XML Schema Documentation

Table of Contents

- Schema Document Properties
- Global Declarations
  - Element: **bondOption**
- Global Definitions
  - Complex Type: **BondOption**
  - Complex Type: **BondOptionStrike**
  - Complex Type: **MakeWholeAmount**
  - Complex Type: **ReferenceSwapCurve**
  - Complex Type: **SwapCurveValuation**
- 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 includes components from the following schema document(s):
	  ○ fpml-option-shared-4-4.xsd
	  ○ fpml-mktenv-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: 2350 $"
attributeFormDefault="unqualified" elementFormDefault="qualified">
   <xsd:include schemaLocation="fpml-option-shared-4-4.xsd"/>
   <xsd:include schemaLocation="fpml-mktenv-4-4.xsd"/>
   ...
</xsd:schema>
```

top

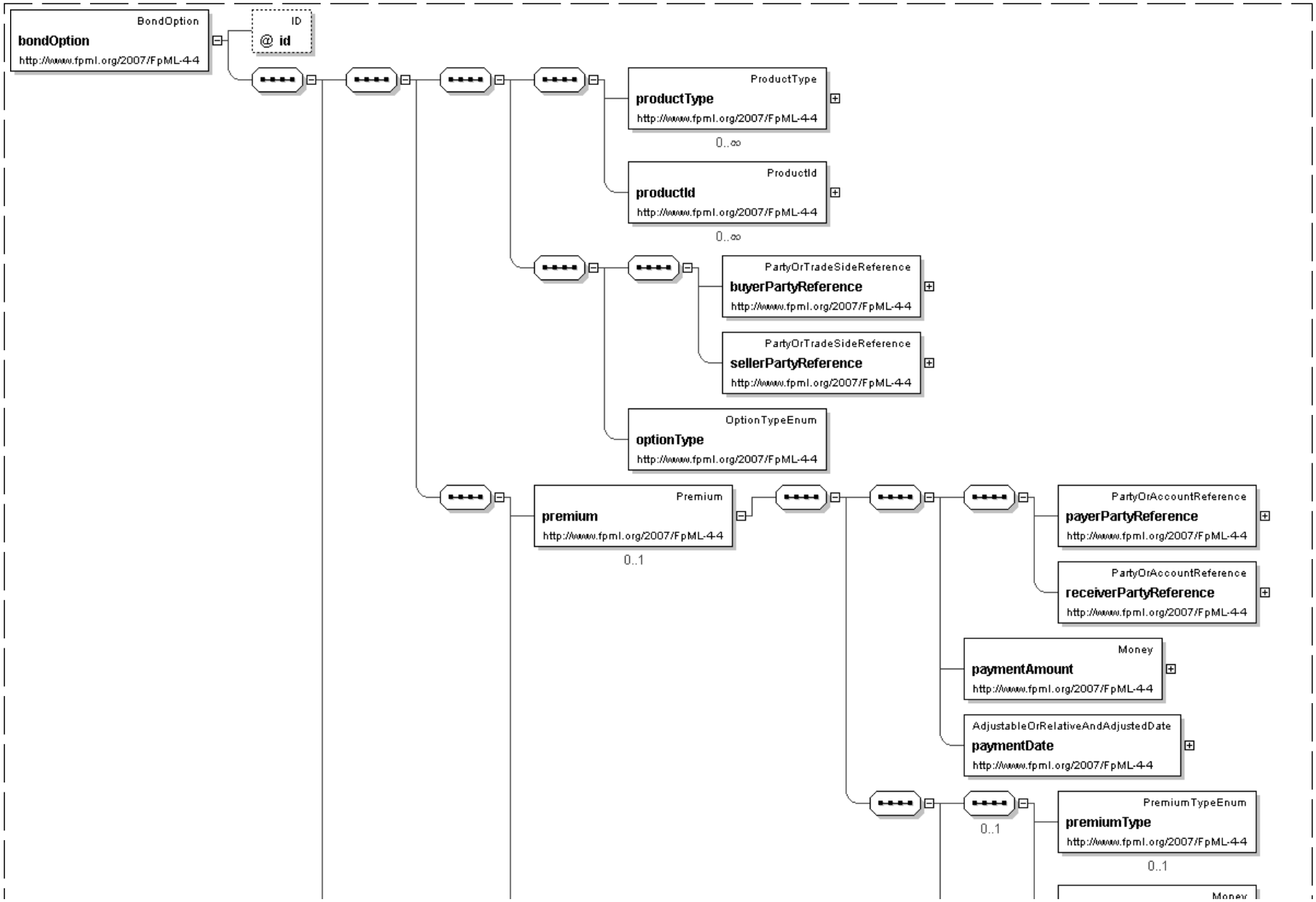
Global Declarations

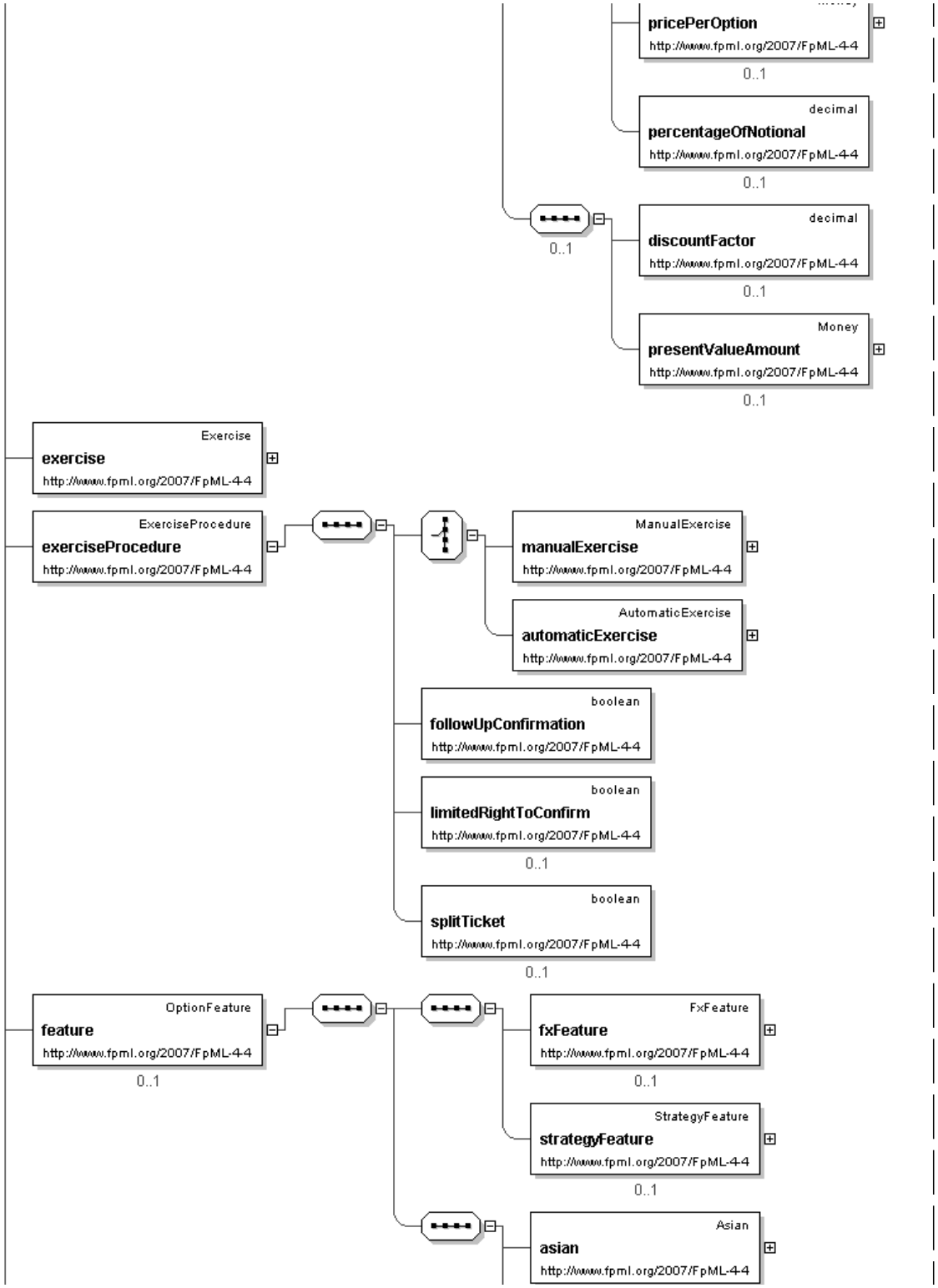
Element: bondOption

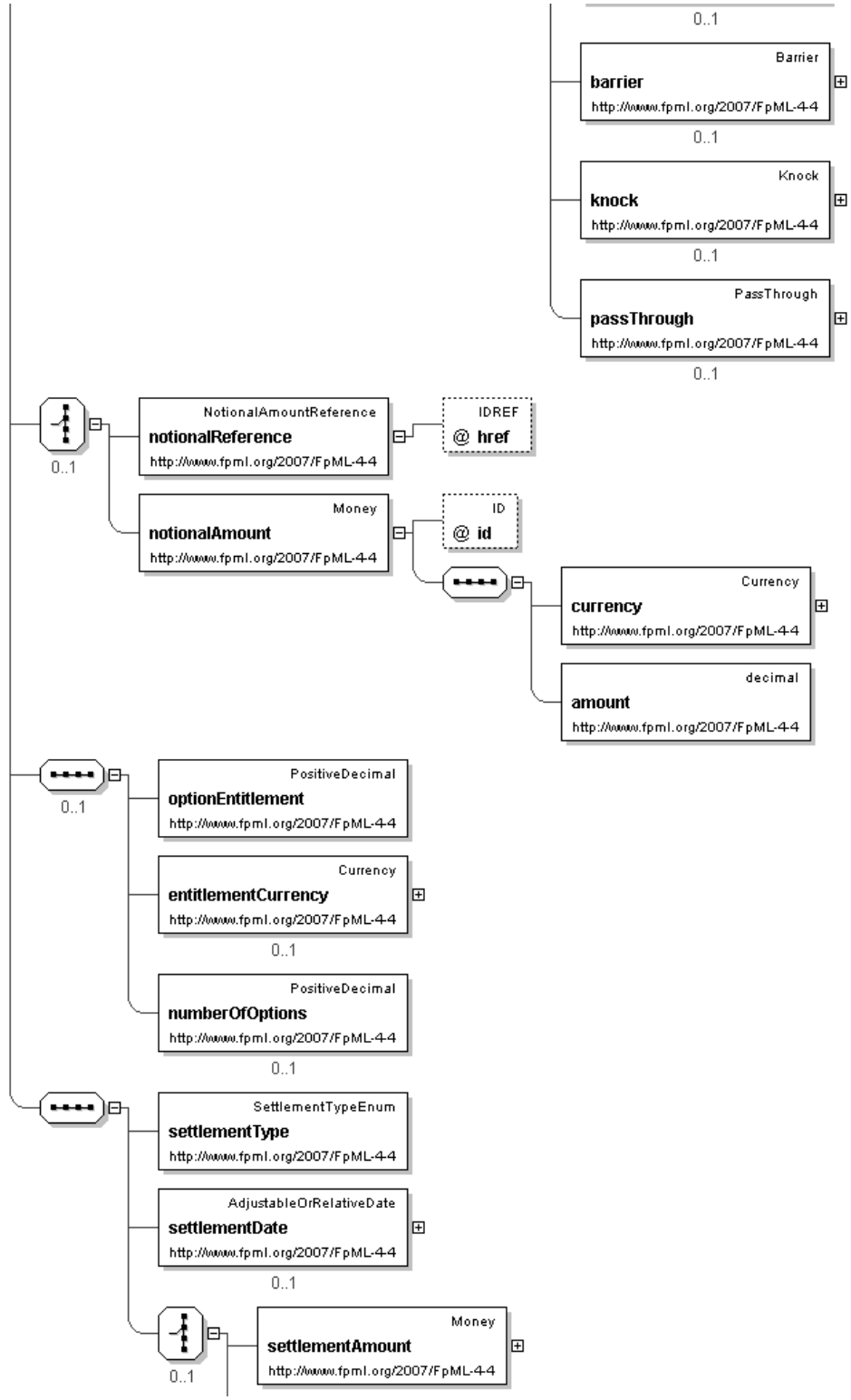
- *This element can be used wherever the following element is referenced:*
  - product

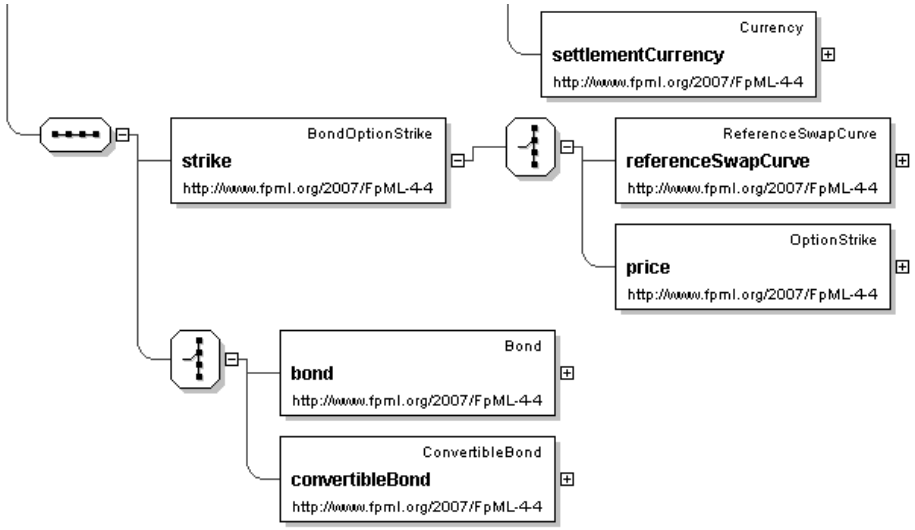
Name	bondOption
Type	BondOption
Nillable	no
Abstract	no
Documentation	A component describing a Bond Option product.

Logical Diagram




pricePerOption
http://www.fpml.org/2007/FpML-4-4
0..1
decimal
percentageOfNotional
http://www.fpml.org/2007/FpML-4-4
0..1
0..1
decimal
discountFactor
http://www.fpml.org/2007/FpML-4-4
0..1
Money
presentValueAmount
http://www.fpml.org/2007/FpML-4-4
0..1
Exercise
exercise
http://www.fpml.org/2007/FpML-4-4
