XML Schema Documentation

Table of Contents

- Schema Document Properties
- Global Declarations
  - Element: **americanExercise**
  - Element: **bermudaExercise**
  - Element: **europeanExercise**
  - Element: **exercise**
  - Element: **product**
- Global Definitions
  - Complex Type: **Account**
  - Complex Type: **AccountId**
  - Complex Type: **AccountReference**
  - Complex Type: **Address**
  - Complex Type: **AdjustableDate**
  - Complex Type: **AdjustableDate2**
  - Complex Type: **AdjustableDates**
  - Complex Type: **AdjustableDatesOrRelativeDateOffset**
  - Complex Type: **AdjustableOrRelativeAndAdjustedDate**
  - Complex Type: **AdjustableOrRelativeDate**
  - Complex Type: **AdjustableOrRelativeDates**
  - Complex Type: **AdjustableRelativeOrPeriodicDates**
  - Complex Type: **AdjustedRelativeDateOffset**
  - Complex Type: **AmericanExercise**
  - Complex Type: **AmountReference**
  - Complex Type: **AmountSchedule**
  - Complex Type: **AutomaticExercise**
  - Complex Type: **Beneficiary**
  - Complex Type: **BermudaExercise**
  - Complex Type: **BrokerConfirmation**
  - Complex Type: **BrokerConfirmationType**
  - Complex Type: **BusinessCenter**
  - Complex Type: **BusinessCenterTime**
  - Complex Type: **BusinessCenters**
  - Complex Type: **BusinessCentersReference**
  - Complex Type: **BusinessDateRange**
  - Complex Type: **BusinessDayAdjustments**
  - Complex Type: **BusinessDayAdjustmentsReference**
  - Complex Type: **CalculationAgent**
  - Complex Type: **CalculationPeriodFrequency**
  - Complex Type: **CashSettlementReferenceBanks**
  - Complex Type: **CashflowType**
  - Complex Type: **ClearanceSystem**
  - Complex Type: **ContractualDefinitions**
  - Complex Type: **ContractualMatrix**
  - Complex Type: **ContractualSupplement**
  - Complex Type: **ContractualTermsSupplement**
  - Complex Type: **CorrespondentInformation**
  - Complex Type: **Country**
  - Complex Type: **CreditSeniority**
  - Complex Type: **Currency**
  - Complex Type: **DateList**
  - Complex Type: **DateOffset**
  - Complex Type: **DateRange**
  - Complex Type: **DateReference**
  - Complex Type: **DateTimeList**
  - Complex Type: **DayCountFraction**
  - Complex Type: **DeterminationMethod**
  - Complex Type: **DividendConditions**
  - Complex Type: **DividendPaymentDate**
  - Complex Type: **Documentation**

- Complex Type: **Empty**
- Complex Type: **EntityId**
- Complex Type: **EntityName**
- Complex Type: **EuropeanExercise**
- Complex Type: **ExchangeId**
- Complex Type: **Exercise**
- Complex Type: **ExerciseFee**
- Complex Type: **ExerciseFeeSchedule**
- Complex Type: **ExerciseNotice**
- Complex Type: **ExerciseProcedure**
- Complex Type: **FloatingRate**
- Complex Type: **FloatingRateCalculation**
- Complex Type: **FloatingRateIndex**
- Complex Type: **ForecastRateIndex**
- Complex Type: **Formula**
- Complex Type: **FormulaComponent**
- Complex Type: **FxCashSettlement**
- Complex Type: **FxFixing**
- Complex Type: **FxRate**
- Complex Type: **FxSpotRateSource**
- Complex Type: **GoverningLaw**
- Complex Type: **IdentifiedCurrency**
- Complex Type: **IdentifiedCurrencyReference**
- Complex Type: **IdentifiedDate**
- Complex Type: **IdentifiedPayerReceiver**
- Complex Type: **InformationProvider**
- Complex Type: **InformationSource**
- Complex Type: **InstrumentId**
- Complex Type: **InterestAccrualsCompoundingMethod**
- Complex Type: **InterestAccrualsMethod**
- Complex Type: **IntermediaryInformation**
- Complex Type: **Interval**
- Complex Type: **Leg**
- Complex Type: **LegalEntity**
- Complex Type: **LegalEntityReference**
- Complex Type: **MainPublication**
- Complex Type: **ManualExercise**
- Complex Type: **MasterAgreement**
- Complex Type: **MasterAgreementType**
- Complex Type: **MasterConfirmation**
- Complex Type: **MasterConfirmationType**
- Complex Type: **Math**
- Complex Type: **MatrixTerm**
- Complex Type: **MatrixType**
- Complex Type: **MimeType**
- Complex Type: **Money**
- Complex Type: **MultipleExercise**
- Complex Type: **NotionalAmountReference**
- Complex Type: **Offset**
- Complex Type: **PartialExercise**
- Complex Type: **Party**
- Complex Type: **PartyId**
- Complex Type: **PartyOrAccountReference**
- Complex Type: **PartyOrTradeSideReference**
- Complex Type: **PartyReference**
- Complex Type: **Payment**
- Complex Type: **PaymentCurrency**
- Complex Type: **PaymentType**
- Complex Type: **PeriodicDates**
- Complex Type: **PricingStructure**
- Complex Type: **PricingStructureReference**
- Complex Type: **PrincipalExchanges**
- Complex Type: **Product**
- Complex Type: **ProductId**
- Complex Type: **ProductReference**

- Complex Type: **ProductType**
- Complex Type: **QuotedCurrencyPair**
- Complex Type: **Rate**
- Complex Type: **RateObservation**
- Complex Type: **RateReference**
- Complex Type: **RateSourcePage**
- Complex Type: **Reference**
- Complex Type: **ReferenceAmount**
- Complex Type: **ReferenceBank**
- Complex Type: **ReferenceBankId**
- Complex Type: **RelativeDateOffset**
- Complex Type: **RelativeDateSequence**
- Complex Type: **RelativeDates**
- Complex Type: **RequiredIdentifierDate**
- Complex Type: **ResetFrequency**
- Complex Type: **Rounding**
- Complex Type: **Routing**
- Complex Type: **RoutingExplicitDetails**
- Complex Type: **RoutingId**
- Complex Type: **RoutingIds**
- Complex Type: **RoutingIdsAndExplicitDetails**
- Complex Type: **Schedule**
- Complex Type: **ScheduleReference**
- Complex Type: **SettlementInformation**
- Complex Type: **SettlementInstruction**
- Complex Type: **SettlementMethod**
- Complex Type: **SettlementPriceSource**
- Complex Type: **SettlementRateSource**
- Complex Type: **SharedAmericanExercise**
- Complex Type: **SimplePayment**
- Complex Type: **SplitSettlement**
- Complex Type: **SpreadSchedule**
- Complex Type: **SpreadScheduleReference**
- Complex Type: **SpreadScheduleType**
- Complex Type: **Step**
- Complex Type: **StreetAddress**
- Complex Type: **Strike**
- Complex Type: **StrikeSchedule**
- Complex Type: **Stub**
- Complex Type: **StubValue**
- Model Group: **BusinessCentersOrReference.model**
- Model Group: **BuyerSeller.model**
- Model Group: **FloatingRateIndex.model**
- Model Group: **PartialExercise.model**
- Model Group: **PayerReceiver.model**
- Model Group: **PaymentDiscounting.model**
- Model Group: **Premium.model**
- Model Group: **Product.model**
- Model Group: **RoutingExplicitDetails.model**
- Model Group: **RoutingIdentification.model**
- Model Group: **SettlementAmountOrCurrency.model**
- Model Group: **VersionHistory.model**
- Simple Type: **CorrelationValue**
- Simple Type: **HourMinuteTime**
- Simple Type: **NonNegativeDecimal**
- Simple Type: **PositiveDecimal**
- Simple Type: **RestrictedPercentage**

- Legend
- Glossary

top

Schema Document Properties

Target Namespace	**http://www.fpml.org/2007/FpML-4-4**

Version	$Revision: 2797 $
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-enum-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
fpml-annotation	http://www.fpml.org/annotation
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: 2797 $"
attributeFormDefault="unqualified" elementFormDefault="qualified">
   <xsd:include schemaLocation="fpml-enum-4-4.xsd"/>
   ...
</xsd:schema>
```

top

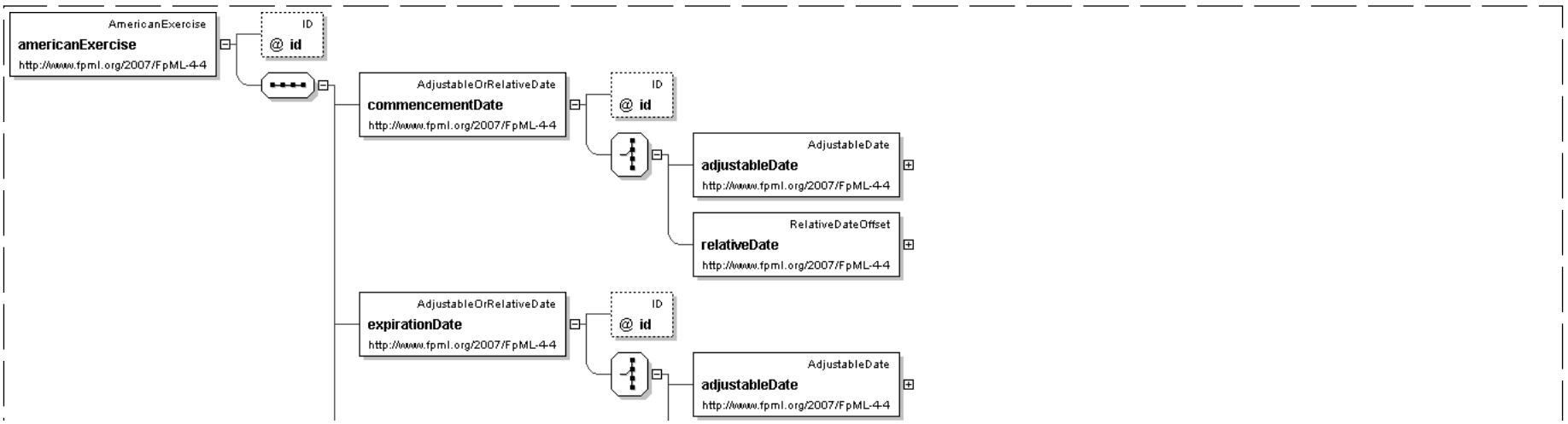
Global Declarations

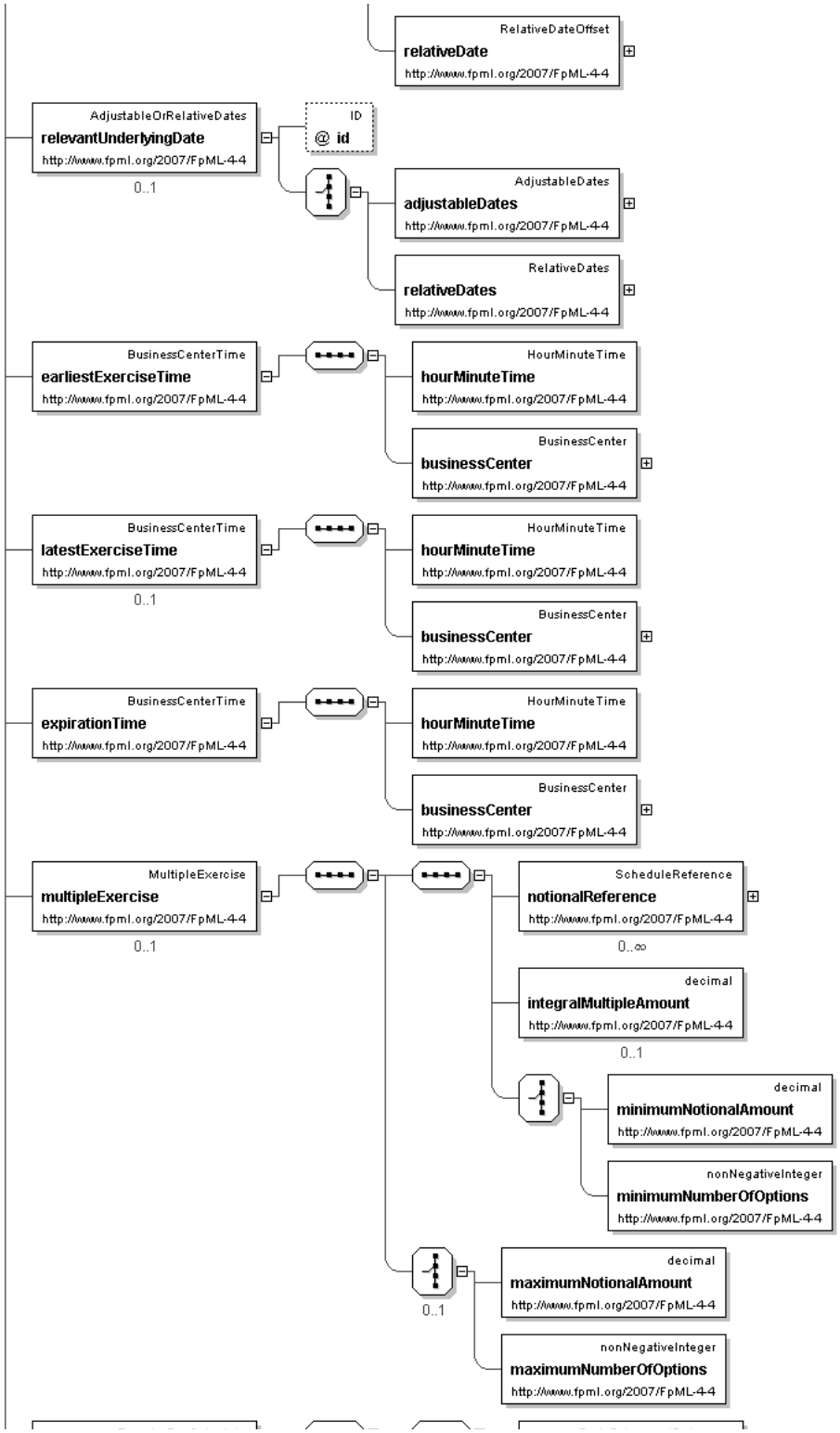
Element: americanExercise

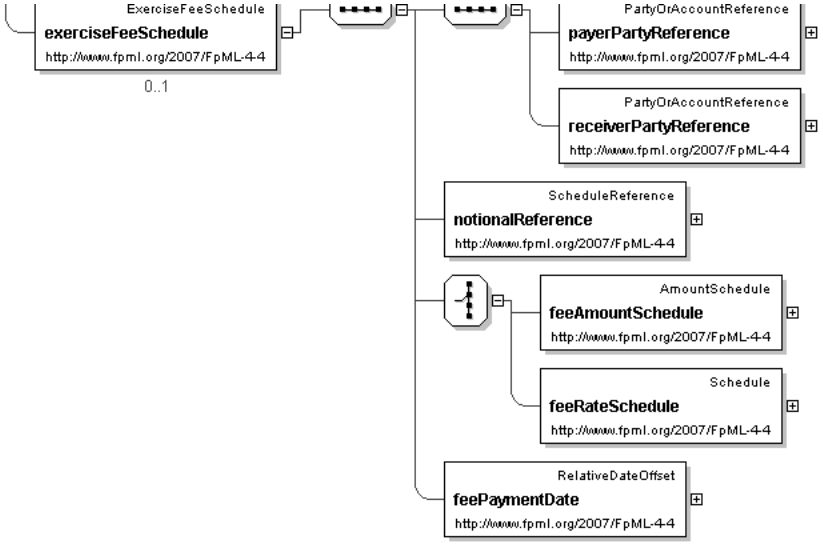
- *This element can be used wherever the following element is referenced:*
  - exercise

Name	americanExercise
Type	AmericanExercise
Nillable	no
Abstract	no
Documentation	The parameters for defining the exercise period for an American style option together with any rules governing the notional amount of the underlying which can be exercised on any given exercise date and any associated exercise fees.

Logical Diagram




RelativeDateOffset
relativeDate
http://www.fpml.org/2007/FpML-4-4
AdjustableOrRelativeDates
relevantUnderlyingDate
http://www.fpml.org/2007/FpML-4-4
0..1
ID
@ id
AdjustableDates
adjustableDates
http://www.fpml.org/2007/FpML-4-4
RelativeDates
relativeDates
http://www.fpml.org/2007/FpML-4-4
BusinessCenterTime
earliestExerciseTime
http://www.fpml.org/2007/FpML-4-4
HourMinuteTime
hourMinuteTime
http://www.fpml.org/2007/FpML-4-4
BusinessCenter
businessCenter
http://www.fpml.org/2007/FpML-4-4
BusinessCenterTime
latestExerciseTime
http://www.fpml.org/2007/FpML-4-4
0..1
HourMinuteTime
hourMinuteTime
http://www.fpml.org/2007/FpML-4-4
BusinessCenter
businessCenter
http://www.fpml.org/2007/FpML-4-4
BusinessCenterTime
expirationTime
http://www.fpml.org/2007/FpML-4-4
HourMinuteTime
hourMinuteTime
http://www.fpml.org/2007/FpML-4-4
BusinessCenter
businessCenter
http://www.fpml.org/2007/FpML-4-4
MultipleExercise
multipleExercise
http://www.fpml.org/2007/FpML-4-4
0..1
ScheduleReference
notionalReference
http://www.fpml.org/2007/FpML-4-4
0..∞
decimal
integralMultipleAmount
http://www.fpml.org/2007/FpML-4-4
0..1
decimal
minimumNotionalAmount
http://www.fpml.org/2007/FpML-4-4
nonNegativeInteger
minimumNumberOfOptions
http://www.fpml.org/2007/FpML-4-4
0..1
decimal
maximumNotionalAmount
http://www.fpml.org/2007/FpML-4-4
nonNegativeInteger
maximumNumberOfOptions
http://www.fpml.org/2007/FpML-4-4

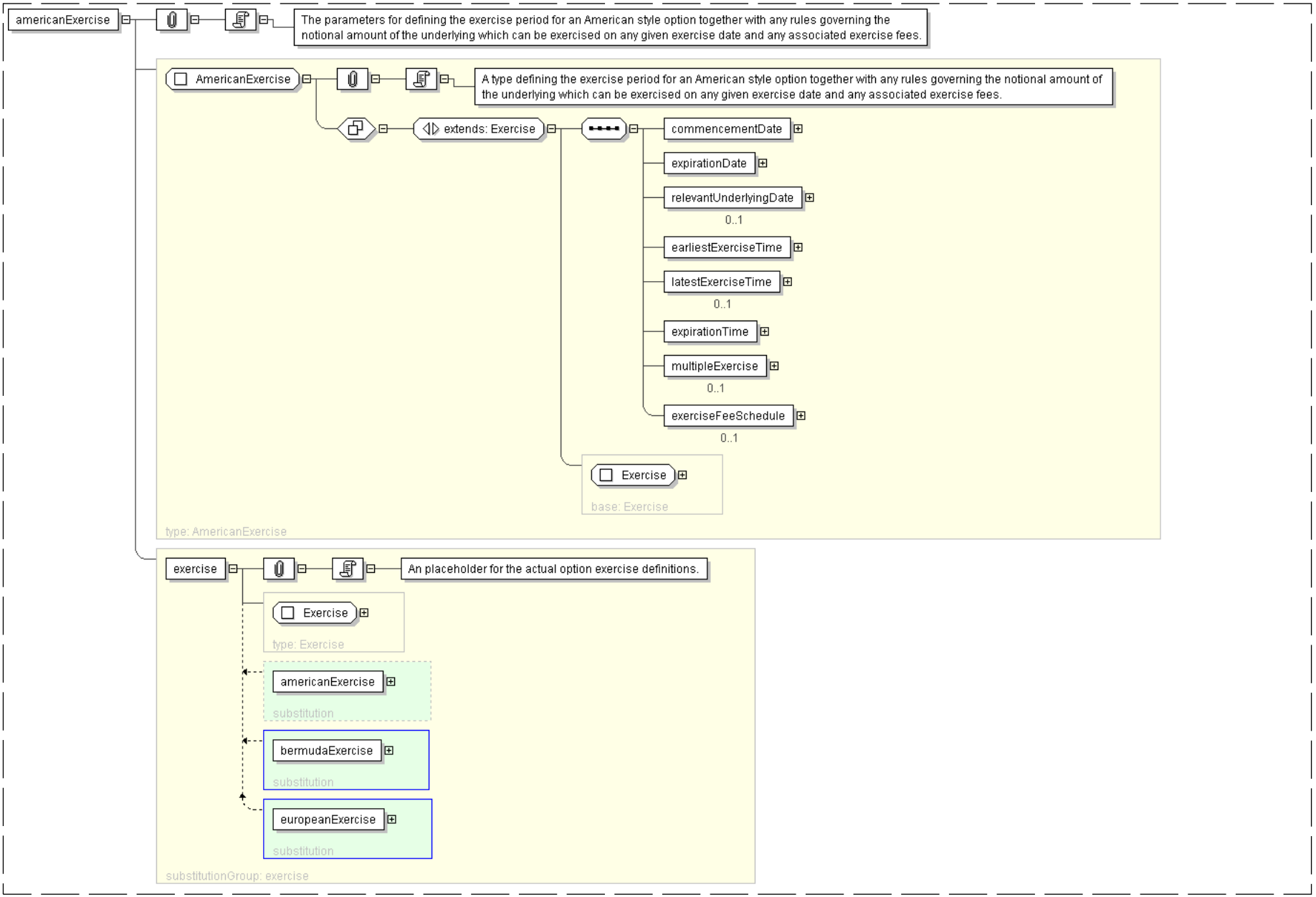


XML Instance Representation

