature> [0..1]
   'A simple strategy feature'

   <asian> Asian </asian> [0..1]
   'An option where and average price is taken on valuation.'

   <barrier> Barrier </barrier> [0..1]
   'An option with a barrier feature.'

   <knock> Knock </knock> [0..1]
   'A knock feature.'

   <passThrough> PassThrough </passThrough> [0..1]
   'Pass through payments from the underlyer, such as dividends.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="OptionFeature">
   <xsd:sequence>
      <xsd:group ref=" OptionBaseFeature.model "/>
      <xsd:group ref=" OptionFeature.model "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: OptionNumericStrike

Super-types:	None
Sub-types:	• OptionStrike (by extension)

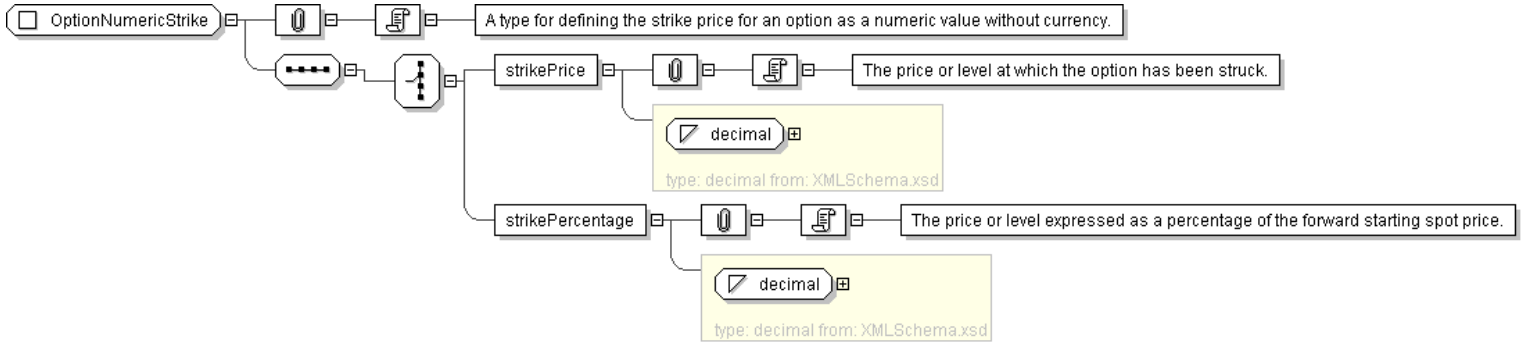
Name	OptionNumericStrike
Abstract	no
Documentation	A type for defining the strike price for an option as a numeric value without currency.

XML Instance Representation

```
<...>
Start Choice [1]
      <strikePrice> xsd:decimal </strikePrice> [1]
```

```
        'The price or level at which the option has been struck.'

        <strikePercentage> xsd:decimal </strikePercentage> [1]
        'The price or level expressed as a percentage of the forward starting spot price.'

End Choice
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="OptionNumericStrike">
   <xsd:sequence>
      <xsd:choice>
         <xsd:element name="strikePrice" type=" xsd:decimal "/>
         <xsd:element name="strikePercentage" type=" xsd:decimal "/>
      </xsd:choice>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: OptionStrike

Super-types:	OptionNumericStrike < **OptionStrike** (by extension)
Sub-types:	None

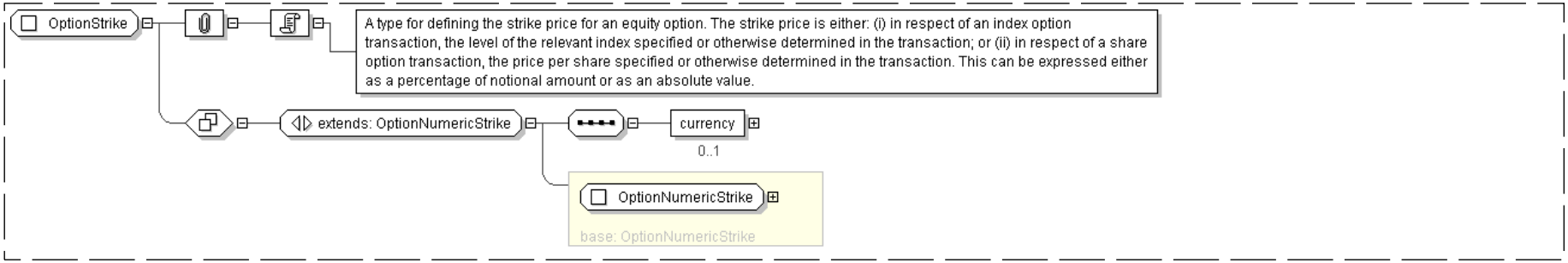
Name	OptionStrike
Used by (from the same schema document)	Complex Type **StrikeSpread**
Abstract	no
Documentation	A type for defining the strike price for an equity option. The strike price is either: (i) in respect of an index option transaction, the level of the relevant index specified or otherwise determined in the transaction; or (ii) in respect of a share option transaction, the price per share specified or otherwise determined in the transaction. This can be expressed either as a percentage of notional amount or as an absolute value.

XML Instance Representation

```
<...>
Start Choice [1]
      <strikePrice> xsd:decimal </strikePrice> [1]
      'The price or level at which the option has been struck.'

      <strikePercentage> xsd:decimal </strikePercentage> [1]
      'The price or level expressed as a percentage of the forward starting spot price.'