ExerciseProcedure
exerciseProcedure
http://www.fpml.org/2007/FpML-4-4
ManualExercise
manualExercise
http://www.fpml.org/2007/FpML-4-4
AutomaticExercise
automaticExercise
http://www.fpml.org/2007/FpML-4-4
boolean
followUpConfirmation
http://www.fpml.org/2007/FpML-4-4
boolean
limitedRightToConfirm
http://www.fpml.org/2007/FpML-4-4
0..1
boolean
splitTicket
http://www.fpml.org/2007/FpML-4-4
0..1
OptionFeature
feature
http://www.fpml.org/2007/FpML-4-4
0..1
FxFeature
fxFeature
http://www.fpml.org/2007/FpML-4-4
0..1
StrategyFeature
strategyFeature
http://www.fpml.org/2007/FpML-4-4
0..1
Asian
asian
http://www.fpml.org/2007/FpML-4-4


0..1
Barrier
barrier
http://www.fpml.org/2007/FpML-4-4
0..1
Knock
knock
http://www.fpml.org/2007/FpML-4-4
0..1
PassThrough
passThrough
http://www.fpml.org/2007/FpML-4-4
0..1
0..1
NotionalAmountReference
notionalReference
http://www.fpml.org/2007/FpML-4-4
IDREF
@ href
Money
notionalAmount
http://www.fpml.org/2007/FpML-4-4
ID
@ id
Currency
currency
http://www.fpml.org/2007/FpML-4-4
decimal
amount
http://www.fpml.org/2007/FpML-4-4
0..1
PositiveDecimal
optionEntitlement
http://www.fpml.org/2007/FpML-4-4
Currency
entitlementCurrency
http://www.fpml.org/2007/FpML-4-4
0..1
PositiveDecimal
numberOfOptions
http://www.fpml.org/2007/FpML-4-4
0..1
SettlementTypeEnum
settlementType
http://www.fpml.org/2007/FpML-4-4
AdjustableOrRelativeDate
settlementDate
http://www.fpml.org/2007/FpML-4-4
0..1
0..1
Money
settlementAmount
http://www.fpml.org/2007/FpML-4-4



XML Instance Representation