```
<americanExercise
id=" xsd:ID [0..1]">
   <commencementDate> AdjustableOrRelativeDate </commencementDate> [1]
   'The first day of the exercise period for an American style option.'

   <expirationDate> AdjustableOrRelativeDate </expirationDate> [1]
   'The last day within an exercise period for an American style option. For a European
   style option it is the only day within the exercise period.'

   <relevantUnderlyingDate> AdjustableOrRelativeDates </relevantUnderlyingDate> [0..1]
   'The daye on the underlying set by the exercise of an option. What this date is depends on
   the option (e.g. in a swaption it is the effective date, in an extendible/cancelable
   provision it is the termination date).'

   <earliestExerciseTime> BusinessCenterTime </earliestExerciseTime> [1]
   'The earliest time at which notice of exercise can be given by the buyer to the seller
   (or seller\'s agent) i) on the expriation date, in the case of a European style option, (ii)
   on each bermuda option exercise date and the expiration date, in the case of a Bermuda
   style option the commencement date to, and including, the expiration date , in the case of
   an American option.'

   <latestExerciseTime> BusinessCenterTime </latestExerciseTime> [0..1]
   'For a Bermuda or American style option, the latest time on an exercise business day
   (excluding the expiration date) within the exercise period that notice can be given by
   the buyer to the seller or seller\'s agent. Notice of exercise given after this time will
   be deemed to have been given on the next exercise business day.'

   <expirationTime> BusinessCenterTime </expirationTime> [1]
   'The latest time for exercise on expirationDate.'

   <multipleExercise> MultipleExercise </multipleExercise> [0..1]
   'As defined in the 2000 ISDA Definitions, Section 12.4. Multiple Exercise, the buyer of
   the option has the right to exercise all or less than all the unexercised notional amount
   of the underlying swap on one or more days in the exercise period, but on any such day may
   not exercise less than the minimum notional amount or more that the maximum notional
   amount, and if an integral multiple amount is specified, the notional amount exercised must
   be equal to, or be an intergral multiple of, the integral multiple amount.'

   <exerciseFeeSchedule> ExerciseFeeSchedule </exerciseFeeSchedule> [0..1]
   'The fees associated with an exercise date. The fees are conditional on the exercise
   occuring. The fees can be specified as actual currency amounts or as percentages of
```

```
    the notional amount being exercised.'
</americanExercise>
```

Diagram



Schema Component Representation

```
<xsd:element name="americanExercise" type=" AmericanExercise " substitutionGroup="exercise"/>
```

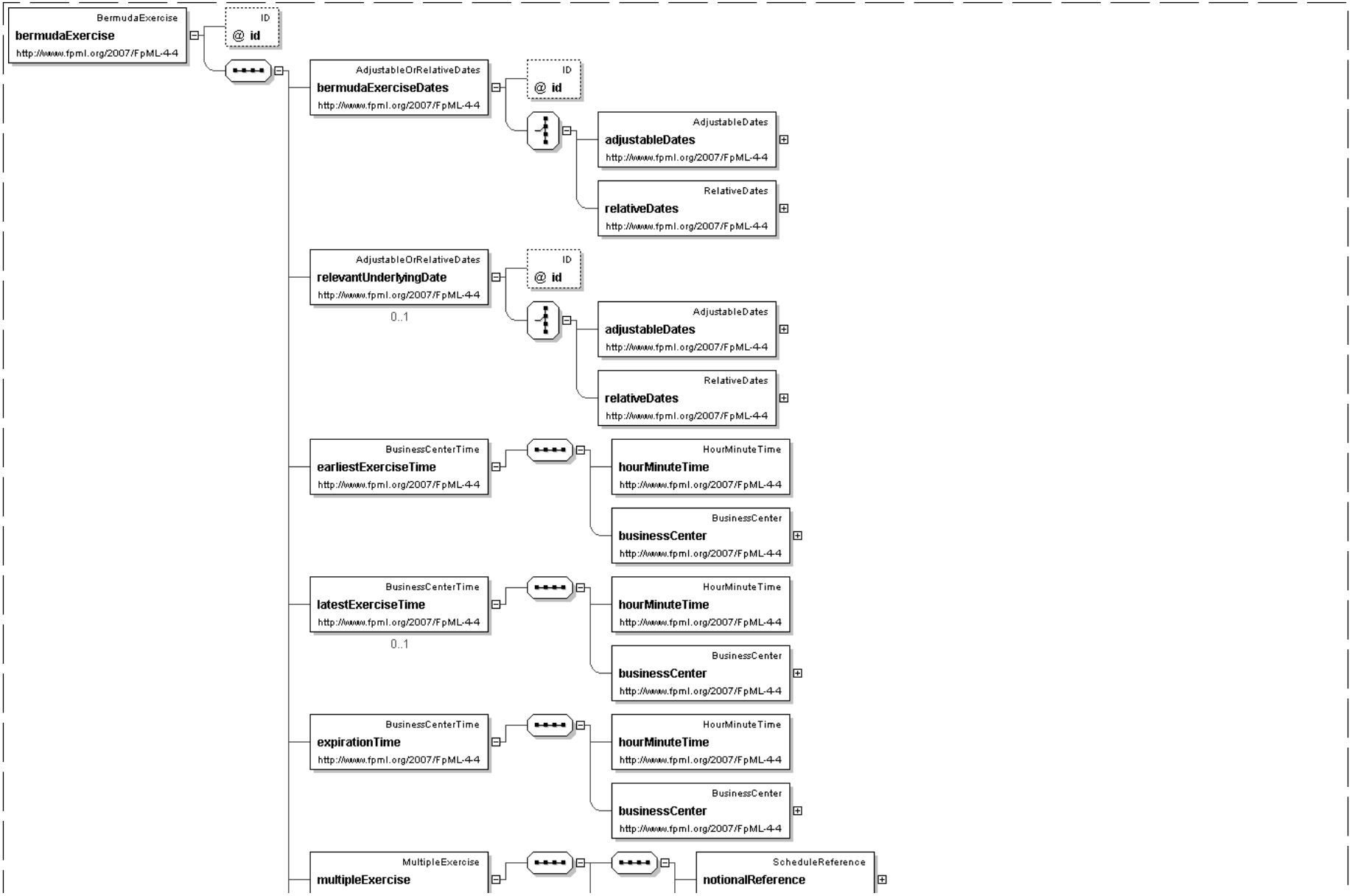
top

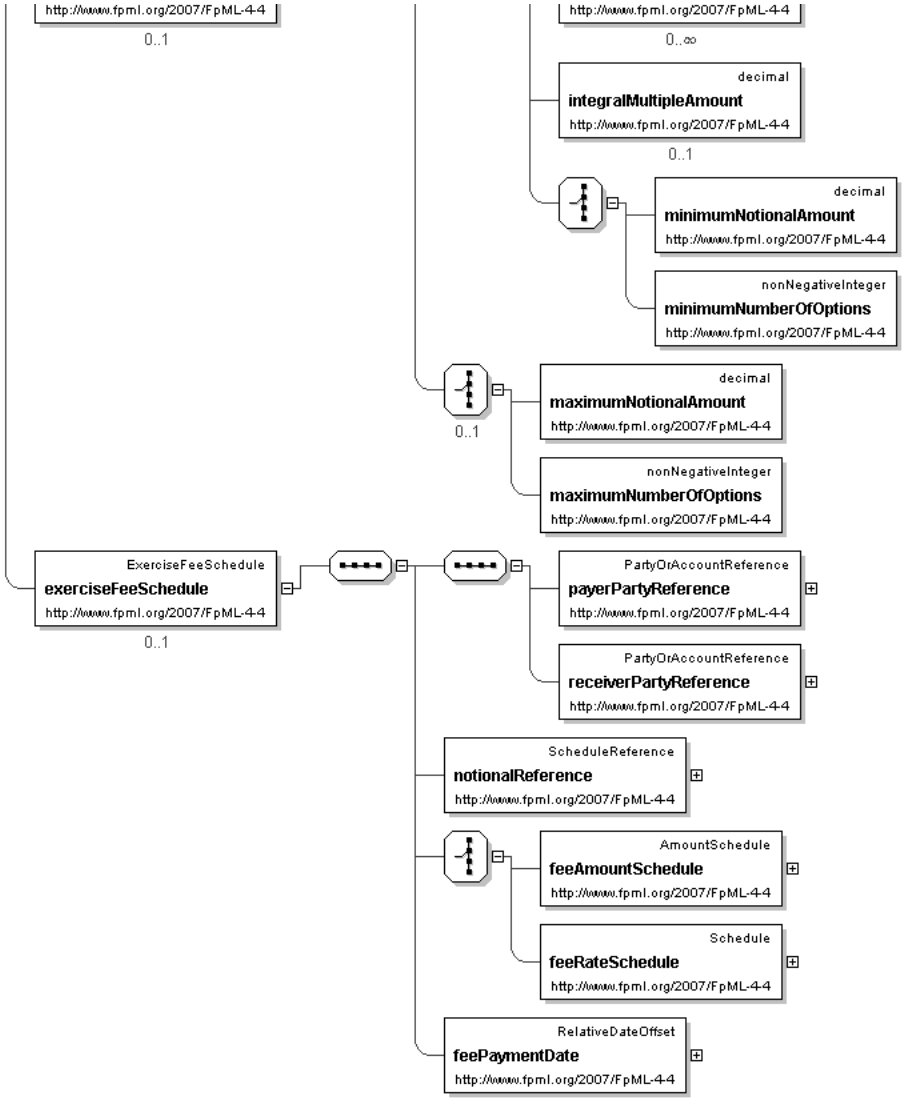
Element: bermudaExercise

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

Name	bermudaExercise
Type	BermudaExercise
Nillable	no
Abstract	no
Documentation	The parameters for defining the exercise period for a Bermuda style option together with any rules governing the notional amount of the underlying which can be exercised on any given exercise date and any associated exercise fees.

Logical Diagram





XML Instance Representation

```
<bermudaExercise
id=" xsd:ID [0..1]">
   <bermudaExerciseDates> AdjustableOrRelativeDates </bermudaExerciseDates> [1]
```

'The dates the define the Bermuda option exercise dates and the expiration date. The last specified date is assumed to be the expiration date. The dates can either be specified as a series of explicit dates and associated adjustments or as a series of dates defined relative to another schedule of dates, for example, the calculation period start dates. Where a relative series of dates are defined the first and last possible exercise dates can be separately specified.'

```
   <relevantUnderlyingDate> AdjustableOrRelativeDates </relevantUnderlyingDate> [0..1]
```

'The daye on the underlying set by the exercise of an option. What this date is depends on the option (e.g. in a swaption it is the effective date, in an extendible/cancelable provision it is the termination date).'

```
   <earliestExerciseTime> BusinessCenterTime </earliestExerciseTime> [1]
```

'The earliest time at which notice of exercise can be given by the buyer to the seller (or seller\'s agent) i) on the expriation date, in the case of a European style option, (ii) on each bermuda option exercise date and the expiration date, in the case of a Bermuda style option the commencement date to, and including, the expiration date , in the case of an American option.'

<latestExerciseTime> BusinessCenterTime </latestExerciseTime> [0..1]

'For a Bermuda or American style option, the latest time on an exercise business day (excluding the expiration date) within the exercise period that notice can be given by the buyer to the seller or seller\'s agent. Notice of exercise given after this time will be deemed to have been given on the next exercise business day.'

<expirationTime> BusinessCenterTime </expirationTime> [1]

'The latest time for exercise on expirationDate.'

<multipleExercise> MultipleExercise </multipleExercise> [0..1]

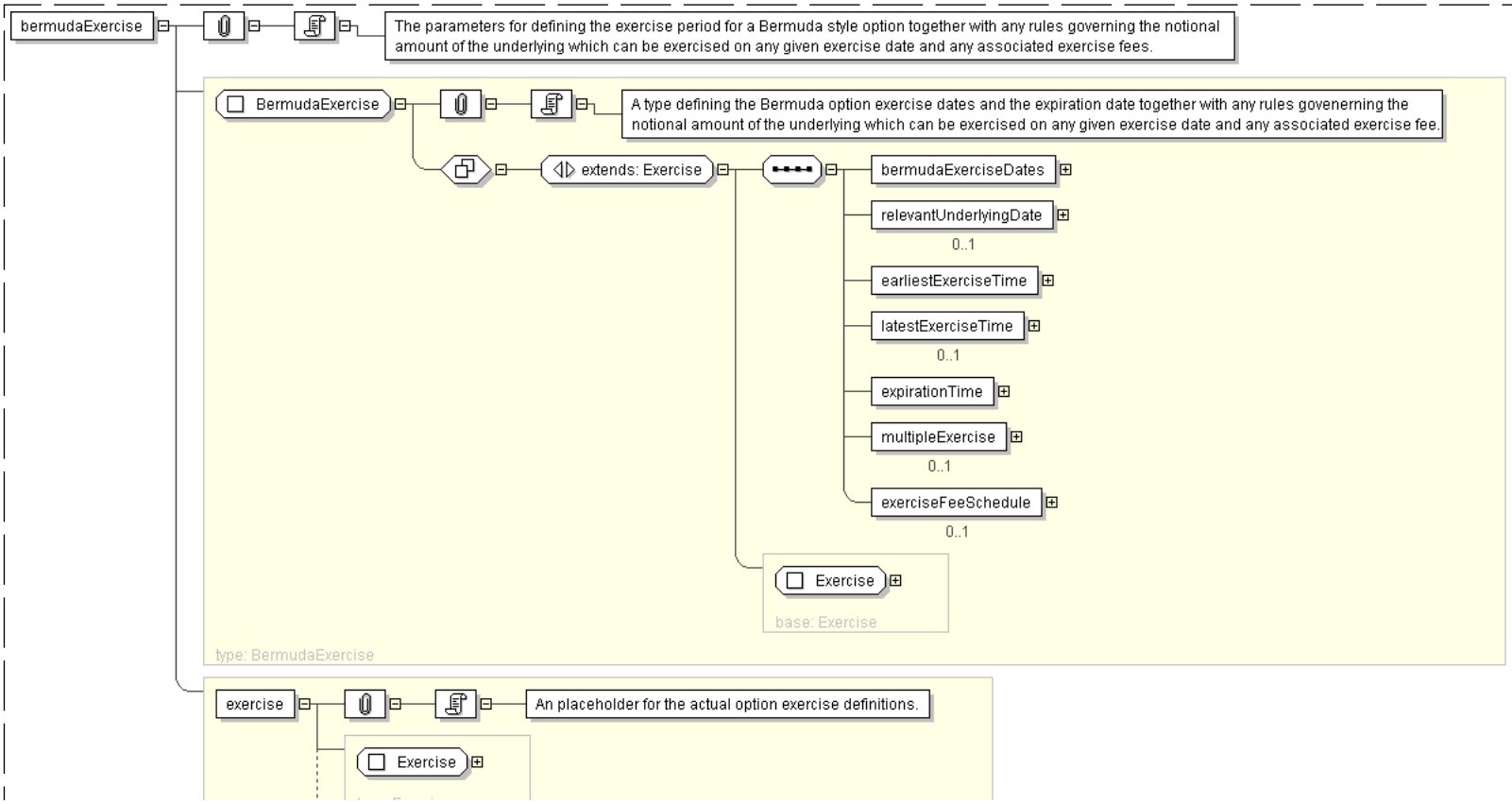
'As defined in the 2000 ISDA Definitions, Section 12.4. Multiple Exercise, the buyer of the option has the right to exercise all or less than all the unexercised notional amount of the underlying swap on one or more days in the exercise period, but on any such day may not exercise less than the minimum notional amount or more that the maximum notional amount, and if an integral multiple amount is specified, the notional amount exercised must be equal to, or be an intergral multiple of, the integral multiple amount.'

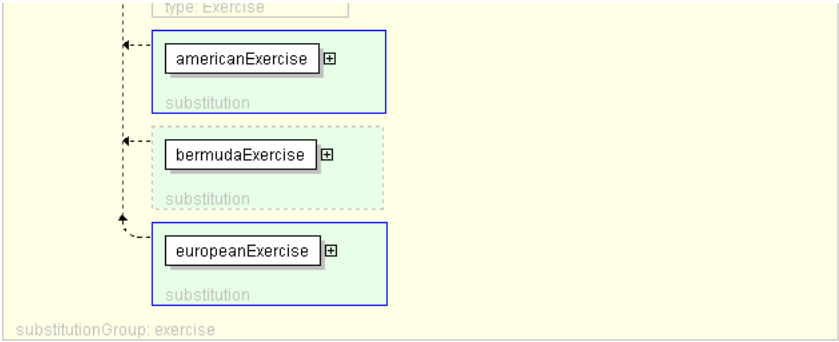
<exerciseFeeSchedule> ExerciseFeeSchedule </exerciseFeeSchedule> [0..1]

'The fees associated with an exercise date. The fees are conditional on the exercise occuring. The fees can be specified as actual currency amounts or as percentages of the notional amount being exercised.'

</bermudaExercise>

Diagram





Schema Component Representation