End Choice
   <currency> Currency </currency> [0..1]
   'The currency in which an amount is denominated.'
```

```
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="OptionStrike">
   <xsd:complexContent>
      <xsd:extension base=" OptionNumericStrike ">
         <xsd:sequence>
            <xsd:element name="currency" type=" Currency " minOccurs="0"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: PassThrough

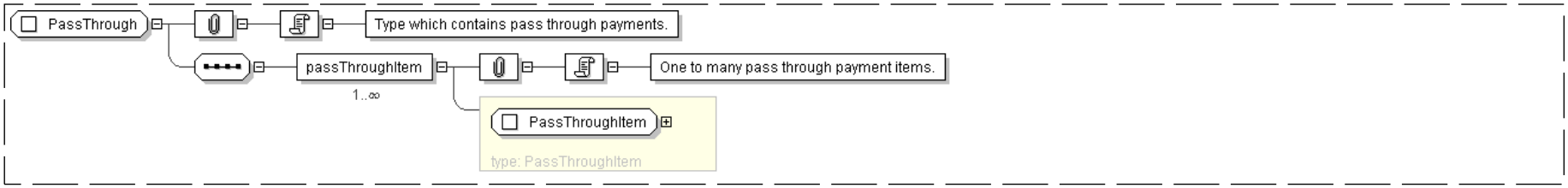
Super-types:	None
Sub-types:	None

Name	PassThrough
Used by (from the same schema document)	Model Group **OptionFeature.model**
Abstract	no
Documentation	Type which contains pass through payments.

XML Instance Representation

```
<...>
   <passThroughItem> PassThroughItem </passThroughItem> [1..*]
   'One to many pass through payment items.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="PassThrough">
   <xsd:sequence>
```

```
      <xsd:element name="passThroughItem" type=" PassThroughItem " maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: PassThroughItem

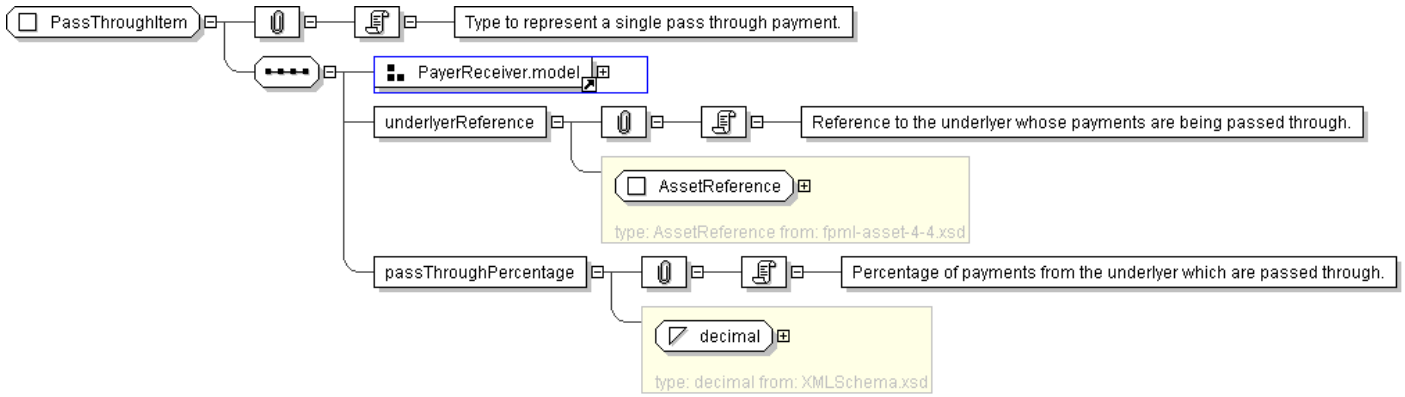
Super-types:	None
Sub-types:	None

Name	PassThroughItem
Used by (from the same schema document)	Complex Type **PassThrough**
Abstract	no
Documentation	Type to represent a single pass through payment.

XML Instance Representation

```
<...>
   <payerPartyReference> PartyOrAccountReference </payerPartyReference> [1]
   'A reference to the party responsible for making the payments defined by this structure.'

   <receiverPartyReference> PartyOrAccountReference </receiverPartyReference> [1]
   'A reference to the party that receives the payments corresponding to this structure.'

   <underlyerReference> AssetReference </underlyerReference> [1]
   'Reference to the underlyer whose payments are being passed through.'

   <passThroughPercentage> xsd:decimal </passThroughPercentage> [1]
   'Percentage of payments from the underlyer which are passed through.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="PassThroughItem">
   <xsd:sequence>
      <xsd:group ref=" PayerReceiver.model "/>
      <xsd:element name="underlyerReference" type=" AssetReference "/>
      <xsd:element name="passThroughPercentage" type=" xsd:decimal "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: Premium

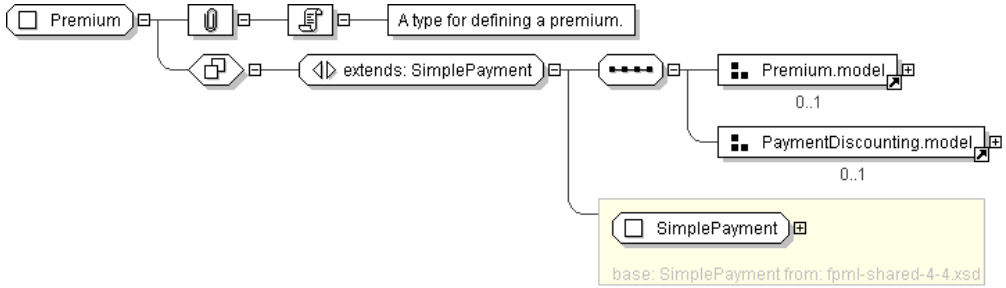
Super-types:	SimplePayment < **Premium** (by extension)
Sub-types:	None

Name	Premium
Used by (from the same schema document)	Complex Type **OptionBaseExtended**
Abstract	no
Documentation	A type for defining a premium.