```
<bondOption
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'

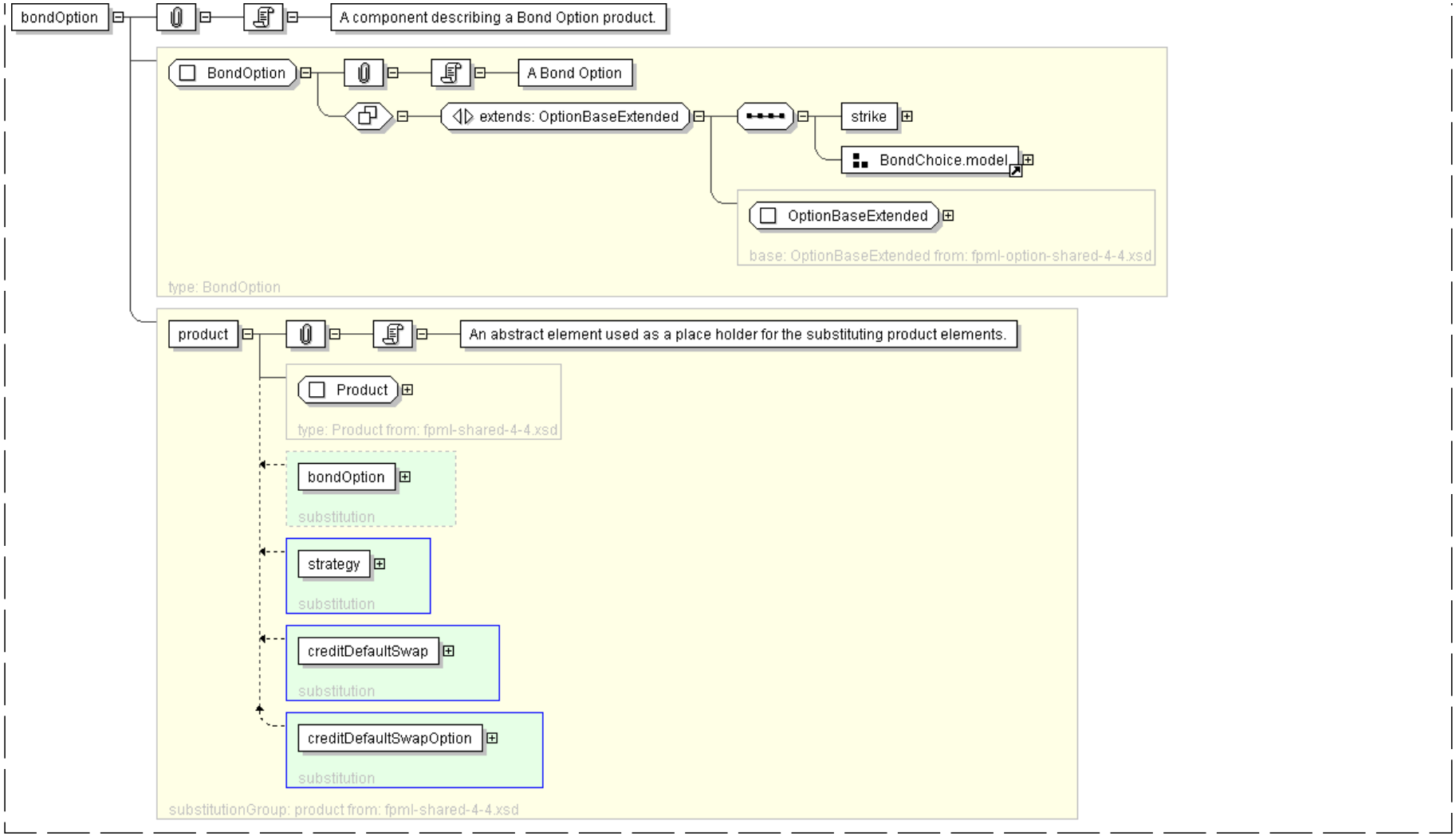
```
   <exercise> ... </exercise> [1]
   <exerciseProcedure> ExerciseProcedure </exerciseProcedure> [1]
```