```
<xsd:element name="bermudaExercise" type=" BermudaExercise " substitutionGroup="exercise"/>
```

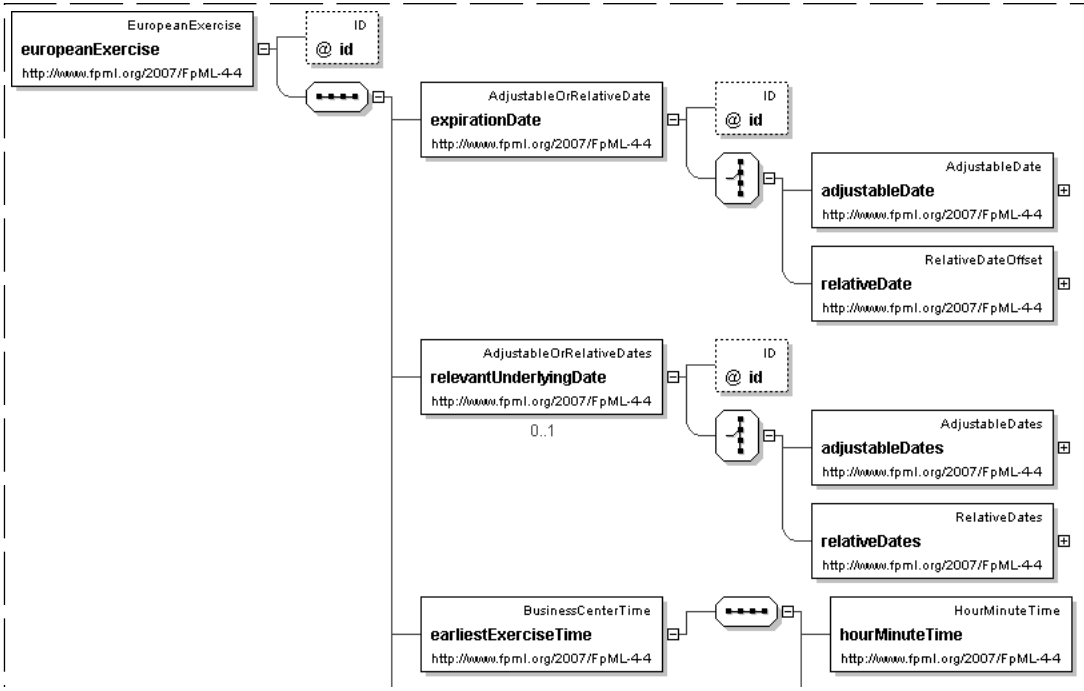
top

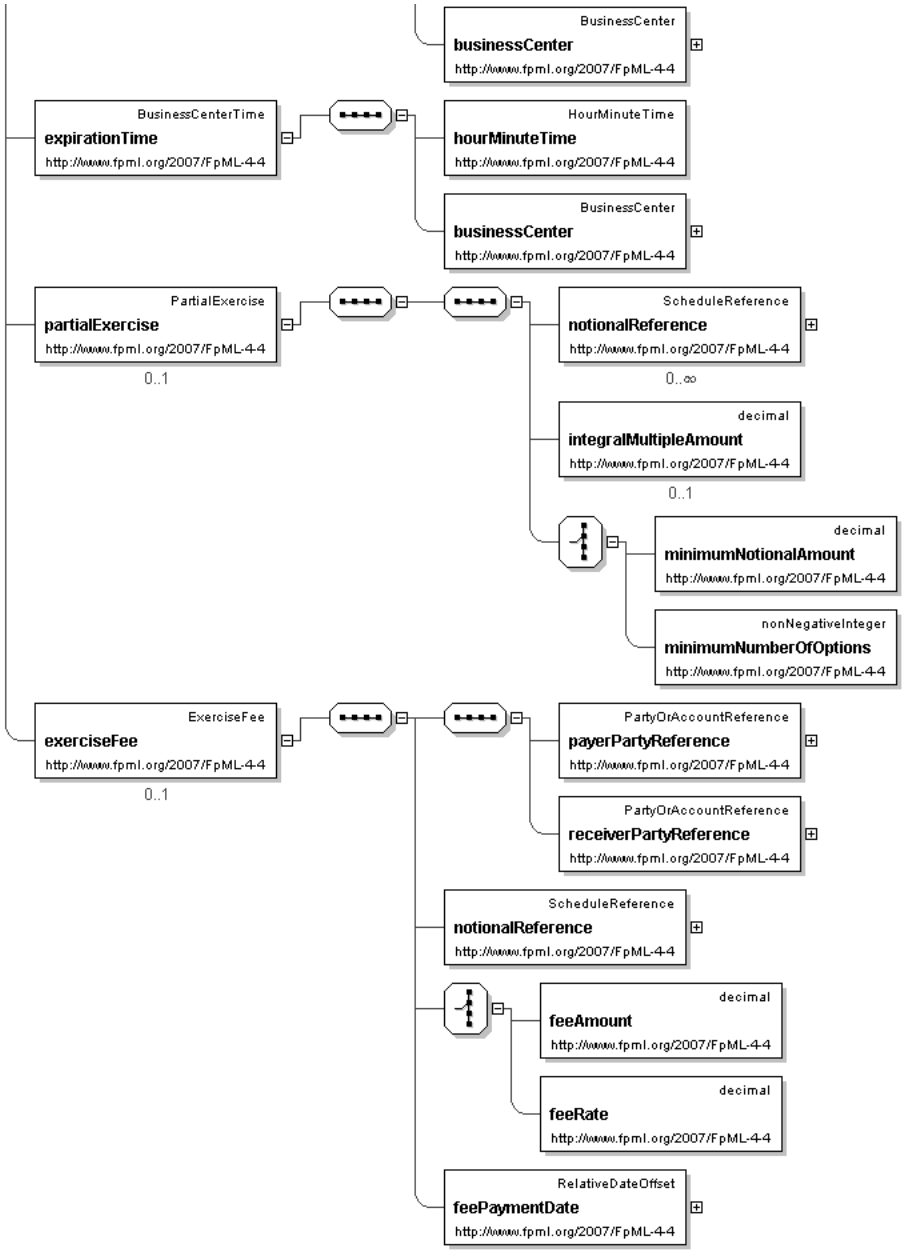
Element: europeanExercise

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

Name	europeanExercise
Type	EuropeanExercise
Nillable	no
Abstract	no
Documentation	The parameters for defining the exercise period for a European style option together with any rules governing the notional amount of the underlying which can be exercised on any given exercise date and any associated exercise fees.

Logical Diagram





XML Instance Representation

```
<europeanExercise
id=" xsd:ID [0..1]">
   <expirationDate> AdjustableOrRelativeDate </expirationDate> [1]
   'The last day within an exercise period for an American style option. For a European
   style option it is the only day within the exercise period.'

   <relevantUnderlyingDate> AdjustableOrRelativeDates </relevantUnderlyingDate> [0..1]
   'The daye on the underlying set by the exercise of an option. What this date is depends on
   the option (e.g. in a swaption it is the effective date, in an extendible/cancelable
   provision it is the termination date).'
```

<earliestExerciseTime> BusinessCenterTime </earliestExerciseTime> [1]

'The earliest time at which notice of exercise can be given by the buyer to the seller (or seller\'s agent) i) on the expriation date, in the case of a European style option, (ii) on each bermuda option exercise date and the expiration date, in the case of a Bermuda style option the commencement date to, and including, the expiration date , in the case of an American option.'

<expirationTime> BusinessCenterTime </expirationTime> [1]

'The latest time for exercise on expirationDate.'

<partialExercise> PartialExercise </partialExercise> [0..1]

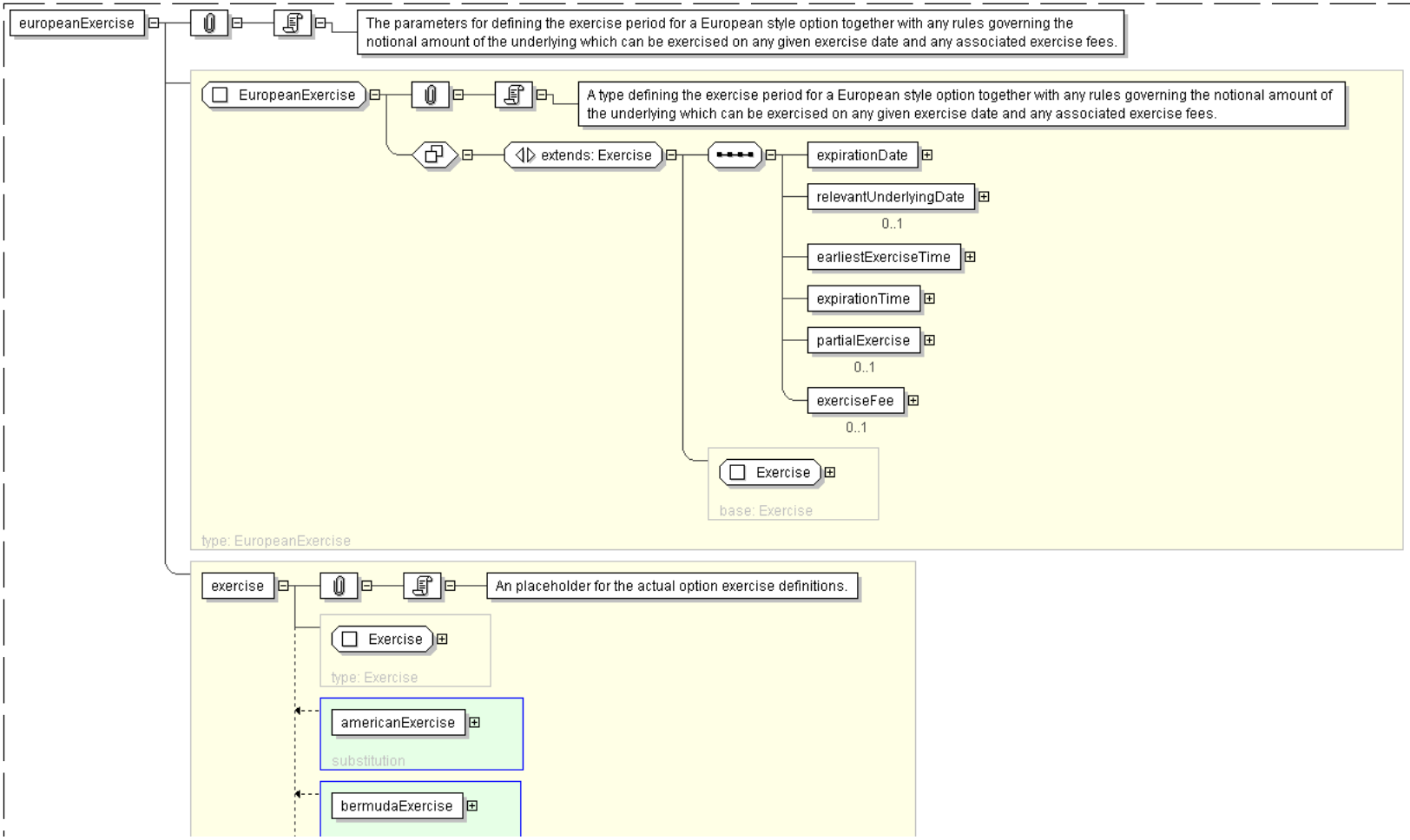
'As defined in the 2000 ISDA Definitions, Section 12.3. Partial Exercise, the buyer of the option has the right to exercise all or less than all the notional amount of the underlying swap on the expiration date, but may not exercise less than the minimum notional amount, and if an integral multiple amount is specified, the notional amount exercised must be equal to, or be an integral multiple of, the integral multiple amount.'

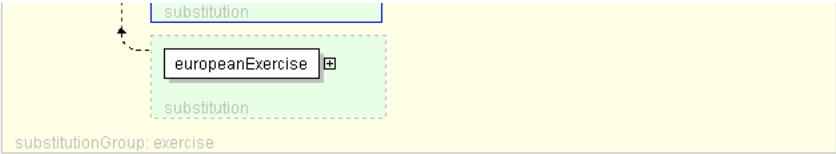
<exerciseFee> ExerciseFee </exerciseFee> [0..1]

'A fee to be paid on exercise. This could be represented as an amount or a rate and notional reference on which to apply the rate.'

</europeanExercise>

Diagram





Schema Component Representation

```
<xsd:element name="europeanExercise" type=" EuropeanExercise " substitutionGroup="exercise"/>
```

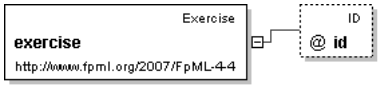
top

Element: exercise

- *The following elements can be used wherever this element is referenced:*
  - americanExercise
  - bermudaExercise
  - europeanExercise

Name	exercise
Type	Exercise
Nillable	no
Abstract	yes
Documentation	An placeholder for the actual option exercise definitions.

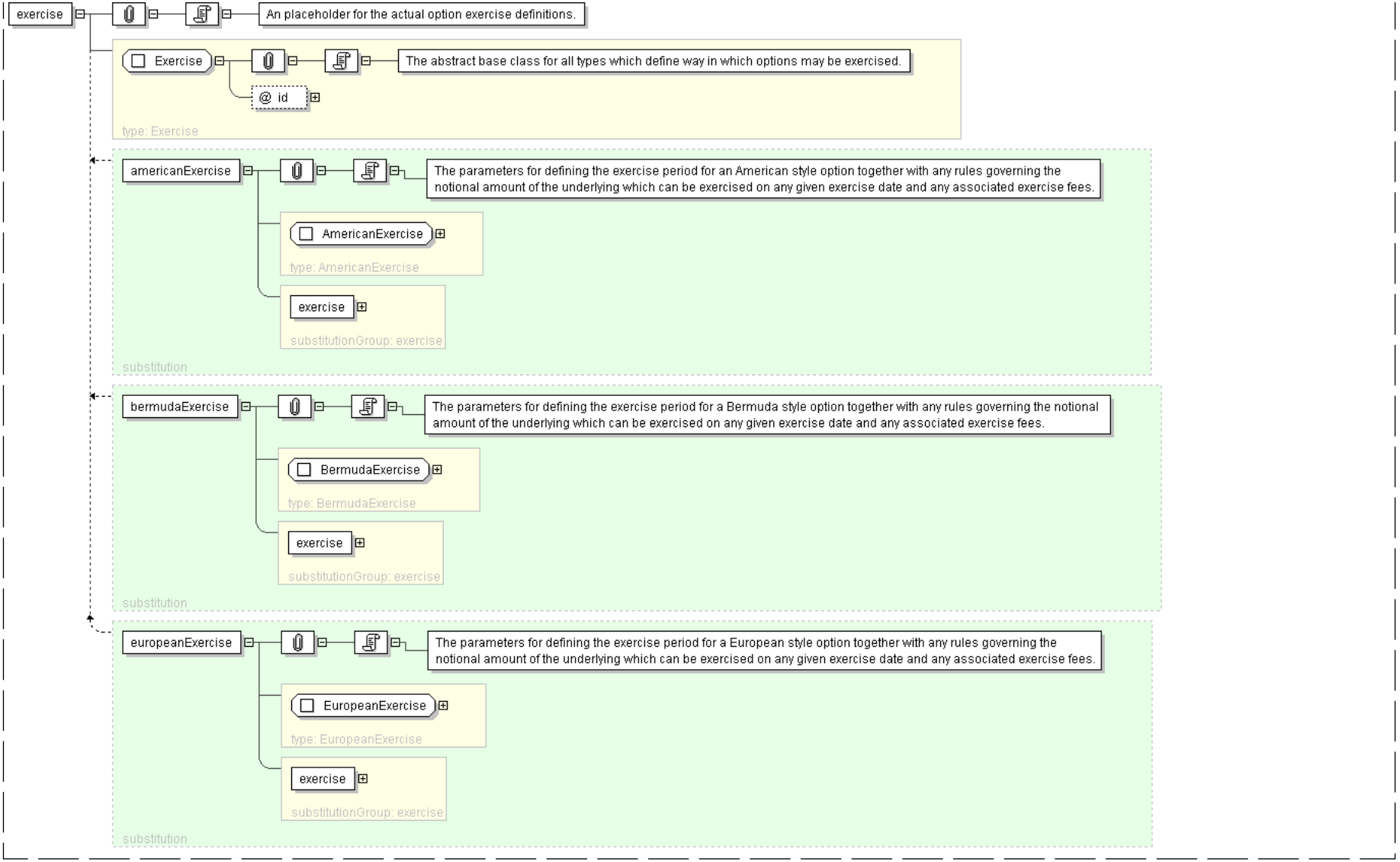
Logical Diagram



XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:element name="exercise" type=" Exercise " abstract="true"/>
```

top

Element: product

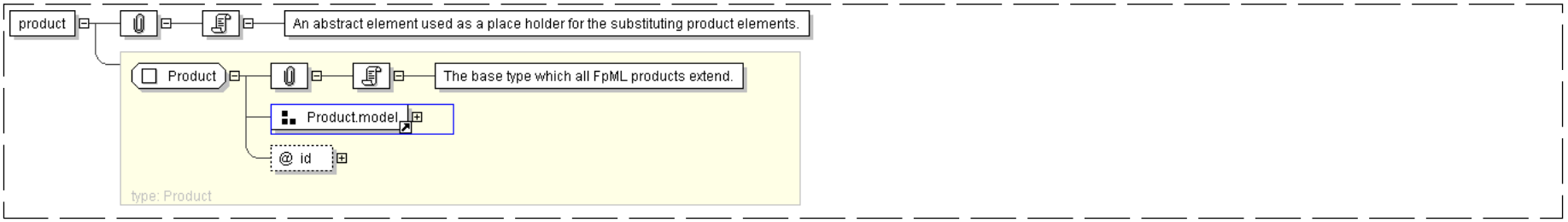
Name	product
Type	Product
Nillable	no
Abstract	yes
Documentation	An abstract element used as a place holder for the substituting product elements.