XML Instance Representation

```
<...>
   <payerPartyReference> PartyOrAccountReference </payerPartyReference> [1]
   'A reference to the party responsible for making the payments defined by this structure.'

   <receiverPartyReference> PartyOrAccountReference </receiverPartyReference> [1]
   'A reference to the party that receives the payments corresponding to this structure.'

   <paymentAmount> Money </paymentAmount> [1]
   <paymentDate> AdjustableOrRelativeAndAdjustedDate </paymentDate> [1]
   'The payment date. This date is subject to adjustment in accordance with any
   applicable business day convention.'

Start Group: Premium.model [0..1]
      <premiumType> PremiumTypeEnum </premiumType> [0..1]
      'Forward start Premium type'

      <pricePerOption> Money </pricePerOption> [0..1]
      'The amount of premium to be paid expressed as a function of the number of options.'

      <percentageOfNotional> xsd:decimal </percentageOfNotional> [0..1]
      'The amount of premium to be paid expressed as a percentage of the notional value of
      the transaction. A percentage of 5% would be expressed as 0.05.'

End Group: Premium.model
Start Group: PaymentDiscounting.model [0..1]
      <discountFactor> xsd:decimal </discountFactor> [0..1]
      'The value representing the discount factor used to calculate the present value of the
      cash flow.'

      <presentValueAmount> Money </presentValueAmount> [0..1]
      'The amount representing the present value of the forecast payment.'

End Group: PaymentDiscounting.model
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Premium">
   <xsd:complexContent>
      <xsd:extension base=" SimplePayment ">
         <xsd:sequence>
            <xsd:group ref=" Premium.model " minOccurs="0"/>
            <xsd:group ref=" PaymentDiscounting.model " minOccurs="0"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: PubliclyAvailableInformation

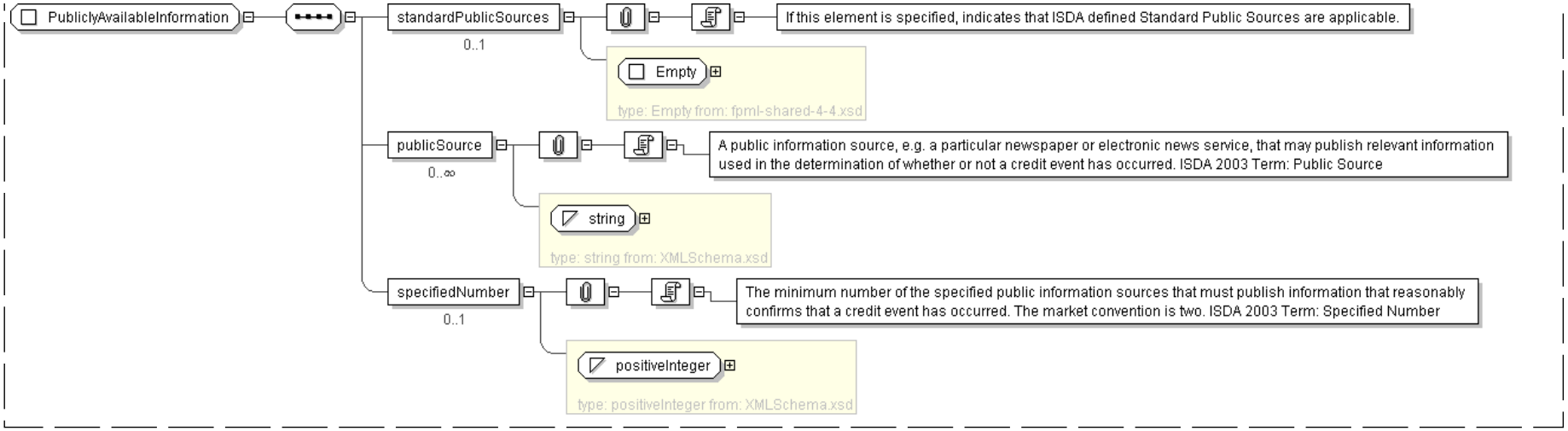
Super-types:	None
Sub-types:	None

Name	PubliclyAvailableInformation
Used by (from the same schema document)	Complex Type **CreditEventNotice**
Abstract	no

XML Instance Representation

```
<...>
   <standardPublicSources> Empty </standardPublicSources> [0..1]
   'If this element is specified, indicates that ISDA defined Standard Public Sources
   are applicable.'

   <publicSource> xsd:string </publicSource> [0..*]
   'A public information source, e.g. a particular newspaper or electronic news service, that
   may publish relevant information used in the determination of whether or not a credit event
   has occurred. ISDA 2003 Term: Public Source'

   <specifiedNumber> xsd:positiveInteger </specifiedNumber> [0..1]
   'The minimum number of the specified public information sources that must publish
   information that reasonably confirms that a credit event has occurred. The market convention
   is two. ISDA 2003 Term: Specified Number'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="PubliclyAvailableInformation">
   <xsd:sequence>
      <xsd:element name="standardPublicSources" type=" Empty " minOccurs="0"/>
      <xsd:element name="publicSource" type=" xsd:string " minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="specifiedNumber" type=" xsd:positiveInteger " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: Quanto