'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
   <strike> BondOptionStrike </strike> [1]
   'Strike of the the Bond Option.'

Start Choice [1]
      <bond> ... </bond> [1]
      'A bond instrument referenced by a contract'

      <convertibleBond> ... </convertibleBond> [1]
      'A convertible bond instrument referenced by a contract.'

End Choice
</bondOption>
```

Diagram



Schema Component Representation

```
<xsd:element name="bondOption" type=" BondOption " substitutionGroup="product"/>
```

top

Global Definitions

Complex Type: BondOption

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

Name	BondOption
Used by (from the same schema document)	Element **bondOption**

Abstract	no
Documentation	A Bond Option

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]
```

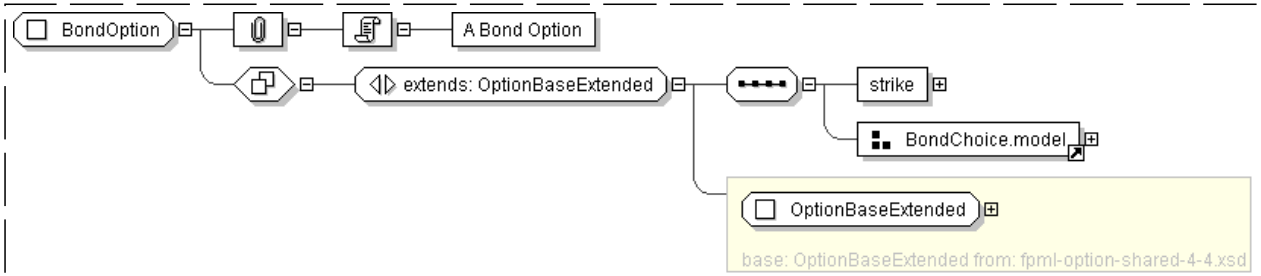
'TODO'

```
      <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
   <strike> BondOptionStrike </strike> [1]
   'Strike of the the Bond Option.'

Start Choice [1]
      <bond> ... </bond> [1]
      'A bond instrument referenced by a contract'

      <convertibleBond> ... </convertibleBond> [1]
      'A convertible bond instrument referenced by a contract.'

End Choice
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="BondOption">
   <xsd:complexContent>
      <xsd:extension base=" OptionBaseExtended ">
         <xsd:sequence>
            <xsd:element name="strike" type=" BondOptionStrike "/>
            <xsd:group ref=" BondChoice.model "/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: BondOptionStrike

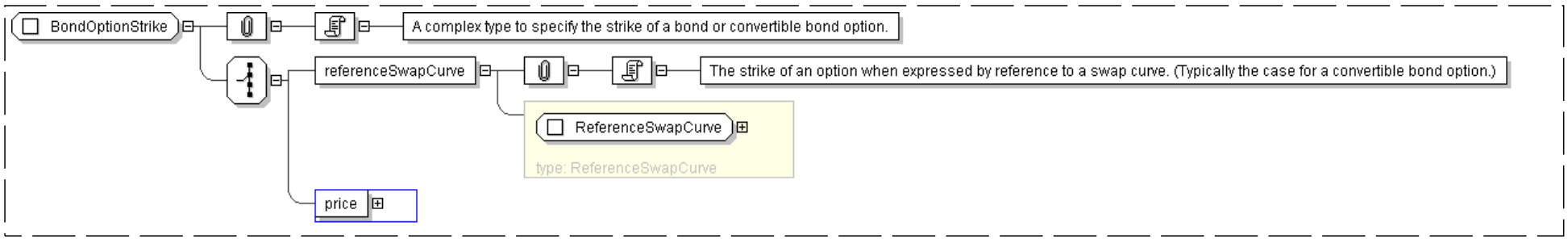
Super-types:	None
Sub-types:	None

Name	BondOptionStrike
Used by (from the same schema document)	Complex Type **BondOption**
Abstract	no
Documentation	A complex type to specify the strike of a bond or convertible bond option.

XML Instance Representation

```
<...>
Start Choice [1]
      <referenceSwapCurve> ReferenceSwapCurve </referenceSwapCurve> [1]
      'The strike of an option when expressed by reference to a swap curve. (Typically the case for
      a convertible bond option.)'