Logical Diagram

XML Instance Representation

```
<product
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.'

</product>
```

Diagram



Schema Component Representation

```
<xsd:element name="product" type=" Product " abstract="true"/>
```

top

Global Definitions

Complex Type: Account

Super-types:	None
Sub-types:	None

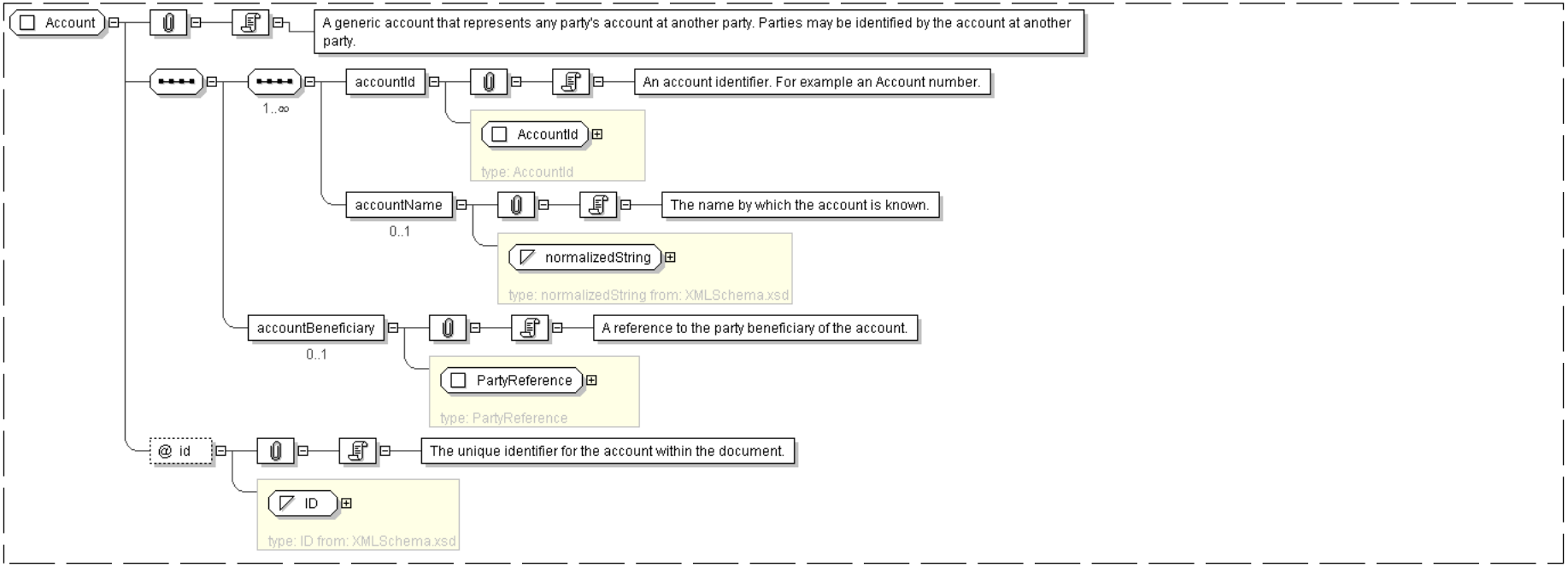
Name	Account
Used by (from the same schema document)	Complex Type **Party**
Abstract	no
Documentation	A generic account that represents any party's account at another party. Parties may be identified by the account at another party.

XML Instance Representation

```
<...
id=" xsd:ID [1]
'The unique identifier for the account within the document.'
```

```
">
Start Sequence [1..*]
      <accountId> AccountId </accountId> [1]
      'An account identifier. For example an Account number.'

      <accountName> xsd:normalizedString </accountName> [0..1]
      'The name by which the account is known.'

End Sequence
   <accountBeneficiary> PartyReference </accountBeneficiary> [0..1]
   'A reference to the party beneficiary of the account.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Account">
   <xsd:sequence>
      <xsd:sequence maxOccurs="unbounded">
         <xsd:element name="accountId" type=" AccountId "/>
         <xsd:element name="accountName" type=" xsd:normalizedString " minOccurs="0"/>
      </xsd:sequence>
      <xsd:element name="accountBeneficiary" type=" PartyReference " minOccurs="0"/>
   </xsd:sequence>
   <xsd:attribute name="id" type=" xsd:ID " use="required"/>
</xsd:complexType>
```

top

Complex Type: AccountId

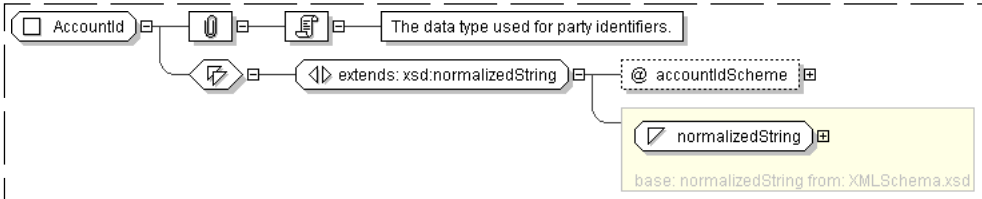
Super-types:	xsd:normalizedString < **AccountId** (by extension)
Sub-types:	None

Name	AccountId
Used by (from the same schema document)	Complex Type **Account**
Abstract	no
Documentation	The data type used for party identifiers.

XML Instance Representation

```
<...
accountIdScheme=" xsd:anyURI [0..1]
 'The identifier scheme used with this accountId. A unique URI to determine the
 authoritative issuer of these identifiers.'

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

Diagram



Schema Component Representation

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

top

Complex Type: AccountReference

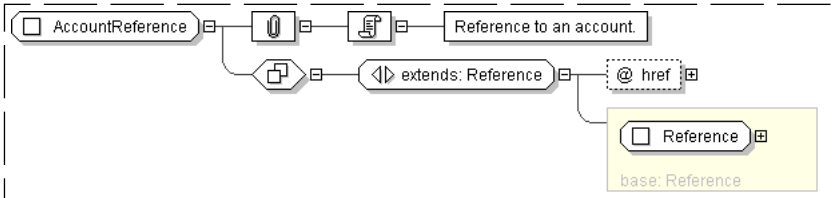
Super-types:	Reference < **AccountReference** (by extension)
Sub-types:	None

Name	AccountReference
Abstract	no
Documentation	Reference to an account.

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="AccountReference">
```

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

top

Complex Type: Address

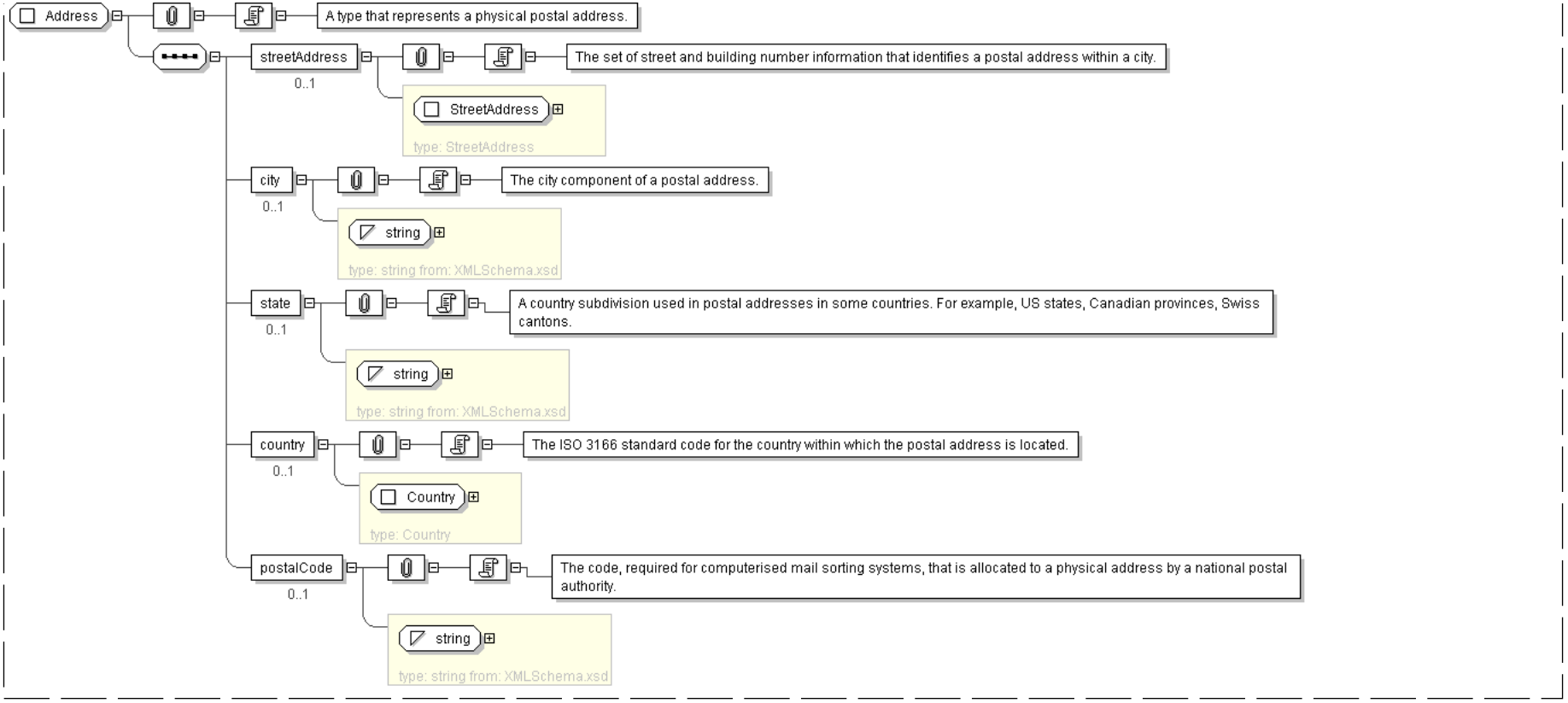
Super-types:	None
Sub-types:	None

Name	Address
Used by (from the same schema document)	Model Group **RoutingExplicitDetails.model**
Abstract	no
Documentation	A type that represents a physical postal address.

XML Instance Representation

```
<...>
   <streetAddress> StreetAddress </streetAddress> [0..1]
   'The set of street and building number information that identifies a postal address within
   a city.'

   <city> xsd:string </city> [0..1]
   'The city component of a postal address.'

   <state> xsd:string </state> [0..1]
   'A country subdivision used in postal addresses in some countries. For example, US
   states, Canadian provinces, Swiss cantons.'

   <country> Country </country> [0..1]
   'The ISO 3166 standard code for the country within which the postal address is located.'

   <postalCode> xsd:string </postalCode> [0..1]
   'The code, required for computerised mail sorting systems, that is allocated to a
   physical address by a national postal authority.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Address">
   <xsd:sequence>
      <xsd:element name="streetAddress" type=" StreetAddress " minOccurs="0"/>
      <xsd:element name="city" type=" xsd:string " minOccurs="0"/>
      <xsd:element name="state" type=" xsd:string " minOccurs="0"/>
      <xsd:element name="country" type=" Country " minOccurs="0"/>
      <xsd:element name="postalCode" type=" xsd:string " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: AdjustableDate

Super-types:	None
Sub-types:	None

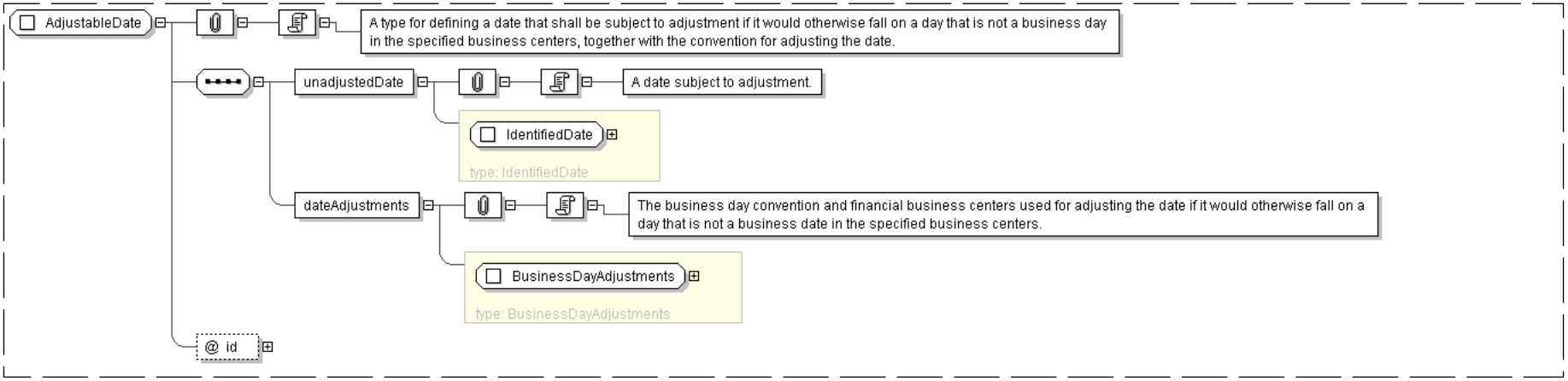
Name	AdjustableDate
Used by (from the same schema document)	Complex Type **AdjustableOrRelativeDate** , Complex Type **DividendPaymentDate** , Complex Type **Payment**
Abstract	no
Documentation	A type for defining a date that shall be subject to adjustment if it would otherwise fall on a day that is not a business day in the specified business centers, together with the convention for adjusting the date.

XML Instance Representation

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

```
    'A date subject to adjustment.'

    <dateAdjustments> BusinessDayAdjustments </dateAdjustments> [1]
    'The business day convention and financial business centers used for adjusting the date if
    it would otherwise fall on a day that is not a business date in the specified business centers.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="AdjustableDate">
   <xsd:sequence>
      <xsd:element name="unadjustedDate" type=" IdentifiedDate "/>
      <xsd:element name="dateAdjustments" type=" BusinessDayAdjustments "/>
   </xsd:sequence>
   <xsd:attribute name="id" type=" xsd:ID "/>
</xsd:complexType>
```

top

Complex Type: AdjustableDate2

Super-types:	None
Sub-types:	None

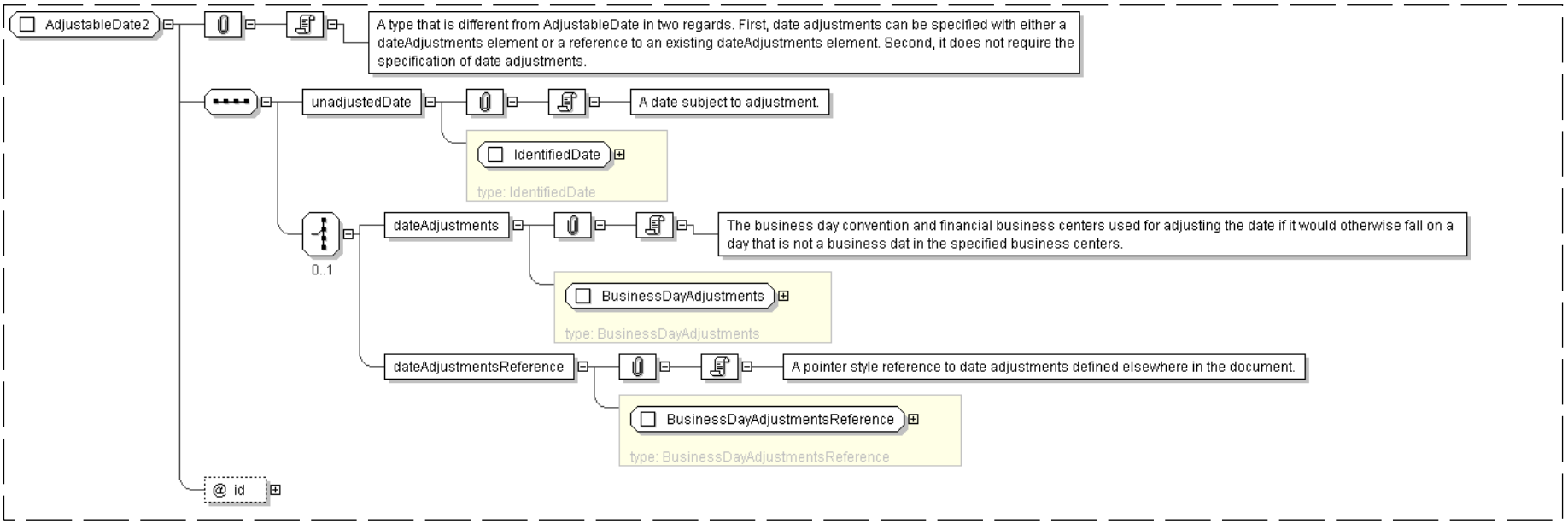
Name	AdjustableDate2
Abstract	no
Documentation	A type that is different from AdjustableDate in two regards. First, date adjustments can be specified with either a dateAdjustments element or a reference to an existing dateAdjustments element. Second, it does not require the specification of date adjustments.

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
   <unadjustedDate> IdentifiedDate </unadjustedDate> [1]
   'A date subject to adjustment.'

Start Choice [0..1]
      <dateAdjustments> BusinessDayAdjustments </dateAdjustments> [1]
      'The business day convention and financial business centers used for adjusting the date if
      it would otherwise fall on a day that is not a business dat in the specified business centers.'

      <dateAdjustmentsReference> BusinessDayAdjustmentsReference </dateAdjustmentsReference> [1]
      'A pointer style reference to date adjustments defined elsewhere in the document.'
```

```
End Choice
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="AdjustableDate2">
   <xsd:sequence>
      <xsd:element name="unadjustedDate" type=" IdentifiedDate "/>
      <xsd:choice minOccurs="0">
         <xsd:element name="dateAdjustments" type=" BusinessDayAdjustments "/>
         <xsd:element name="dateAdjustmentsReference" type=" BusinessDayAdjustmentsReference "/>
      </xsd:choice>
   </xsd:sequence>
   <xsd:attribute name="id" type=" xsd:ID "/>
</xsd:complexType>
```