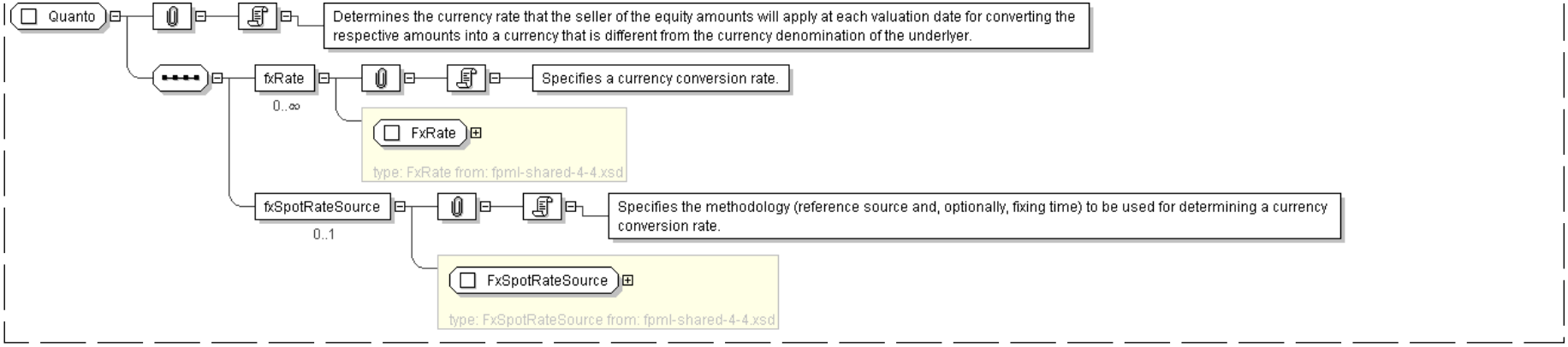
Super-types:	None
Sub-types:	None

Name	Quanto
Used by (from the same schema document)	Complex Type **FxFeature**
Abstract	no
Documentation	Determines the currency rate that the seller of the equity amounts will apply at each valuation date for converting the respective amounts into a currency that is different from the currency denomination of the underlyer.

XML Instance Representation

```
<...>
   <fxRate> FxRate </fxRate> [0..*]
   'Specifies a currency conversion rate.'

   <fxSpotRateSource> FxSpotRateSource </fxSpotRateSource> [0..1]
   'Specifies the methodology (reference source and, optionally, fixing time) to be used
   for determining a currency conversion rate.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Quanto">
   <xsd:sequence>
      <xsd:element name="fxRate" type=" FxRate " minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="fxSpotRateSource" type=" FxSpotRateSource " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: Restructuring

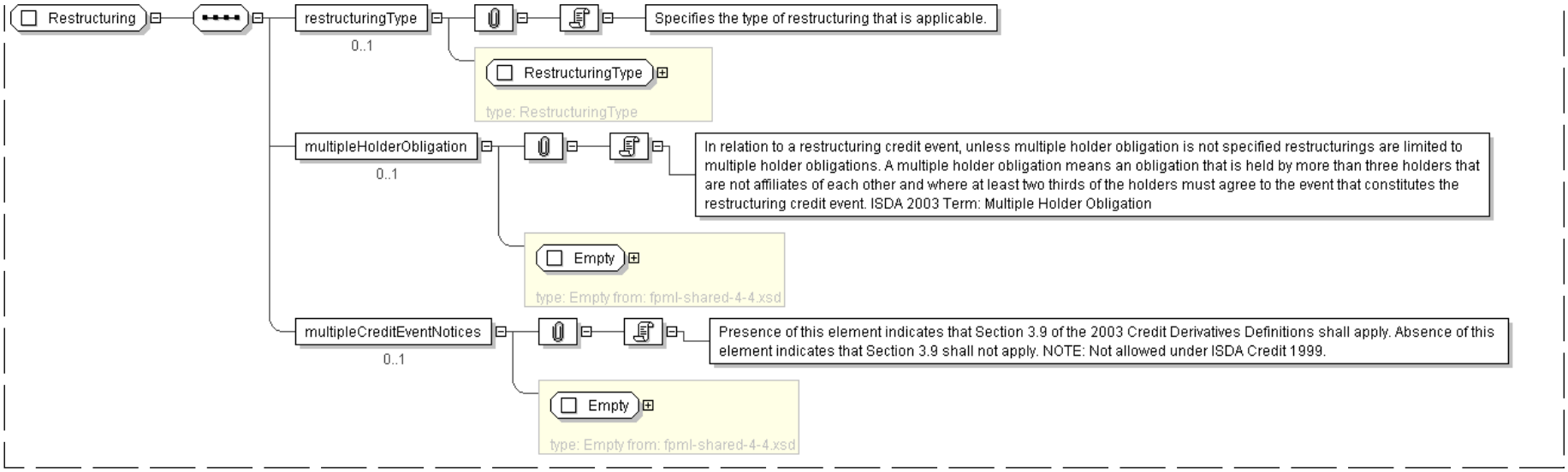
Super-types:	None
Sub-types:	None

Name	Restructuring
Used by (from the same schema document)	Complex Type **CreditEvents**
Abstract	no

XML Instance Representation

```
<...>
   <restructuringType> RestructuringType </restructuringType> [0..1]
   'Specifies the type of restructuring that is applicable.'