      <price> OptionStrike </price> [1]
End Choice
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="BondOptionStrike">
   <xsd:choice>
      <xsd:element name="referenceSwapCurve" type=" ReferenceSwapCurve "/>
      <xsd:element name="price" type=" OptionStrike "/>
   </xsd:choice>
</xsd:complexType>
```

top

Complex Type: MakeWholeAmount

Super-types:	SwapCurveValuation < **MakeWholeAmount** (by extension)
Sub-types:	None

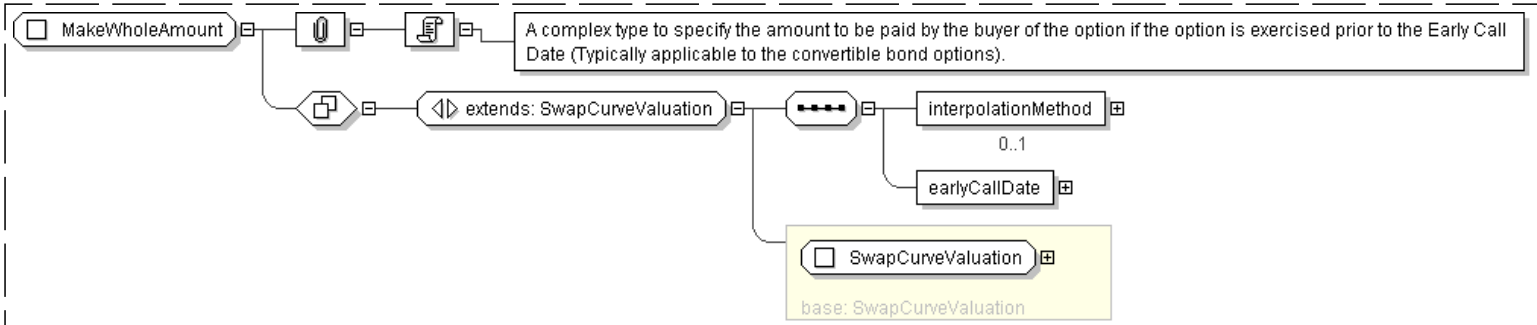
Name	MakeWholeAmount
Used by (from the same schema document)	Complex Type **ReferenceSwapCurve**
Abstract	no
Documentation	A complex type to specify the amount to be paid by the buyer of the option if the option is exercised prior to the Early Call Date (Typically applicable to the convertible bond options).

XML Instance Representation

```
<...>
```

```
<floatingRateIndex> FloatingRateIndex </floatingRateIndex> [1]
<indexTenor> Interval </indexTenor> [0..1]
'The ISDA Designated Maturity, i.e. the tenor of the floating rate.'

<spread> xsd:decimal </spread> [1]
'Spread in basis points over the floating rate index.'

<side> QuotationSideEnum </side> [0..1]
'The side (bid/mid/ask) of the measure.'

<interpolationMethod> InterpolationMethod </interpolationMethod> [0..1]
'The type of interpolation method that the calculation agent reserves the right to use.'

<earlyCallDate> IdentifiedDate </earlyCallDate> [1]
'Date prior to which the option buyer will have to pay a Make Whole Amount to the option
seller if he/she exercises the option.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="MakeWholeAmount">
   <xsd:complexContent>
      <xsd:extension base=" SwapCurveValuation ">
         <xsd:sequence>
            <xsd:element name="interpolationMethod" type=" InterpolationMethod " minOccurs="0"/>
            <xsd:element name="earlyCallDate" type=" IdentifiedDate "/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: ReferenceSwapCurve

Super-types:	None
Sub-types:	None

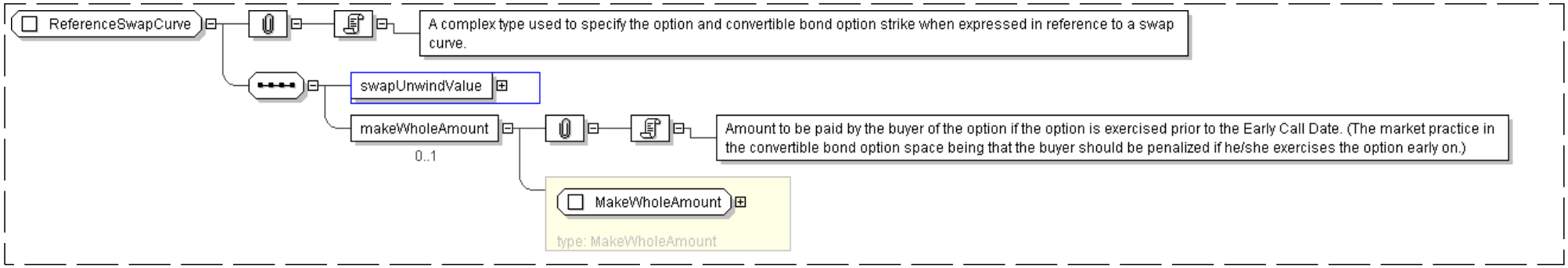
Name	ReferenceSwapCurve
Used by (from the same schema document)	Complex Type **BondOptionStrike**
Abstract	no

Documentation	A complex type used to specify the option and convertible bond option strike when expressed in reference to a swap curve.