top

Complex Type: AdjustableDates

Super-types:	None
Sub-types:	None

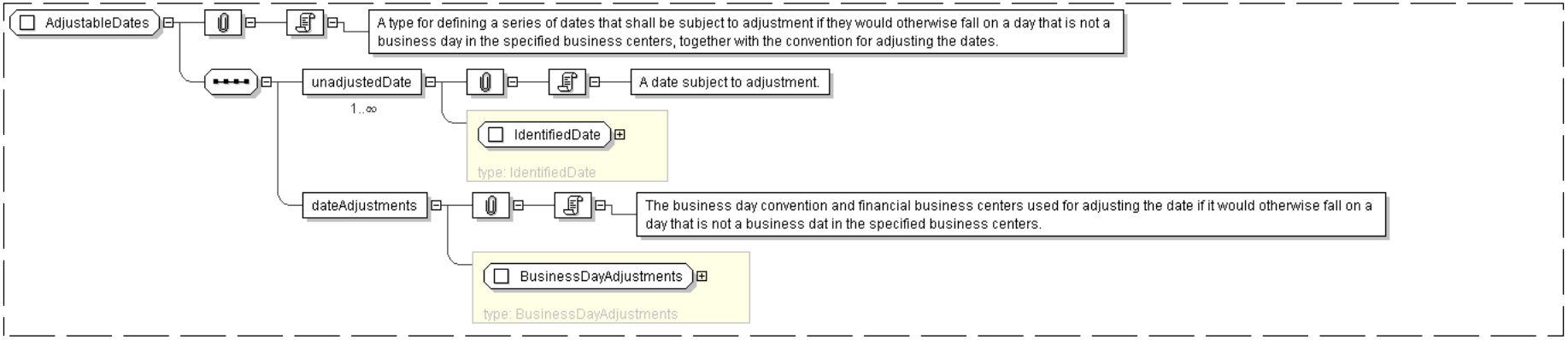
Name	AdjustableDates
Used by (from the same schema document)	Complex Type **AdjustableDatesOrRelativeDateOffset** , Complex Type **AdjustableOrRelativeDates** , Complex Type **AdjustableRelativeOrPeriodicDates**
Abstract	no
Documentation	A type for defining a series of dates that shall be subject to adjustment if they would otherwise fall on a day that is not a business day in the specified business centers, together with the convention for adjusting the dates.

XML Instance Representation

```
<...>
   <unadjustedDate> IdentifiedDate </unadjustedDate> [1..*]
   'A date subject to adjustment.'

   <dateAdjustments> BusinessDayAdjustments </dateAdjustments> [1]
   'The business day convention and financial business centers used for adjusting the date if
```

```
    it would otherwise fall on a day that is not a business dat in the specified business centers.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="AdjustableDates">
   <xsd:sequence>
      <xsd:element name="unadjustedDate" type=" IdentifiedDate " maxOccurs="unbounded"/>
      <xsd:element name="dateAdjustments" type=" BusinessDayAdjustments "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: AdjustableDatesOrRelativeDateOffset

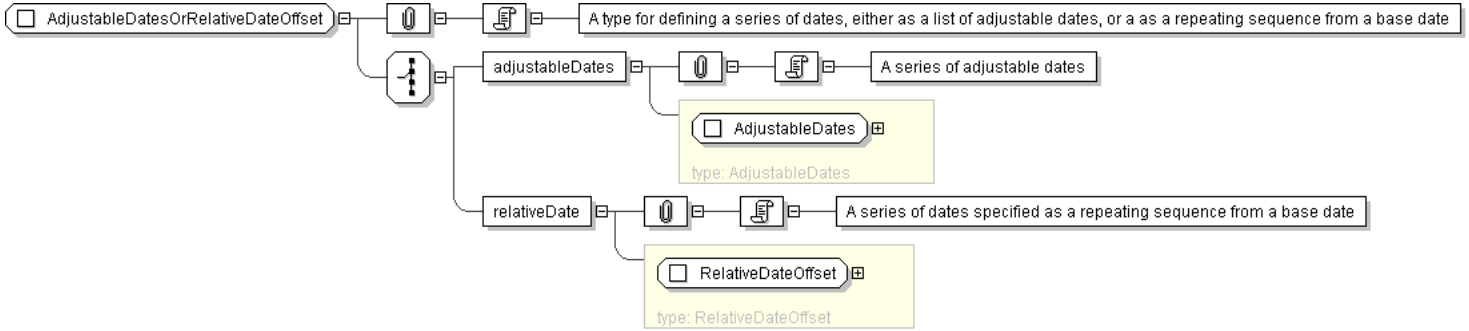
Super-types:	None
Sub-types:	None

Name	AdjustableDatesOrRelativeDateOffset
Abstract	no
Documentation	A type for defining a series of dates, either as a list of adjustable dates, or a as a repeating sequence from a base date

XML Instance Representation

```
<...>
Start Choice [1]
      <adjustableDates> AdjustableDates </adjustableDates> [1]
      'A series of adjustable dates'

      <relativeDate> RelativeDateOffset </relativeDate> [1]
      'A series of dates specified as a repeating sequence from a base date'

End Choice
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="AdjustableDatesOrRelativeDateOffset">
   <xsd:choice>
      <xsd:element name="adjustableDates" type=" AdjustableDates "/>
      <xsd:element name="relativeDate" type=" RelativeDateOffset "/>
   </xsd:choice>
</xsd:complexType>
```

top

Complex Type: AdjustableOrRelativeAndAdjustedDate

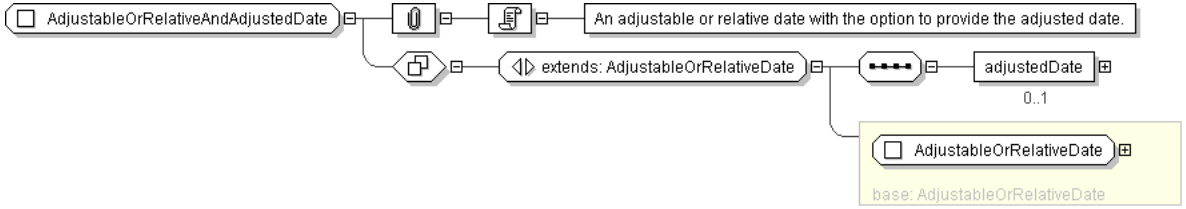
Super-types:	AdjustableOrRelativeDate < **AdjustableOrRelativeAndAdjustedDate** (by extension)
Sub-types:	None

Name	AdjustableOrRelativeAndAdjustedDate
Used by (from the same schema document)	Complex Type **SimplePayment**
Abstract	no
Documentation	An adjustable or relative date with the option to provide the adjusted date.

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
Start Choice [1]
      <adjustableDate> AdjustableDate </adjustableDate> [1]
      'A date that shall be subject to adjustment if it would otherwise fall on a day that is not
      a business day in the specified business centers, together with the convention for
      adjusting the date.'

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

End Choice
   <adjustedDate> IdentifiedDate </adjustedDate> [0..1]
   'The adjusted date. This date should already be adjusted for any applicable business
   day convention. This component is not intended for use in trade confirmation but my
   be specified to allow the fee structure to also serve as a cashflow type component (all
   dates the the Cashflows type are adjusted payment dates).'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="AdjustableOrRelativeAndAdjustedDate">
   <xsd:complexContent>
      <xsd:extension base=" AdjustableOrRelativeDate ">
         <xsd:sequence>
            <xsd:element name="adjustedDate" type=" IdentifiedDate " minOccurs="0"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: AdjustableOrRelativeDate

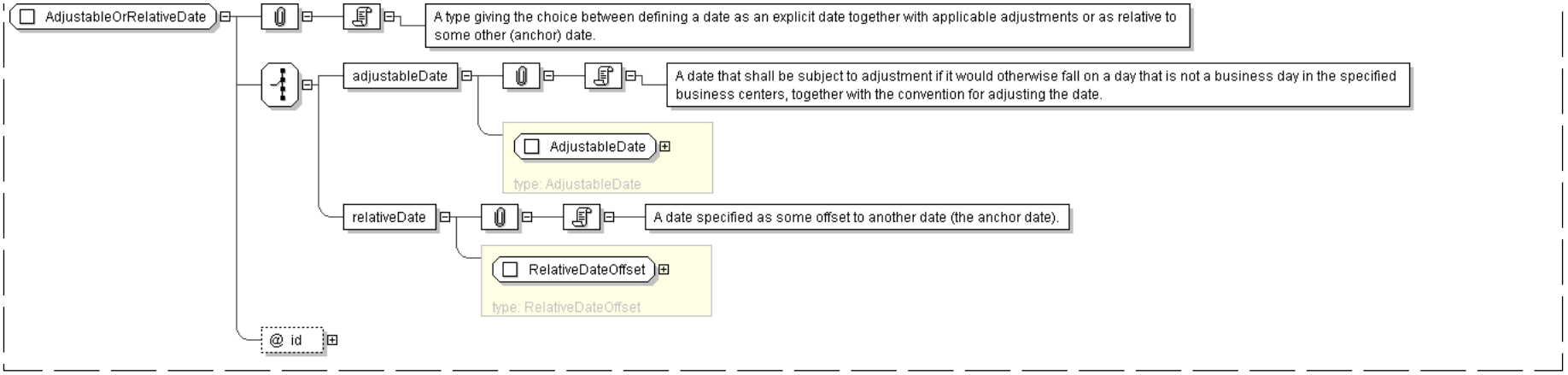
Super-types:	None
Sub-types:	• AdjustableOrRelativeAndAdjustedDate (by extension)

Name	AdjustableOrRelativeDate
Used by (from the same schema document)	Complex Type **AmericanExercise** , Complex Type **AmericanExercise** , Complex Type **EuropeanExercise** , Complex Type **PeriodicDates** , Complex Type **PeriodicDates** , Complex Type **SharedAmericanExercise** , Complex Type **SharedAmericanExercise** , Complex Type **Stub** , Complex Type **Stub**
Abstract	no
Documentation	A type giving the choice between defining a date as an explicit date together with applicable adjustments or as relative to some other (anchor) date.

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
Start Choice [1]
      <adjustableDate> AdjustableDate </adjustableDate> [1]
      'A date that shall be subject to adjustment if it would otherwise fall on a day that is not
      a business day in the specified business centers, together with the convention for
      adjusting the date.'

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

End Choice
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="AdjustableOrRelativeDate">
   <xsd:choice>
      <xsd:element name="adjustableDate" type=" AdjustableDate "/>
      <xsd:element name="relativeDate" type=" RelativeDateOffset "/>
   </xsd:choice>
   <xsd:attribute name="id" type=" xsd:ID "/>
</xsd:complexType>
```

top

Complex Type: AdjustableOrRelativeDates

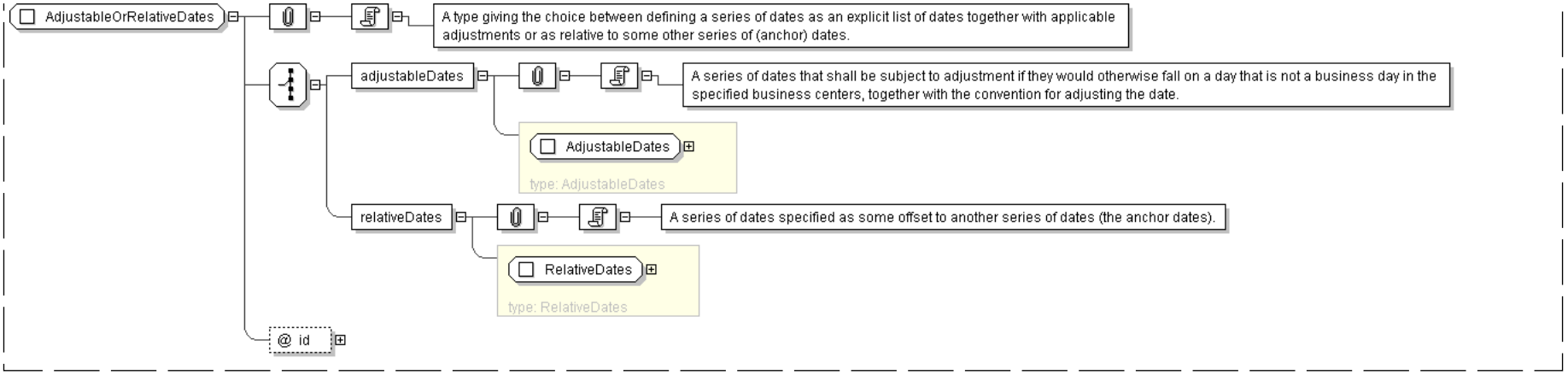
Super-types:	None
Sub-types:	None

Name	AdjustableOrRelativeDates
Used by (from the same schema document)	Complex Type **AmericanExercise** , Complex Type **BermudaExercise** , Complex Type **BermudaExercise** , Complex Type **EuropeanExercise**
Abstract	no
Documentation	A type giving the choice between defining a series of dates as an explicit list of dates together with applicable adjustments or as relative to some other series of (anchor) dates.

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
Start Choice [1]
      <adjustableDates> AdjustableDates </adjustableDates> [1]
      'A series of dates that shall be subject to adjustment if they would otherwise fall on a
      day that is not a business day in the specified business centers, together with the
      convention for adjusting the date.'

      <relativeDates> RelativeDates </relativeDates> [1]
      'A series of dates specified as some offset to another series of dates (the anchor dates).'

End Choice
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="AdjustableOrRelativeDates">
   <xsd:choice>
      <xsd:element name="adjustableDates" type=" AdjustableDates "/>
      <xsd:element name="relativeDates" type=" RelativeDates "/>
   </xsd:choice>
   <xsd:attribute name="id" type=" xsd:ID "/>
</xsd:complexType>
```

top

Complex Type: AdjustableRelativeOrPeriodicDates

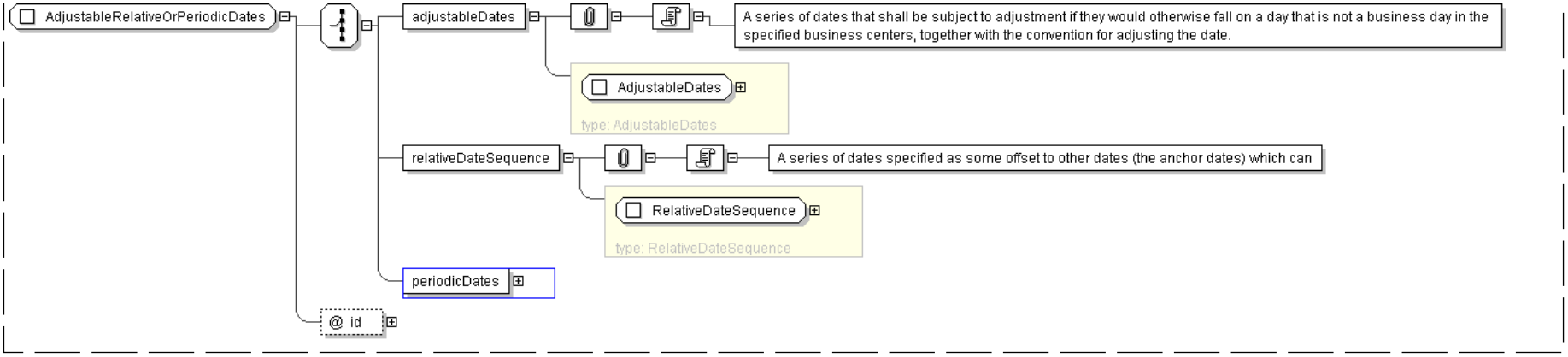
Super-types:	None
Sub-types:	None

Name	AdjustableRelativeOrPeriodicDates
Abstract	no

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
Start Choice [1]
      <adjustableDates> AdjustableDates </adjustableDates> [1]
      'A series of dates that shall be subject to adjustment if they would otherwise fall on a
      day that is not a business day in the specified business centers, together with the
      convention for adjusting the date.'

      <relativeDateSequence> RelativeDateSequence </relativeDateSequence> [1]
      'A series of dates specified as some offset to other dates (the anchor dates) which can'

      <periodicDates> PeriodicDates </periodicDates> [1]
End Choice
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="AdjustableRelativeOrPeriodicDates">
   <xsd:choice>
      <xsd:element name="adjustableDates" type=" AdjustableDates "/>
      <xsd:element name="relativeDateSequence" type=" RelativeDateSequence "/>
      <xsd:element name="periodicDates" type=" PeriodicDates "/>
   </xsd:choice>
   <xsd:attribute name="id" type=" xsd:ID "/>
</xsd:complexType>
```

top

Complex Type: AdjustedRelativeDateOffset

Super-types:	Interval < Offset (by extension) < RelativeDateOffset (by extension) < **AdjustedRelativeDateOffset** (by extension)
Sub-types:	None

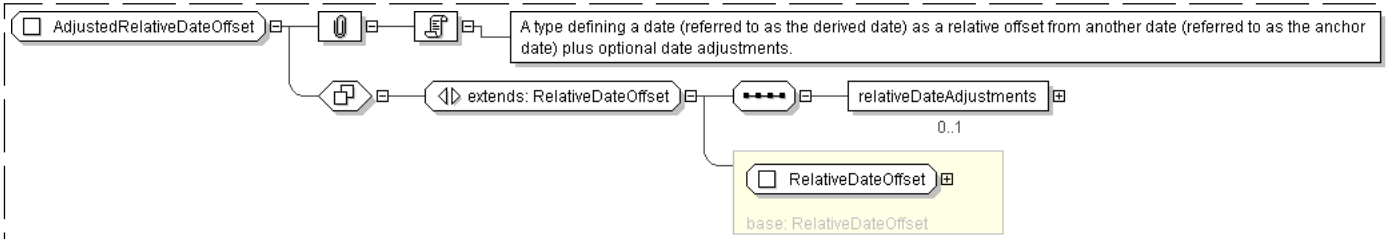
Name	AdjustedRelativeDateOffset
Abstract	no
Documentation	A type defining a date (referred to as the derived date) as a relative offset from another date (referred to as the anchor date) plus optional date adjustments.