   <multipleHolderObligation> Empty </multipleHolderObligation> [0..1]
   'In relation to a restructuring credit event, unless multiple holder obligation is
   not specified restructurings are limited to multiple holder obligations. A multiple
   holder obligation means an obligation that is held by more than three holders that are
   not affiliates of each other and where at least two thirds of the holders must agree to
   the event that constitutes the restructuring credit event. ISDA 2003 Term: Multiple
   Holder Obligation'

   <multipleCreditEventNotices> Empty </multipleCreditEventNotices> [0..1]
   'Presence of this element indicates that Section 3.9 of the 2003 Credit Derivatives
   Definitions shall apply. Absence of this element indicates that Section 3.9 shall not
   apply. NOTE: Not allowed under ISDA Credit 1999.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Restructuring">
   <xsd:sequence>
      <xsd:element name="restructuringType" type=" RestructuringType " minOccurs="0"/>
      <xsd:element name="multipleHolderObligation" type=" Empty " minOccurs="0"/>
      <xsd:element name="multipleCreditEventNotices" type=" Empty " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: RestructuringType

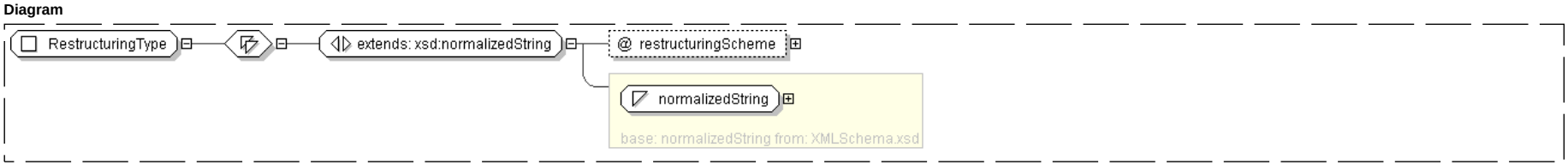
Super-types:	xsd:normalizedString < **RestructuringType** (by extension)
Sub-types:	None

Name	RestructuringType
Used by (from the same schema document)	Complex Type **Restructuring**
Abstract	no

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="RestructuringType">
   <xsd:simpleContent>
```

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

top

Complex Type: StrategyFeature

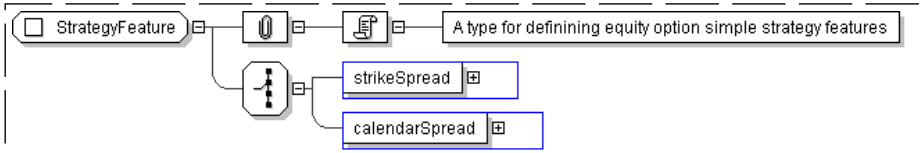
Super-types:	None
Sub-types:	None

Name	StrategyFeature
Used by (from the same schema document)	Model Group **OptionBaseFeature.model**
Abstract	no
Documentation	A type for definining equity option simple strategy features

XML Instance Representation

```
<...>
Start Choice [1]
      <strikeSpread> StrikeSpread </strikeSpread> [1]
      <calendarSpread> CalendarSpread </calendarSpread> [1]
End Choice
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="StrategyFeature">
   <xsd:choice>
      <xsd:element name="strikeSpread" type=" StrikeSpread "/>
      <xsd:element name="calendarSpread" type=" CalendarSpread "/>
   </xsd:choice>
</xsd:complexType>
```

top

Complex Type: StrikeSpread

Super-types:	None
Sub-types:	None

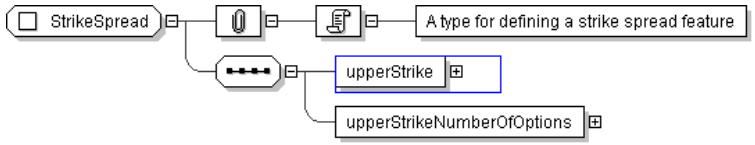
Name	StrikeSpread
Used by (from the same schema document)	Complex Type **StrategyFeature**
Abstract	no
Documentation	A type for defining a strike spread feature

XML Instance Representation

```
<...>
   <upperStrike> OptionStrike </upperStrike> [1]
   <upperStrikeNumberOfOptions> xsd:decimal </upperStrikeNumberOfOptions> [1]
```

```
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="StrikeSpread">
   <xsd:sequence>
      <xsd:element name="upperStrike" type=" OptionStrike "/>
      <xsd:element name="upperStrikeNumberOfOptions" type=" xsd:decimal "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: Trigger

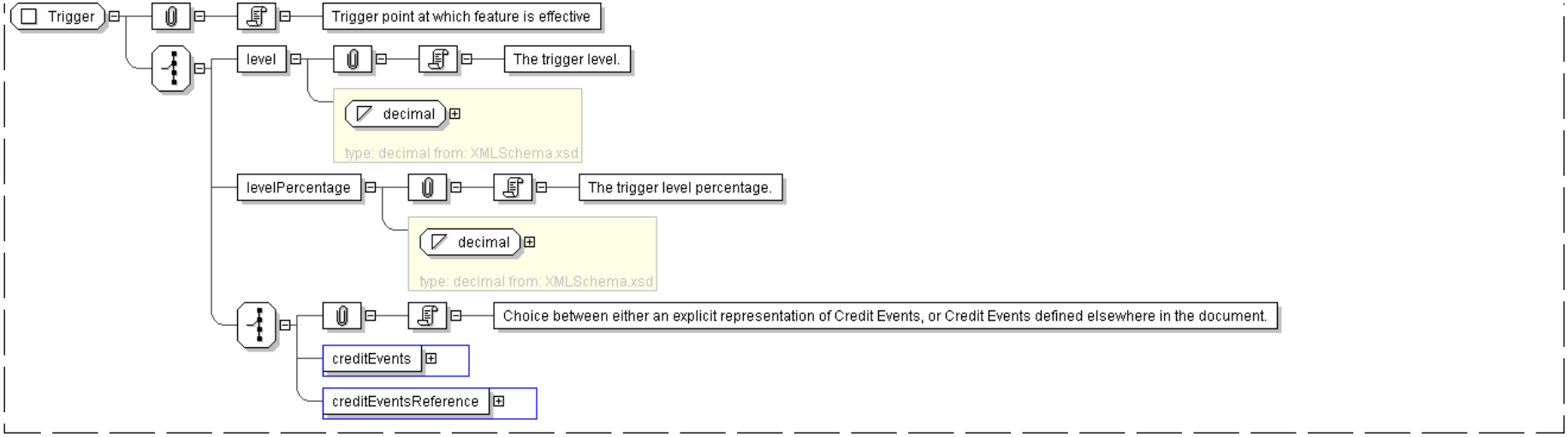
Super-types:	None
Sub-types:	None

Name	Trigger
Used by (from the same schema document)	Complex Type **TriggerEvent**
Abstract	no
Documentation	Trigger point at which feature is effective