XML Instance Representation

```
<...>
   <swapUnwindValue> SwapCurveValuation </swapUnwindValue> [1]
   <makeWholeAmount> MakeWholeAmount </makeWholeAmount> [0..1]
   'Amount to be paid by the buyer of the option if the option is exercised prior to the
   Early Call Date. (The market practice in the convertible bond option space being that the
   buyer should be penalized if he/she exercises the option early on.)'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="ReferenceSwapCurve">
   <xsd:sequence>
      <xsd:element name="swapUnwindValue" type=" SwapCurveValuation "/>
      <xsd:element name="makeWholeAmount" type=" MakeWholeAmount " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: SwapCurveValuation

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

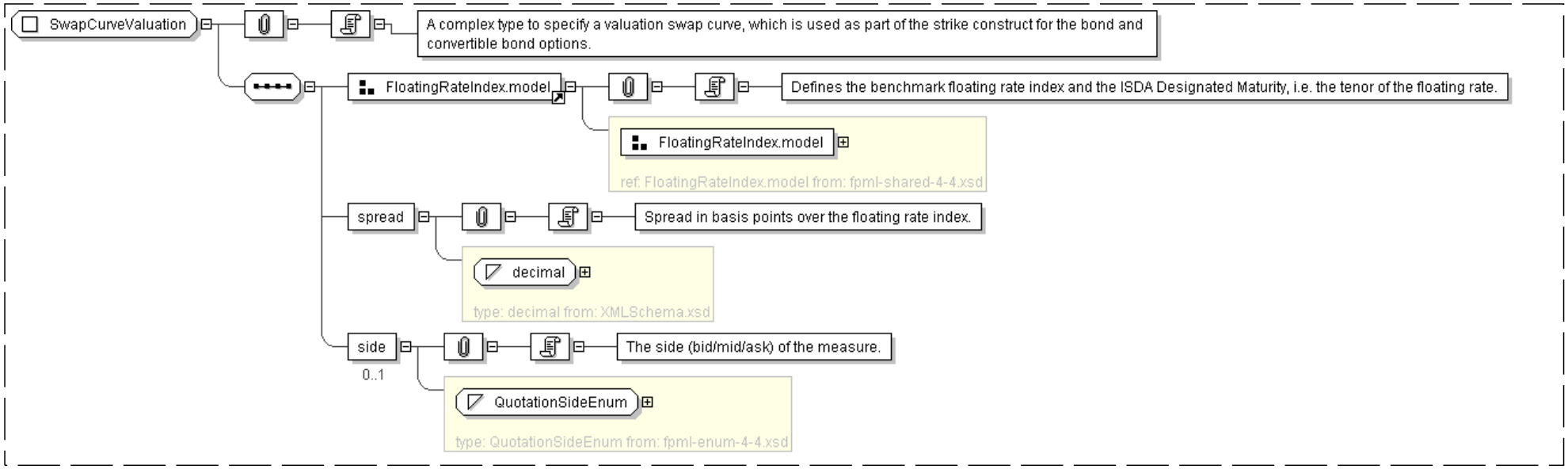
Name	SwapCurveValuation
Used by (from the same schema document)	Complex Type **ReferenceSwapCurve**
Abstract	no
Documentation	A complex type to specify a valuation swap curve, which is used as part of the strike construct for the bond and convertible bond options.

XML Instance Representation

```
<...>
   <floatingRateIndex> FloatingRateIndex </floatingRateIndex> [1]
   <indexTenor> Interval </indexTenor> [0..1]
   'The ISDA Designated Maturity, i.e. the tenor of the floating rate.'

   <spread> xsd:decimal </spread> [1]
   'Spread in basis points over the floating rate index.'
```

```
   <side> QuotationSideEnum </side> [0..1]
   'The side (bid/mid/ask) of the measure.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="SwapCurveValuation">
   <xsd:sequence>
      <xsd:group ref=" FloatingRateIndex.model "/>
      <xsd:element name="spread" type=" xsd:decimal "/>
      <xsd:element name="side" type=" QuotationSideEnum " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

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.