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
   <periodMultiplier> xsd:integer </periodMultiplier> [1]
   'A time period multiplier, e.g. 1, 2 or 3 etc. A negative value can be used when specifying
   an offset relative to another date, e.g. -2 days. If the period value is T (Term)
   then periodMultiplier must contain the value 1.'

   <period> PeriodEnum </period> [1]
   'A time period, e.g. a day, week, month, year or term of the stream. If the
   periodMultiplier value is 0 (zero) then period must contain the value D (day).'

   <dayType> DayTypeEnum </dayType> [0..1]
   'In the case of an offset specified as a number of days, this element defines
   whether consideration is given as to whether a day is a good business day or not. If a day
   type of business days is specified then non-business days are ignored when calculating
   the offset. The financial business centers to use for determination of business days
   are implied by the context in which this element is used. This element must only be
   included when the offset is specified as a number of days. If the offset is zero days then
   the dayType element should not be included.'

   <businessDayConvention> BusinessDayConventionEnum </businessDayConvention> [1]
   'The convention for adjusting a date if it would otherwise fall on a day that is not a
   business day.'
```

```
Start Group: BusinessCentersOrReference.model [0..1]
Start Choice [1]
        <businessCentersReference> BusinessCentersReference </businessCentersReference> [1]
        'A pointer style reference to a set of financial business centers defined elsewhere in
        the document. This set of business centers is used to determine whether a particular day is
        a business day or not.'

        <businessCenters> BusinessCenters </businessCenters> [1]
End Choice
End Group: BusinessCentersOrReference.model
   <dateRelativeTo> DateReference </dateRelativeTo> [1]
   'Specifies the anchor as an href attribute. The href attribute value is a pointer
   style reference to the element or component elsewhere in the document where the anchor date
   is defined.'

   <relativeDateAdjustments> BusinessDayAdjustments </relativeDateAdjustments> [0..1]
   'The business day convention and financial business centers used for adjusting the
   relative date if it would otherwise fall on a day that is not a business date in the
   specified business centers.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="AdjustedRelativeDateOffset">
   <xsd:complexContent>
      <xsd:extension base=" RelativeDateOffset ">
         <xsd:sequence>
            <xsd:element name="relativeDateAdjustments" type=" BusinessDayAdjustments " minOccurs="0"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: AmericanExercise

Super-types:	Exercise < **AmericanExercise** (by extension)
Sub-types:	None

Name	AmericanExercise
Used by (from the same schema document)	Element **americanExercise**
Abstract	no
Documentation	A type defining the exercise period for an American style option together with any rules governing the notional amount of the underlying which can be exercised on any given exercise date and any associated exercise fees.

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
   <commencementDate> AdjustableOrRelativeDate </commencementDate> [1]
   'The first day of the exercise period for an American style option.'
```

<expirationDate> AdjustableOrRelativeDate **</expirationDate>** [1]

'The last day within an exercise period for an American style option. For a European style option it is the only day within the exercise period.'

<relevantUnderlyingDate> AdjustableOrRelativeDates **</relevantUnderlyingDate>** [0..1]

'The daye on the underlying set by the exercise of an option. What this date is depends on the option (e.g. in a swaption it is the effective date, in an extendible/cancelable provision it is the termination date).'

<earliestExerciseTime> BusinessCenterTime **</earliestExerciseTime>** [1]

'The earliest time at which notice of exercise can be given by the buyer to the seller (or seller\'s agent) i) on the expriation date, in the case of a European style option, (ii) on each bermuda option exercise date and the expiration date, in the case of a Bermuda style option the commencement date to, and including, the expiration date , in the case of an American option.'

<latestExerciseTime> BusinessCenterTime **</latestExerciseTime>** [0..1]

'For a Bermuda or American style option, the latest time on an exercise business day (excluding the expiration date) within the exercise period that notice can be given by the buyer to the seller or seller\'s agent. Notice of exercise given after this time will be deemed to have been given on the next exercise business day.'

<expirationTime> BusinessCenterTime **</expirationTime>** [1]

'The latest time for exercise on expirationDate.'

<multipleExercise> MultipleExercise **</multipleExercise>** [0..1]

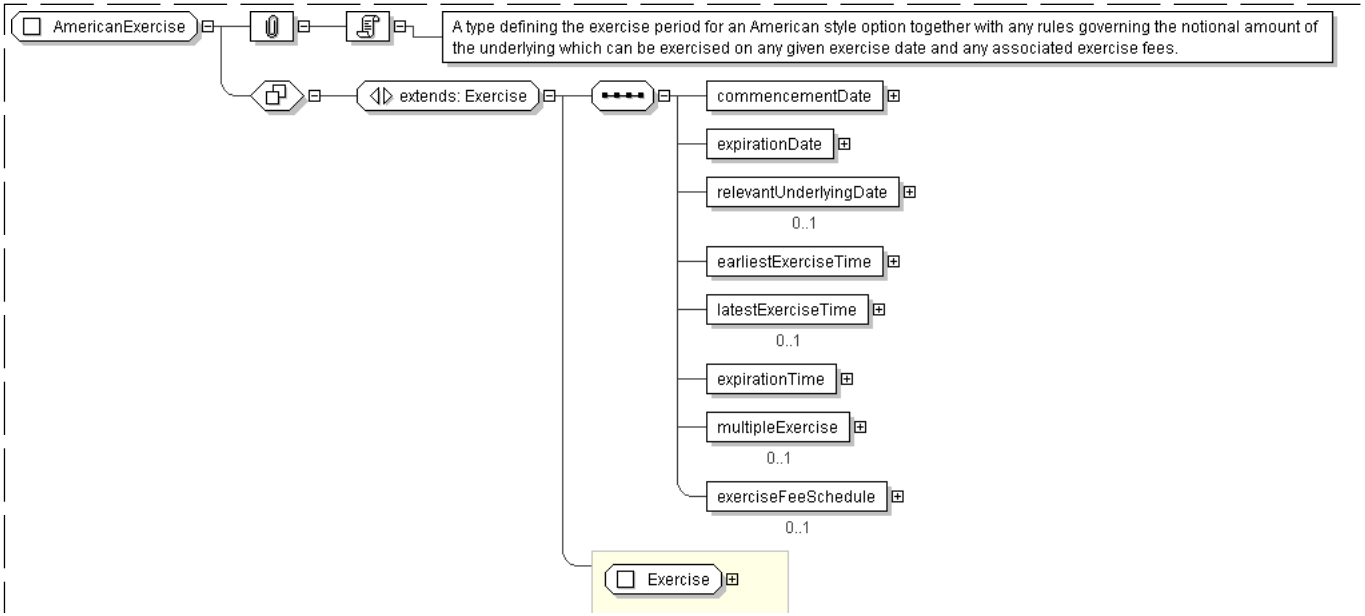
'As defined in the 2000 ISDA Definitions, Section 12.4. Multiple Exercise, the buyer of the option has the right to exercise all or less than all the unexercised notional amount of the underlying swap on one or more days in the exercise period, but on any such day may not exercise less than the minimum notional amount or more that the maximum notional amount, and if an integral multiple amount is specified, the notional amount exercised must be equal to, or be an intergral multiple of, the integral multiple amount.'

<exerciseFeeSchedule> ExerciseFeeSchedule **</exerciseFeeSchedule>** [0..1]

'The fees associated with an exercise date. The fees are conditional on the exercise occuring. The fees can be specified as actual currency amounts or as percentages of the notional amount being exercised.'

</...>

Diagram



base: Exercise

Schema Component Representation

```
<xsd:complexType name="AmericanExercise">
   <xsd:complexContent>
      <xsd:extension base=" Exercise ">
         <xsd:sequence>
            <xsd:element name="commencementDate" type=" AdjustableOrRelativeDate "/>
            <xsd:element name="expirationDate" type=" AdjustableOrRelativeDate "/>
            <xsd:element name="relevantUnderlyingDate" type=" AdjustableOrRelativeDates " minOccurs="0"/>
            <xsd:element name="earliestExerciseTime" type=" BusinessCenterTime "/>
            <xsd:element name="latestExerciseTime" type=" BusinessCenterTime " minOccurs="0"/>
            <xsd:element name="expirationTime" type=" BusinessCenterTime "/>
            <xsd:element name="multipleExercise" type=" MultipleExercise " minOccurs="0"/>
            <xsd:element name="exerciseFeeSchedule" type=" ExerciseFeeSchedule " minOccurs="0"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: AmountReference

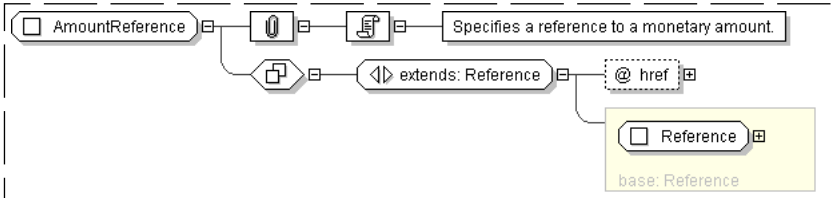
Super-types:	Reference < **AmountReference** (by extension)
Sub-types:	None

Name	AmountReference
Abstract	no
Documentation	Specifies a reference to a monetary amount.

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: AmountSchedule

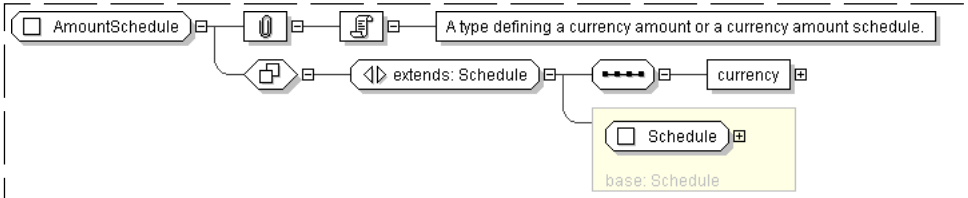
Super-types:	Schedule < **AmountSchedule** (by extension)
Sub-types:	None

Name	AmountSchedule
Used by (from the same schema document)	Complex Type **ExerciseFeeSchedule**
Abstract	no
Documentation	A type defining a currency amount or a currency amount schedule.

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
   <initialValue> xsd:decimal </initialValue> [1]
   'The initial rate or amount, as the case may be. An initial rate of 5% would be represented
   as 0.05.'

   <step> Step </step> [0..*]
   'The schedule of step date and value pairs. On each step date the associated step value
   becomes effective A list of steps may be ordered in the document by ascending step date.
   An FpML document containing an unordered list of steps is still regarded as a
   conformant document.'

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

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="AmountSchedule">
   <xsd:complexContent>
      <xsd:extension base=" Schedule ">
         <xsd:sequence>
            <xsd:element name="currency" type=" Currency "/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: AutomaticExercise

Super-types:	None
Sub-types:	None

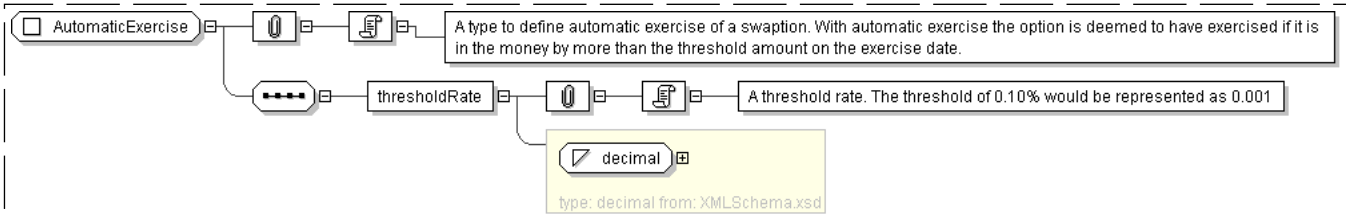
Name	AutomaticExercise
Used by (from the same schema document)	Complex Type **ExerciseProcedure**
Abstract	no
Documentation	A type to define automatic exercise of a swaption. With automatic exercise the option is deemed to have exercised if it is in the money by more than the threshold amount on the exercise date.

XML Instance Representation

```
<...>
   <thresholdRate> xsd:decimal </thresholdRate> [1]
   'A threshold rate. The threshold of 0.10% would be represented as 0.001'
```

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

Diagram



Schema Component Representation

```
<xsd:complexType name="AutomaticExercise">
   <xsd:sequence>
      <xsd:element name="thresholdRate" type=" xsd:decimal "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: Beneficiary

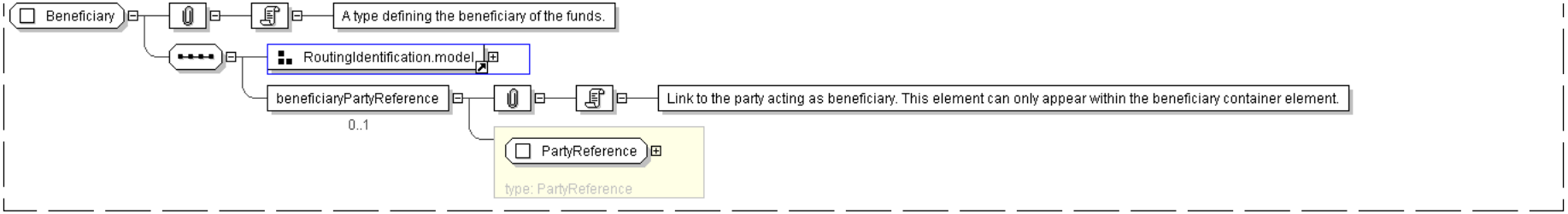
Super-types:	None
Sub-types:	None

Name	Beneficiary
Used by (from the same schema document)	Complex Type **SettlementInstruction** , Complex Type **SettlementInstruction**
Abstract	no
Documentation	A type defining the beneficiary of the funds.

XML Instance Representation

```
<...>
Start Choice [1]
      <routingIds> RoutingIds </routingIds> [1]
      'A set of unique identifiers for a party, eachone identifying the party within a
      payment system. The assumption is that each party will not have more than one identifier
      within the same payment system.'

      <routingExplicitDetails> RoutingExplicitDetails </routingExplicitDetails> [1]
      'A set of details that is used to identify a party involved in the routing of a payment
      when the party does not have a code that identifies it within one of the recognized
      payment systems.'

      <routingIdsAndExplicitDetails> RoutingIdsAndExplicitDetails </routingIdsAndExplicitDetails> [1]
      'A combination of coded payment system identifiers and details for physical addressing for
      a party involved in the routing of a payment.'

End Choice
   <beneficiaryPartyReference> PartyReference </beneficiaryPartyReference> [0..1]
   'Link to the party acting as beneficiary. This element can only appear within the
   beneficiary container element.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Beneficiary">
   <xsd:sequence>
      <xsd:group ref=" RoutingIdentification.model "/>
      <xsd:element name="beneficiaryPartyReference" type=" PartyReference " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: BermudaExercise

Super-types:	Exercise < **BermudaExercise** (by extension)
Sub-types:	None

Name	BermudaExercise
Used by (from the same schema document)	Element **bermudaExercise**
Abstract	no
Documentation	A type defining the Bermuda option exercise dates and the expiration date together with any rules govenerning the notional amount of the underlying which can be exercised on any given exercise date and any associated exercise fee.

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
   <bermudaExerciseDates> AdjustableOrRelativeDates </bermudaExerciseDates> [1]
   'The dates the define the Bermuda option exercise dates and the expiration date. The
   last specified date is assumed to be the expiration date. The dates can either be specified
   as a series of explicit dates and associated adjustments or as a series of dates
   defined relative to another schedule of dates, for example, the calculation period start
   dates. Where a relative series of dates are defined the first and last possible exercise
   dates can be separately specified.'

   <relevantUnderlyingDate> AdjustableOrRelativeDates </relevantUnderlyingDate> [0..1]
   'The daye on the underlying set by the exercise of an option. What this date is depends on
   the option (e.g. in a swaption it is the effective date, in an extendible/cancelable
   provision it is the termination date).'

   <earliestExerciseTime> BusinessCenterTime </earliestExerciseTime> [1]
   'The earliest time at which notice of exercise can be given by the buyer to the seller
   (or seller\'s agent) i) on the expriation date, in the case of a European style option, (ii)
   on each bermuda option exercise date and the expiration date, in the case of a Bermuda
   style option the commencement date to, and including, the expiration date , in the case of
   an American option.'

   <latestExerciseTime> BusinessCenterTime </latestExerciseTime> [0..1]
   'For a Bermuda or American style option, the latest time on an exercise business day
   (excluding the expiration date) within the exercise period that notice can be given by
   the buyer to the seller or seller\'s agent. Notice of exercise given after this time will
   be deemed to have been given on the next exercise business day.'