XML Instance Representation

```
<...>
Start Choice [1]
      <level> xsd:decimal </level> [1]
      'The trigger level.'

      <levelPercentage> xsd:decimal </levelPercentage> [1]
      'The trigger level percentage.'

Start Choice [1]
'Choice between either an explicit representation of Credit Events, or Credit Events
defined elsewhere in the document.'

         <creditEvents> CreditEvents </creditEvents> [1]
         <creditEventsReference> CreditEventsReference </creditEventsReference> [1]
End Choice
End Choice
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Trigger">
   <xsd:choice>
      <xsd:element name="level" type=" xsd:decimal "/>
      <xsd:element name="levelPercentage" type=" xsd:decimal "/>
      <xsd:choice>
         <xsd:element name="creditEvents" type=" CreditEvents "/>
         <xsd:element name="creditEventsReference" type=" CreditEventsReference "/>
      </xsd:choice>
   </xsd:choice>
</xsd:complexType>
```

top

Complex Type: TriggerEvent

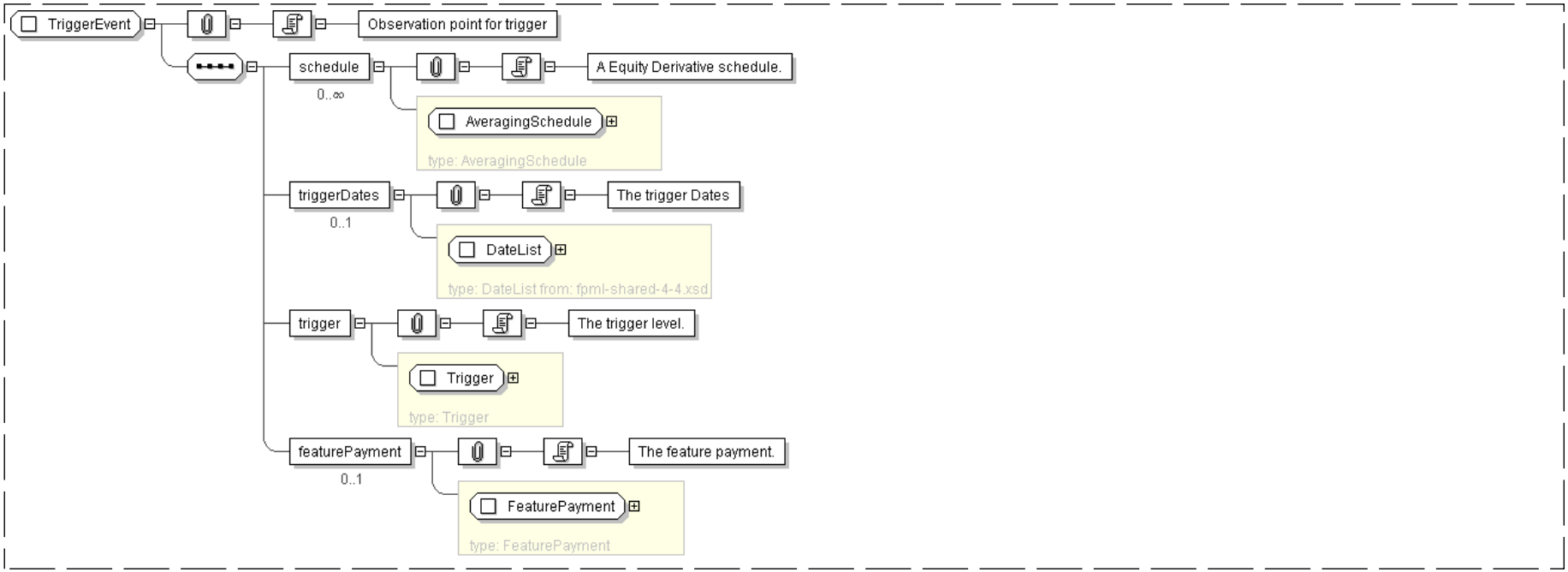
Super-types:	None
Sub-types:	None

Name	TriggerEvent
Used by (from the same schema document)	Complex Type **Barrier** , Complex Type **Barrier** , Complex Type **Knock** , Complex Type **Knock**
Abstract	no
Documentation	Observation point for trigger