   <expirationTime> BusinessCenterTime </expirationTime> [1]
   'The latest time for exercise on expirationDate.'
```

```
<multipleExercise> MultipleExercise </multipleExercise> [0..1]
```

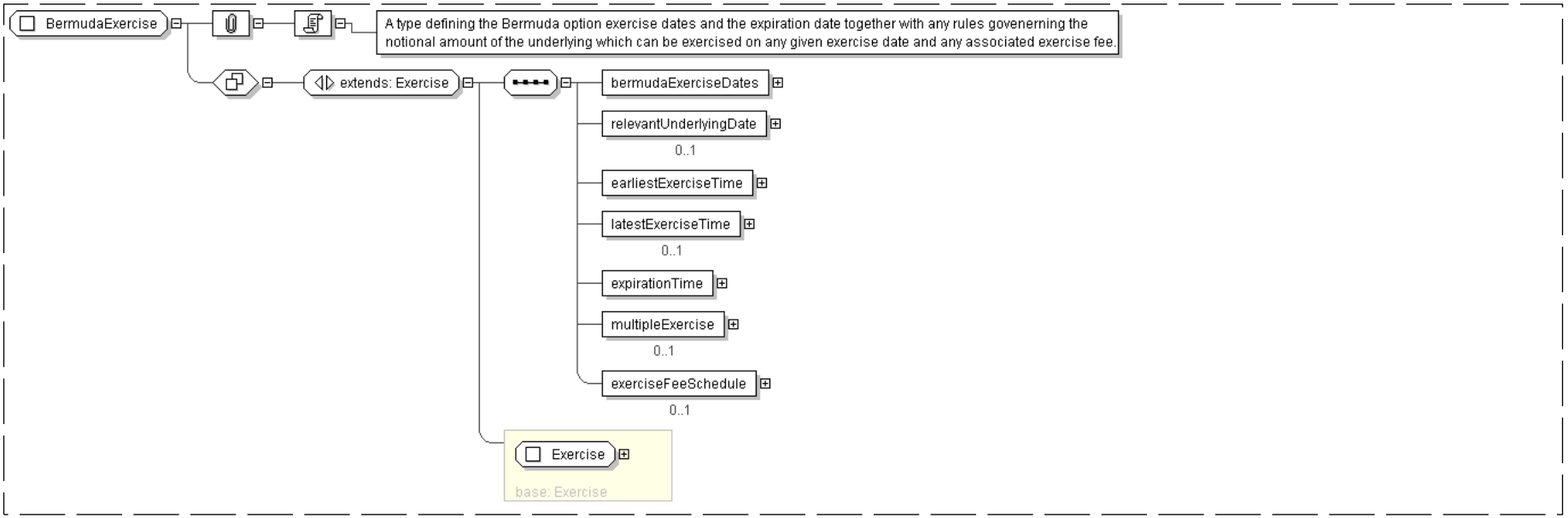
'As defined in the 2000 ISDA Definitions, Section 12.4. Multiple Exercise, the buyer of the option has the right to exercise all or less than all the unexercised notional amount of the underlying swap on one or more days in the exercise period, but on any such day may not exercise less than the minimum notional amount or more that the maximum notional amount, and if an integral multiple amount is specified, the notional amount exercised must be equal to, or be an intergral multiple of, the integral multiple amount.'

```
<exerciseFeeSchedule> ExerciseFeeSchedule </exerciseFeeSchedule> [0..1]
```

'The fees associated with an exercise date. The fees are conditional on the exercise occuring. The fees can be specified as actual currency amounts or as percentages of the notional amount being exercised.'

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

Diagram



Schema Component Representation

```
<xsd:complexType name="BermudaExercise">
   <xsd:complexContent>
      <xsd:extension base=" Exercise ">
         <xsd:sequence>
            <xsd:element name="bermudaExerciseDates" type=" AdjustableOrRelativeDates "/>
            <xsd:element name="relevantUnderlyingDate" type=" AdjustableOrRelativeDates " minOccurs="0"/>
            <xsd:element name="earliestExerciseTime" type=" BusinessCenterTime "/>
            <xsd:element name="latestExerciseTime" type=" BusinessCenterTime " minOccurs="0"/>
            <xsd:element name="expirationTime" type=" BusinessCenterTime "/>
            <xsd:element name="multipleExercise" type=" MultipleExercise " minOccurs="0"/>
            <xsd:element name="exerciseFeeSchedule" type=" ExerciseFeeSchedule " minOccurs="0"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: BrokerConfirmation

Super-types:	None

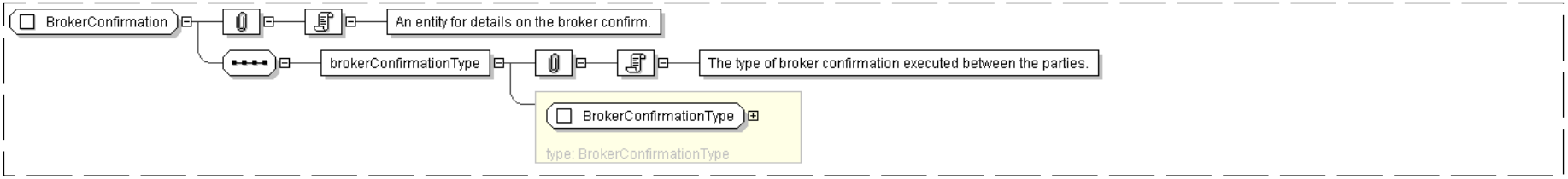
Sub-types:	None

Name	BrokerConfirmation
Used by (from the same schema document)	Complex Type **Documentation**
Abstract	no
Documentation	An entity for details on the broker confirm.

XML Instance Representation

```
<...>
   <brokerConfirmationType> BrokerConfirmationType </brokerConfirmationType> [1]
   'The type of broker confirmation executed between the parties.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="BrokerConfirmation">
   <xsd:sequence>
      <xsd:element name="brokerConfirmationType" type=" BrokerConfirmationType "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: BrokerConfirmationType

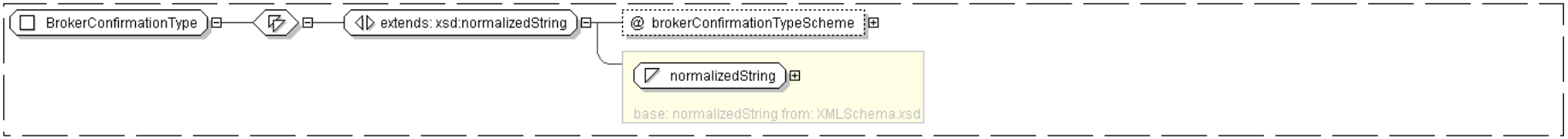
Super-types:	xsd:normalizedString < **BrokerConfirmationType** (by extension)
Sub-types:	None

Name	BrokerConfirmationType
Used by (from the same schema document)	Complex Type **BrokerConfirmation**
Abstract	no

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="BrokerConfirmationType">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="brokerConfirmationTypeScheme" type=" xsd:anyURI " default="http://www.
```

```
        fpml.org/coding-scheme/broker-confirmation-type-3-2"/>
      </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: BusinessCenter

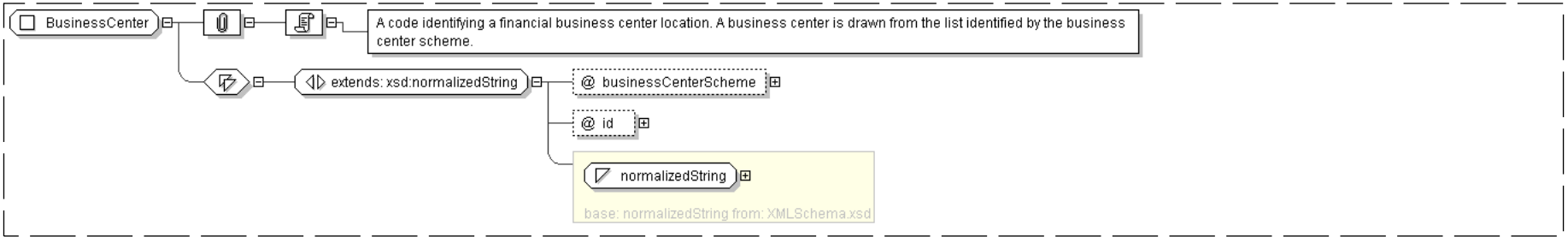
Super-types:	xsd:normalizedString < **BusinessCenter** (by extension)
Sub-types:	None

Name	BusinessCenter
Used by (from the same schema document)	Complex Type **BusinessCenters** , Complex Type **BusinessCenterTime** , Complex Type **ExerciseNotice**
Abstract	no
Documentation	A code identifying a financial business center location. A business center is drawn from the list identified by the business center scheme.

XML Instance Representation

```
<...
businessCenterScheme=" xsd:anyURI [0..1]"
id=" xsd:ID [0..1]">
xsd:normalizedString
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="BusinessCenter">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="businessCenterScheme" type=" xsd:anyURI " default="http://www.fpml.
         org/coding-scheme/business-center-6-4"/>
         <xsd:attribute name="id" type=" xsd:ID "/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: BusinessCenterTime

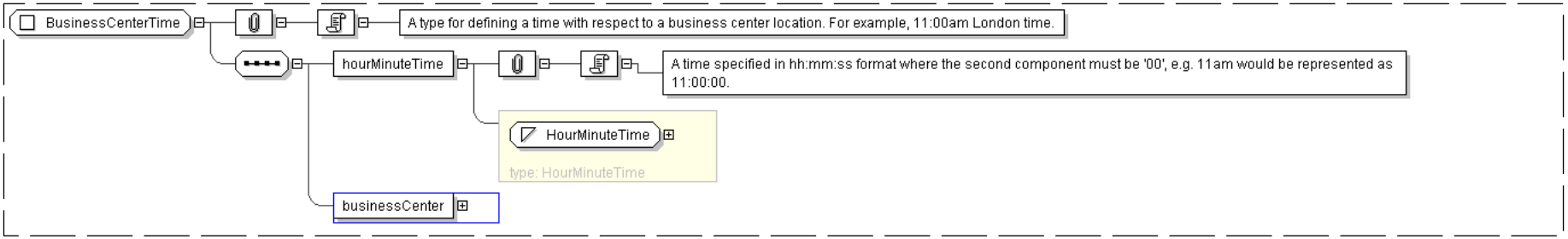
Super-types:	None
Sub-types:	None

Name	BusinessCenterTime
Used by (from the same schema document)	Complex Type **AmericanExercise** , Complex Type **AmericanExercise** , Complex Type **AmericanExercise** , Complex Type **BermudaExercise** , Complex Type **BermudaExercise** , Complex Type **BermudaExercise** , Complex Type **EuropeanExercise** , Complex Type **EuropeanExercise** , Complex Type **FxSpotRateSource** , Complex Type **SharedAmericanExercise**
Abstract	no
Documentation	A type for defining a time with respect to a business center location. For example, 11:00am London time.

XML Instance Representation

```
<...>
   <hourMinuteTime> HourMinuteTime </hourMinuteTime> [1]
   'A time specified in hh:mm:ss format where the second component must be \'00\', e.g. 11am
   would be represented as 11:00:00.'

   <businessCenter> BusinessCenter </businessCenter> [1]
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="BusinessCenterTime">
   <xsd:sequence>
      <xsd:element name="hourMinuteTime" type=" HourMinuteTime "/>
      <xsd:element name="businessCenter" type=" BusinessCenter "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: BusinessCenters

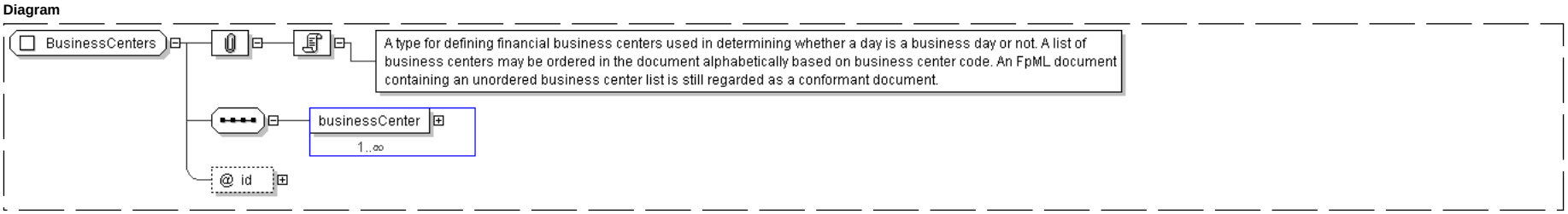
Super-types:	None
Sub-types:	None

Name	BusinessCenters
Used by (from the same schema document)	Model Group **BusinessCentersOrReference.model**
Abstract	no
Documentation	A type for defining financial business centers used in determining whether a day is a business day or not. A list of business centers may be ordered in the document alphabetically based on business center code. An FpML document containing an unordered business center list is still regarded as a conformant document.

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="BusinessCenters">
   <xsd:sequence>
      <xsd:element name="businessCenter" type=" BusinessCenter " maxOccurs="unbounded"/>
   </xsd:sequence>
   <xsd:attribute name="id" type=" xsd:ID "/>
</xsd:complexType>
```

top

Complex Type: BusinessCentersReference

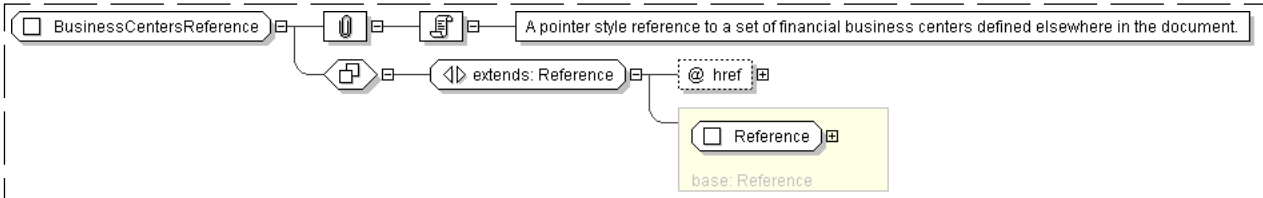
Super-types:	Reference < **BusinessCentersReference** (by extension)
Sub-types:	None

Name	BusinessCentersReference
Used by (from the same schema document)	Model Group **BusinessCentersOrReference.model**
Abstract	no
Documentation	A pointer style reference to a set of financial business centers defined elsewhere in the document.

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: BusinessDateRange

Super-types:	DateRange < **BusinessDateRange** (by extension)
Sub-types:	None

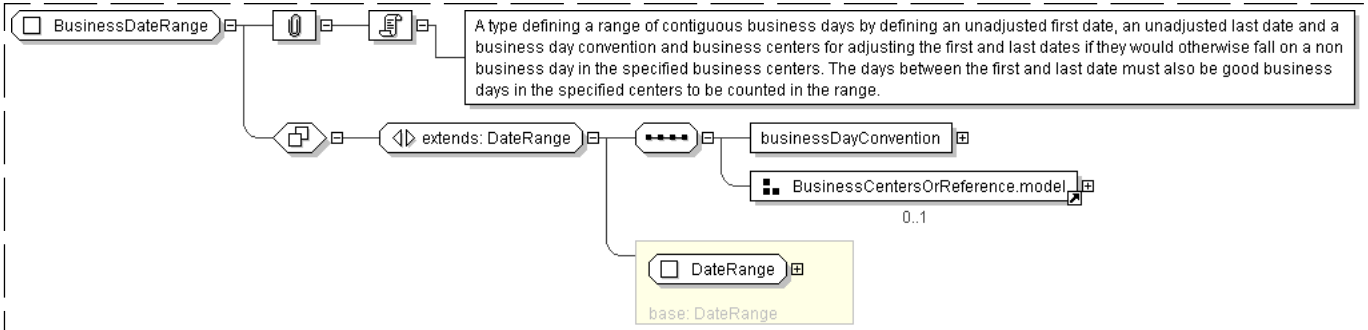
Name	BusinessDateRange
Abstract	no
Documentation	A type defining a range of contiguous business days by defining an unadjusted first date, an unadjusted last date and a business day convention and business centers for adjusting the first and last dates if they would otherwise fall on a non business day in the specified business centers. The days between the first and last date must also be good business days in the specified centers to be counted in the range.

XML Instance Representation

```
<...>
   <unadjustedFirstDate> xsd:date </unadjustedFirstDate> [1]
   'The first date of a date range.'
   <unadjustedLastDate> xsd:date </unadjustedLastDate> [1]
```

```
    'The last date of a date range.'

    <businessDayConvention> BusinessDayConventionEnum </businessDayConvention> [1]
    'The convention for adjusting a date if it would otherwise fall on a day that is not a
    business day.'

Start Group: BusinessCentersOrReference.model [0..1]
Start Choice [1]
        <businessCentersReference> BusinessCentersReference </businessCentersReference> [1]
        'A pointer style reference to a set of financial business centers defined elsewhere in
        the document. This set of business centers is used to determine whether a particular day is
        a business day or not.'

        <businessCenters> BusinessCenters </businessCenters> [1]
End Choice
End Group: BusinessCentersOrReference.model
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="BusinessDateRange">
   <xsd:complexContent>
      <xsd:extension base=" DateRange ">
         <xsd:sequence>
            <xsd:element name="businessDayConvention" type=" BusinessDayConventionEnum "/>
            <xsd:group ref=" BusinessCentersOrReference.model " minOccurs="0"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: BusinessDayAdjustments

Super-types:	None
Sub-types:	None

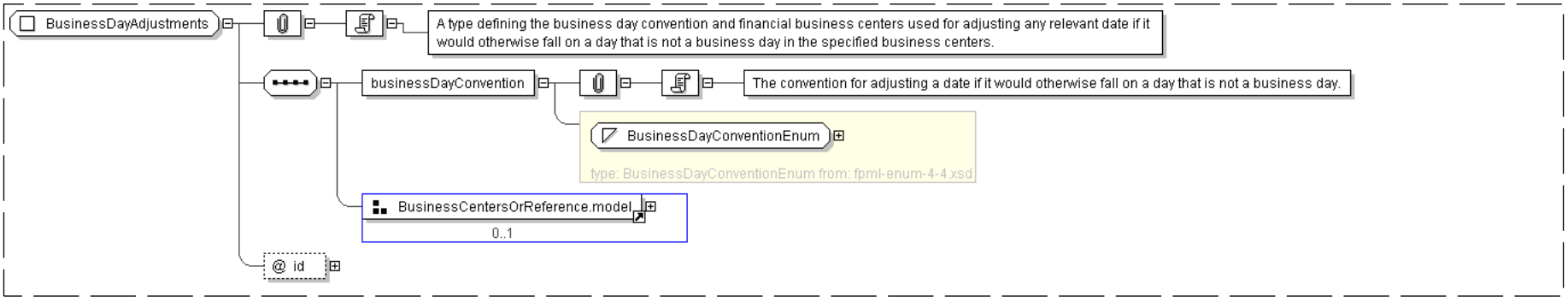
Name	BusinessDayAdjustments
Used by (from the same schema document)	Complex Type **AdjustableDate** , Complex Type **AdjustableDate2** , Complex Type **AdjustableDates** , Complex Type **AdjustedRelativeDateOffset** , Complex Type **PeriodicDates**
Abstract	no
Documentation	A type defining the business day convention and financial business centers used for adjusting any relevant date if it would otherwise fall on a day that is not a business day in the specified business centers.

XML Instance Representation

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

```
    'The convention for adjusting a date if it would otherwise fall on a day that is not a
    business day.'