XML Instance Representation

```
<...>
   <schedule> AveragingSchedule </schedule> [0..*]
   'A Equity Derivative schedule.'

   <triggerDates> DateList </triggerDates> [0..1]
   'The trigger Dates'

   <trigger> Trigger </trigger> [1]
   'The trigger level.'

   <featurePayment> FeaturePayment </featurePayment> [0..1]
   'The feature payment.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="TriggerEvent">
   <xsd:sequence>
      <xsd:element name="schedule" type=" AveragingSchedule " minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="triggerDates" type=" DateList " minOccurs="0"/>
      <xsd:element name="trigger" type=" Trigger "/>
      <xsd:element name="featurePayment" type=" FeaturePayment " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

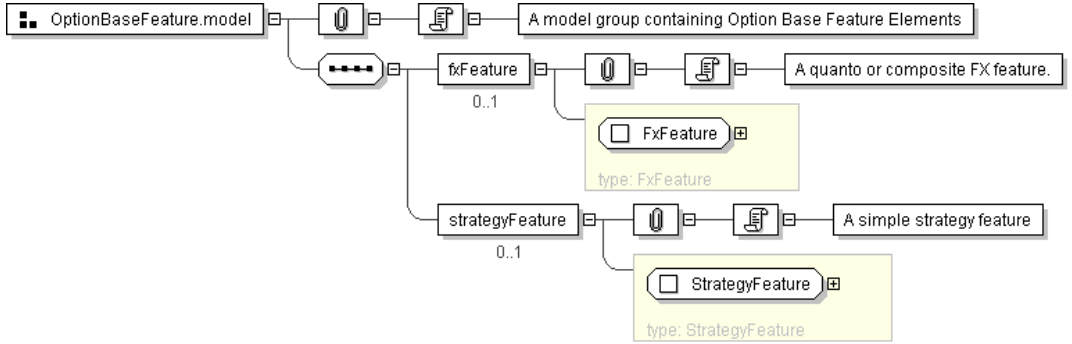
Model Group: OptionBaseFeature.model

Name	OptionBaseFeature.model
Used by (from the same schema document)	Complex Type **OptionFeature**
Documentation	A model group containing Option Base Feature Elements

XML Instance Representation

```
<fxFeature> FxFeature </fxFeature> [0..1]
'A quanto or composite FX feature.'

<strategyFeature> StrategyFeature </strategyFeature> [0..1]
'A simple strategy feature'
```

Diagram



Schema Component Representation

```
<xsd:group name="OptionBaseFeature.model">
   <xsd:sequence>
      <xsd:element name="fxFeature" type=" FxFeature " minOccurs="0"/>
      <xsd:element name="strategyFeature" type=" StrategyFeature " minOccurs="0"/>
   </xsd:sequence>
</xsd:group>
```

top

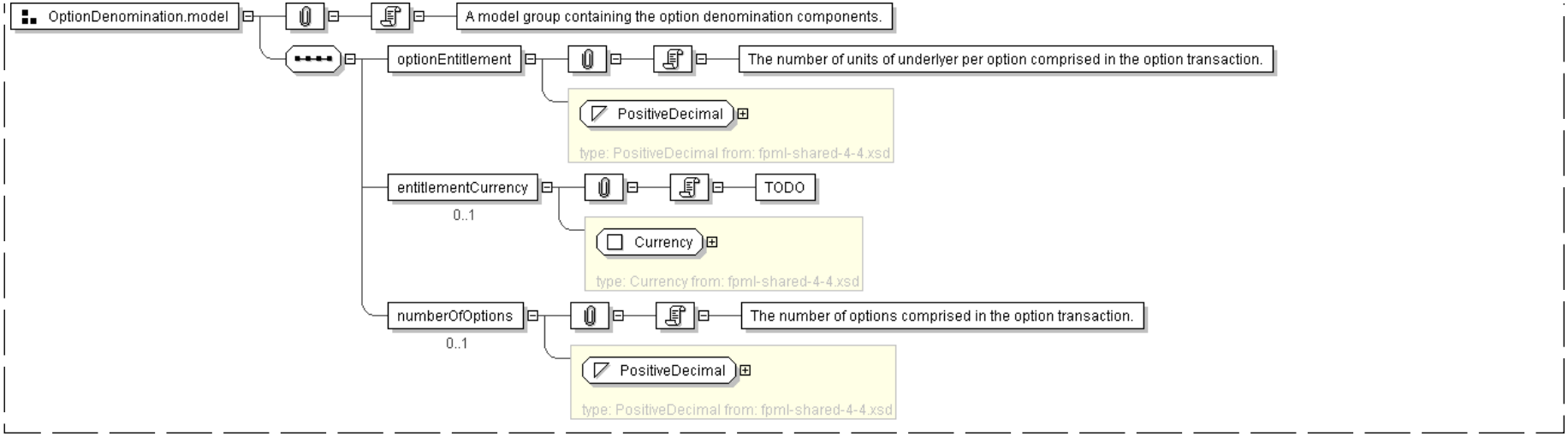
Model Group: OptionDenomination.model

Name	OptionDenomination.model
Used by (from the same schema document)	Complex Type **OptionBaseExtended**
Documentation	A model group containing the option denomination components.

XML Instance Representation

```
<optionEntitlement> PositiveDecimal </optionEntitlement> [1]
'The number of units of underlyer per option comprised in the option transaction.'

<entitlementCurrency> Currency </entitlementCurrency> [0..1]
'TODO'

<numberOfOptions> PositiveDecimal </numberOfOptions> [0..1]
'The number of options comprised in the option transaction.'
```

Diagram



Schema Component Representation

```
<xsd:group name="OptionDenomination.model">
   <xsd:sequence>
      <xsd:element name="optionEntitlement" type=" PositiveDecimal "/>
      <xsd:element name="entitlementCurrency" type=" Currency " minOccurs="0"/>
      <xsd:element name="numberOfOptions" type=" PositiveDecimal " minOccurs="0"/>
   </xsd:sequence>
</xsd:group>
```

top

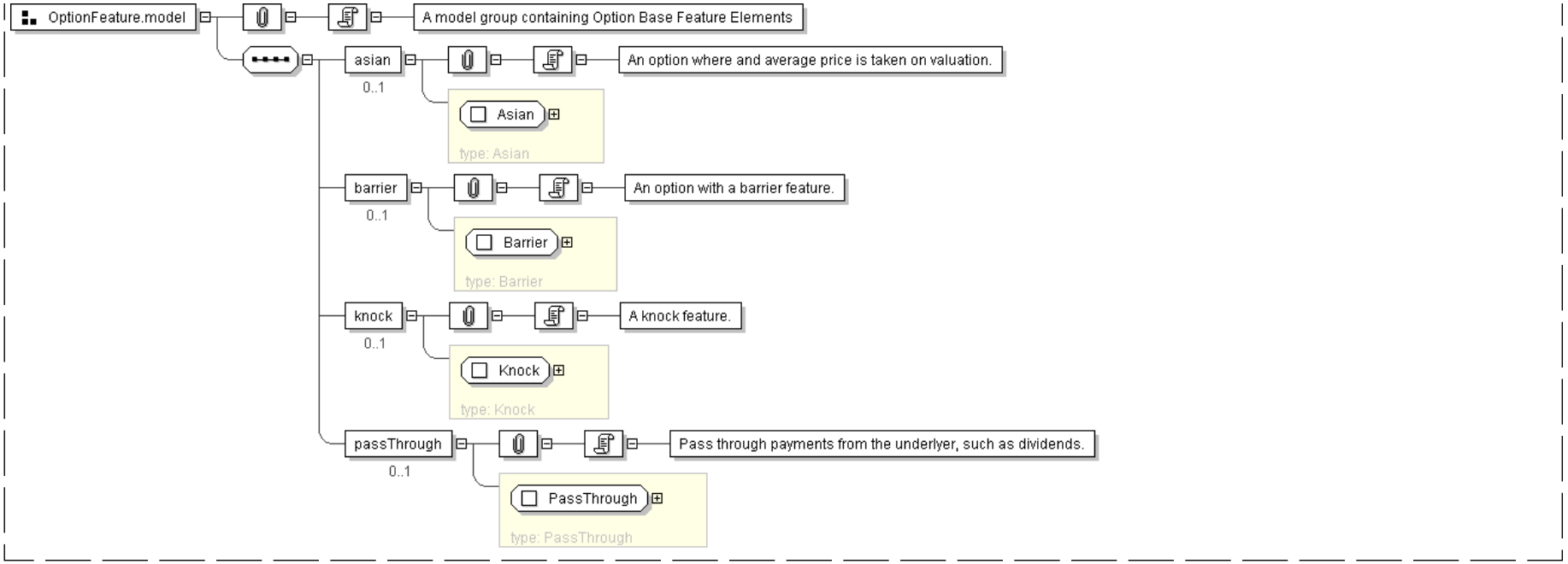
Model Group: OptionFeature.model

Name	OptionFeature.model
Used by (from the same schema document)	Complex Type **OptionFeature**
Documentation	A model group containing Option Base Feature Elements

XML Instance Representation

```
<asian> Asian </asian> [0..1]
'An option where and average price is taken on valuation.'

<barrier> Barrier </barrier> [0..1]
'An option with a barrier feature.'

<knock> Knock </knock> [0..1]
'A knock feature.'

<passThrough> PassThrough </passThrough> [0..1]
'Pass through payments from the underlyer, such as dividends.'
```

Diagram



Schema Component Representation

```
<xsd:group name="OptionFeature.model">
   <xsd:sequence>
      <xsd:element name="asian" type=" Asian " minOccurs="0"/>
      <xsd:element name="barrier" type=" Barrier " minOccurs="0"/>
      <xsd:element name="knock" type=" Knock " minOccurs="0"/>
      <xsd:element name="passThrough" type=" PassThrough " minOccurs="0"/>
   </xsd:sequence>
</xsd:group>
```

top

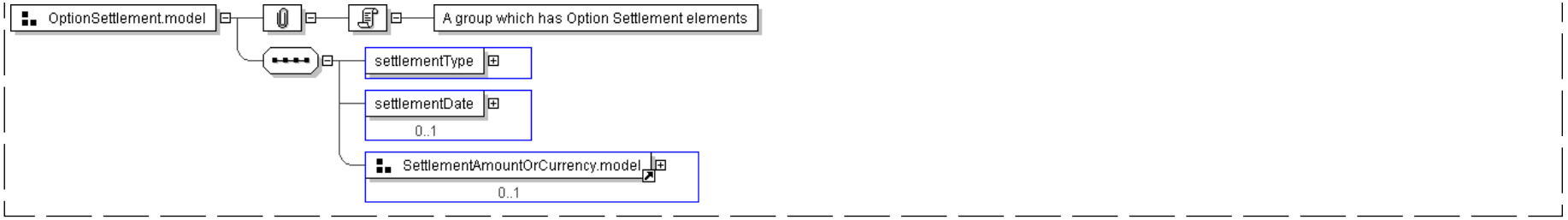
Model Group: OptionSettlement.model

Name	OptionSettlement.model
Used by (from the same schema document)	Complex Type **OptionBaseExtended**
Documentation	A group which has Option Settlement elements

XML Instance Representation

```
<settlementType> SettlementTypeEnum </settlementType> [1]
<settlementDate> AdjustableOrRelativeDate </settlementDate> [0..1]
Start Group: SettlementAmountOrCurrency.model [0..1]
Start Choice [1]
      <settlementAmount> Money </settlementAmount> [1]
      'Settlement Amount'

      <settlementCurrency> Currency </settlementCurrency> [1]
      'Settlement Currency for use where the Settlement Amount cannot be known in advance'

End Choice
End Group: SettlementAmountOrCurrency.model
```

Diagram



Schema Component Representation

```
<xsd:group name="OptionSettlement.model">
   <xsd:sequence>
      <xsd:element name="settlementType" type=" SettlementTypeEnum "/>
      <xsd:element name="settlementDate" type=" AdjustableOrRelativeDate " minOccurs="0"/>
      <xsd:group ref=" SettlementAmountOrCurrency.model " minOccurs="0"/>
   </xsd:sequence>
</xsd:group>
```

top

Legend

Complex Type: **AusAddress**

Schema Component Type — 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.