Start Group: BusinessCentersOrReference.model [0..1]
Start Choice [1]
          <businessCentersReference> BusinessCentersReference </businessCentersReference> [1]
          'A pointer style reference to a set of financial business centers defined elsewhere in
          the document. This set of business centers is used to determine whether a particular day is
          a business day or not.'

          <businessCenters> BusinessCenters </businessCenters> [1]
End Choice
End Group: BusinessCentersOrReference.model
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="BusinessDayAdjustments">
   <xsd:sequence>
      <xsd:element name="businessDayConvention" type=" BusinessDayConventionEnum "/>
      <xsd:group ref=" BusinessCentersOrReference.model " minOccurs="0"/>
   </xsd:sequence>
   <xsd:attribute name="id" type=" xsd:ID "/>
</xsd:complexType>
```

top

Complex Type: BusinessDayAdjustmentsReference

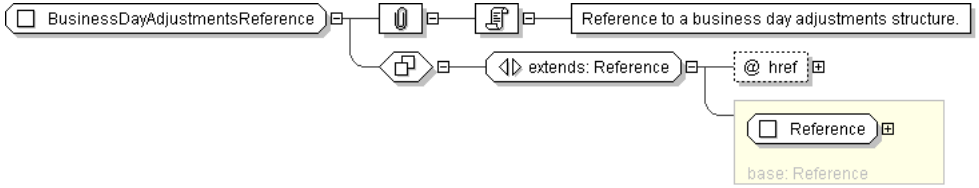
Super-types:	Reference < **BusinessDayAdjustmentsReference** (by extension)
Sub-types:	None

Name	BusinessDayAdjustmentsReference
Used by (from the same schema document)	Complex Type **AdjustableDate2**
Abstract	no
Documentation	Reference to a business day adjustments structure.

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: CalculationAgent

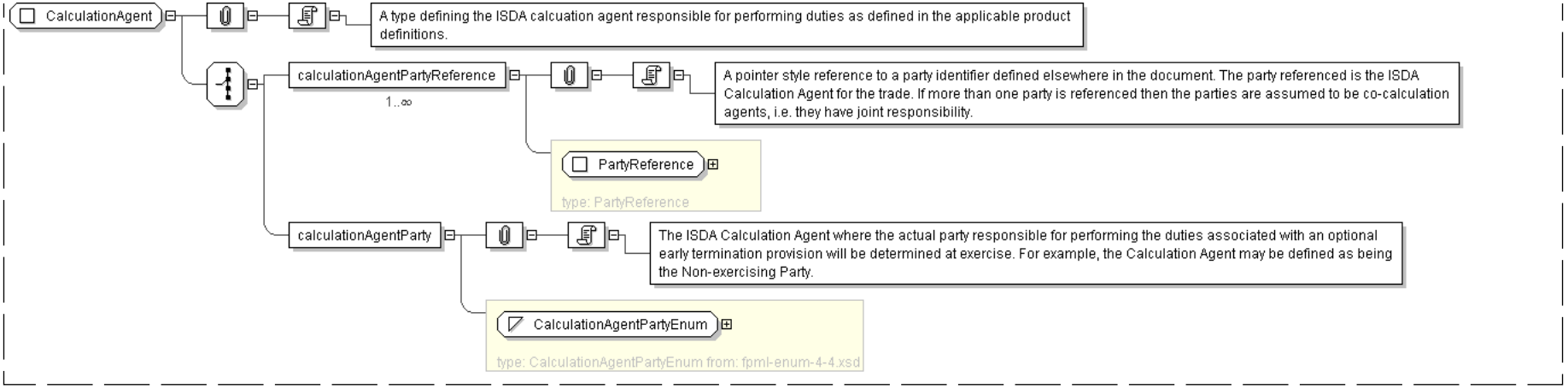
Super-types:	None
Sub-types:	None

Name	CalculationAgent
Abstract	no
Documentation	A type defining the ISDA calcuation agent responsible for performing duties as defined in the applicable product definitions.

XML Instance Representation

```
<...>
Start Choice [1]
      <calculationAgentPartyReference> PartyReference </calculationAgentPartyReference> [1..*]
      'A pointer style reference to a party identifier defined elsewhere in the document. The
      party referenced is the ISDA Calculation Agent for the trade. If more than one party
      is referenced then the parties are assumed to be co-calculation agents, i.e. they have
      joint responsibility.'

      <calculationAgentParty> CalculationAgentPartyEnum </calculationAgentParty> [1]
      'The ISDA Calculation Agent where the actual party responsible for performing the
      duties associated with an optional early termination provision will be determined at
      exercise. For example, the Calculation Agent may be defined as being the Non-exercising Party.'

End Choice
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="CalculationAgent">
   <xsd:choice>
      <xsd:element name="calculationAgentPartyReference" type=" PartyReference
      " maxOccurs="unbounded"/>
      <xsd:element name="calculationAgentParty" type=" CalculationAgentPartyEnum "/>
   </xsd:choice>
</xsd:complexType>
```

top

Complex Type: CalculationPeriodFrequency

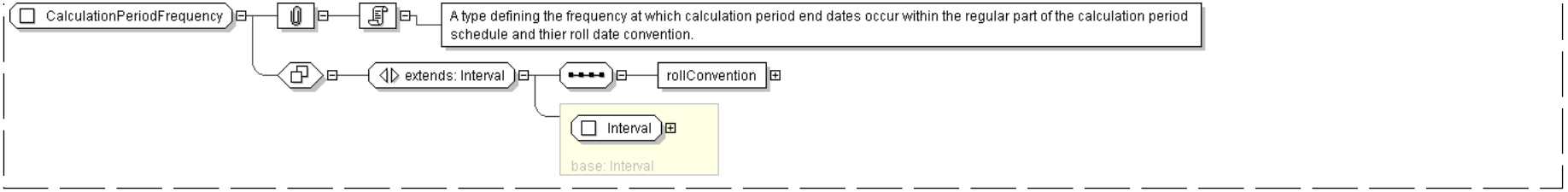
Super-types:	Interval < **CalculationPeriodFrequency** (by extension)
Sub-types:	None

Name	CalculationPeriodFrequency
Used by (from the same schema document)	Complex Type **PeriodicDates**
Abstract	no
Documentation	A type defining the frequency at which calculation period end dates occur within the regular part of the calculation period schedule and thier roll date convention.

XML Instance Representation

```
<...
 id=" xsd:ID [0..1]">
    <periodMultiplier> xsd:integer </periodMultiplier> [1]
    'A time period multiplier, e.g. 1, 2 or 3 etc. A negative value can be used when specifying
    an offset relative to another date, e.g. -2 days. If the period value is T (Term)
    then periodMultiplier must contain the value 1.'

    <period> PeriodEnum </period> [1]
    'A time period, e.g. a day, week, month, year or term of the stream. If the
    periodMultiplier value is 0 (zero) then period must contain the value D (day).'

    <rollConvention> RollConventionEnum </rollConvention> [1]
    'Used in conjunction with a frequency and the regular period start date of a
    calculation period, determines each calculation period end date within the regular part of
    a calculation period schedule.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="CalculationPeriodFrequency">
   <xsd:complexContent>
      <xsd:extension base=" Interval ">
         <xsd:sequence>
            <xsd:element name="rollConvention" type=" RollConventionEnum "/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: CashSettlementReferenceBanks

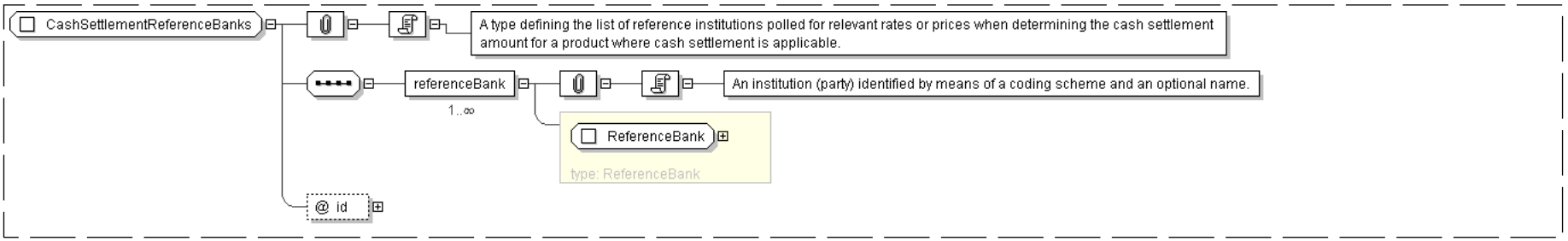
Super-types:	None
Sub-types:	None

Name	CashSettlementReferenceBanks
Used by (from the same schema document)	Complex Type **SettlementRateSource**
Abstract	no
Documentation	A type defining the list of reference institutions polled for relevant rates or prices when determining the cash settlement amount for a product where cash settlement is applicable.

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
   <referenceBank> ReferenceBank </referenceBank> [1..*]
   'An institution (party) identified by means of a coding scheme and an optional name.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="CashSettlementReferenceBanks">
   <xsd:sequence>
      <xsd:element name="referenceBank" type=" ReferenceBank " maxOccurs="unbounded"/>
   </xsd:sequence>
   <xsd:attribute name="id" type=" xsd:ID "/>
</xsd:complexType>
```

top

Complex Type: CashflowType

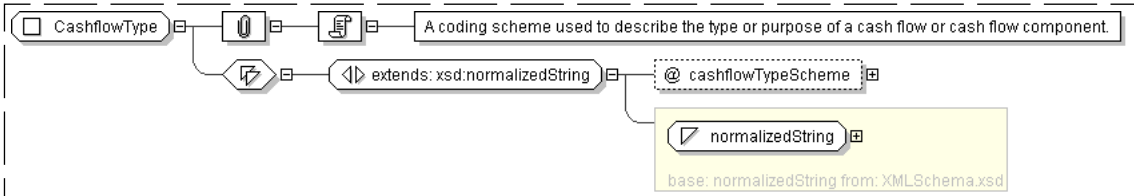
Super-types:	xsd:normalizedString < **CashflowType** (by extension)
Sub-types:	None

Name	CashflowType
Abstract	no
Documentation	A coding scheme used to describe the type or purpose of a cash flow or cash flow component.

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="CashflowType">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="cashflowTypeScheme" type=" xsd:anyURI " default="http://www.fpml.
         org/coding-scheme/cashflow-type-2-0"/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: ClearanceSystem

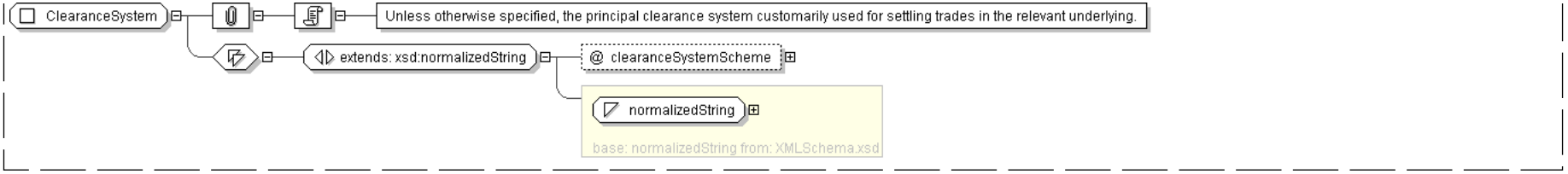
Super-types:	xsd:normalizedString < **ClearanceSystem** (by extension)
Sub-types:	None

Name	ClearanceSystem
Abstract	no
Documentation	Unless otherwise specified, the principal clearance system customarily used for settling trades in the relevant underlying.

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: ContractualDefinitions

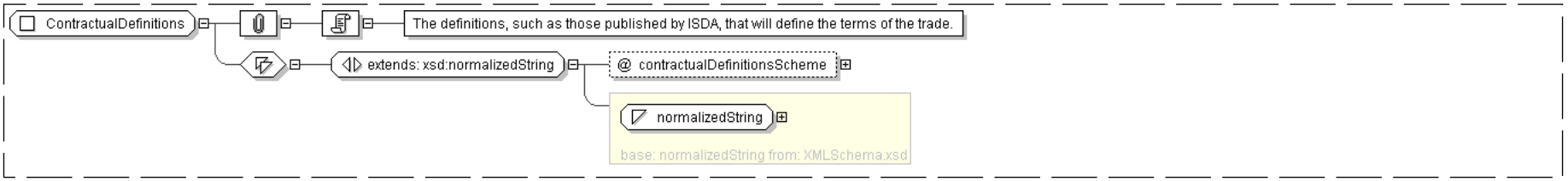
Super-types:	xsd:normalizedString < **ContractualDefinitions** (by extension)
Sub-types:	None

Name	ContractualDefinitions
Used by (from the same schema document)	Complex Type **Documentation**
Abstract	no
Documentation	The definitions, such as those published by ISDA, that will define the terms of the trade.

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="ContractualDefinitions">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="contractualDefinitionsScheme" type=" xsd:anyURI " default="http://www.
         fpml.org/coding-scheme/contractual-definitions-3-2"/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: ContractualMatrix

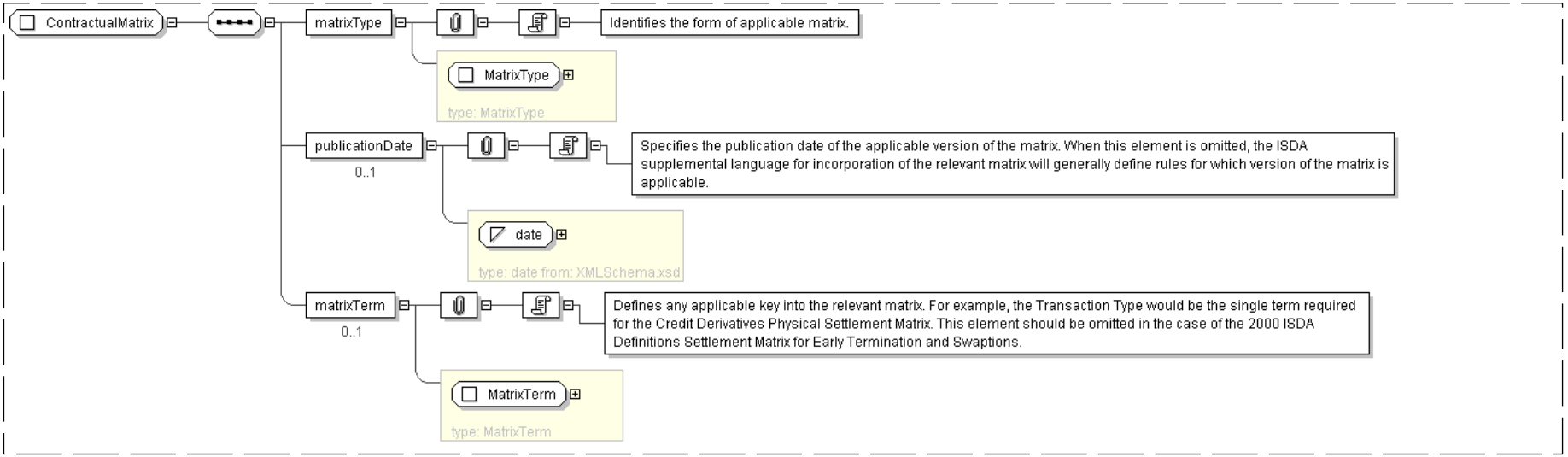
Super-types:	None
Sub-types:	None

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

XML Instance Representation

```
<...>
   <matrixType> MatrixType </matrixType> [1]
   'Identifies the form of applicable matrix.'

   <publicationDate> xsd:date </publicationDate> [0..1]
   'Specifies the publication date of the applicable version of the matrix. When this element
   is omitted, the ISDA supplemental language for incorporation of the relevant matrix
   will generally define rules for which version of the matrix is applicable.'

   <matrixTerm> MatrixTerm </matrixTerm> [0..1]
   'Defines any applicable key into the relevant matrix. For example, the Transaction Type
   would be the single term required for the Credit Derivatives Physical Settlement Matrix.
   This element should be omitted in the case of the 2000 ISDA Definitions Settlement Matrix
   for Early Termination and Swaptions.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="ContractualMatrix">
   <xsd:sequence>
      <xsd:element name="matrixType" type=" MatrixType "/>
      <xsd:element name="publicationDate" type=" xsd:date " minOccurs="0"/>
      <xsd:element name="matrixTerm" type=" MatrixTerm " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: ContractualSupplement

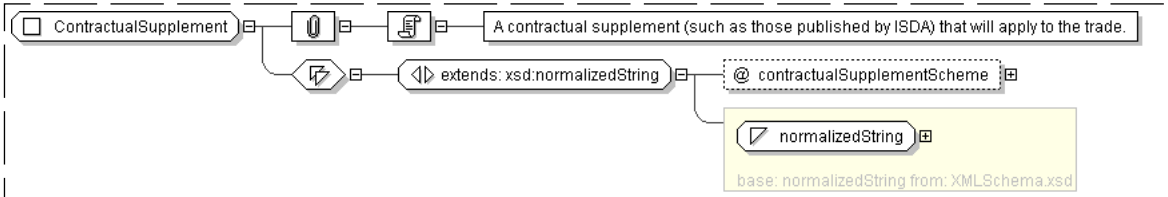
Super-types:	xsd:normalizedString < **ContractualSupplement** (by extension)
Sub-types:	None

Name	ContractualSupplement
Used by (from the same schema document)	Complex Type **ContractualTermsSupplement** , Complex Type **Documentation**
Abstract	no
Documentation	A contractual supplement (such as those published by ISDA) that will apply to the trade.

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="ContractualSupplement">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="contractualSupplementScheme" type=" xsd:anyURI " default="http://www.
         fpml.org/coding-scheme/contractual-supplement-6-3"/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: ContractualTermsSupplement

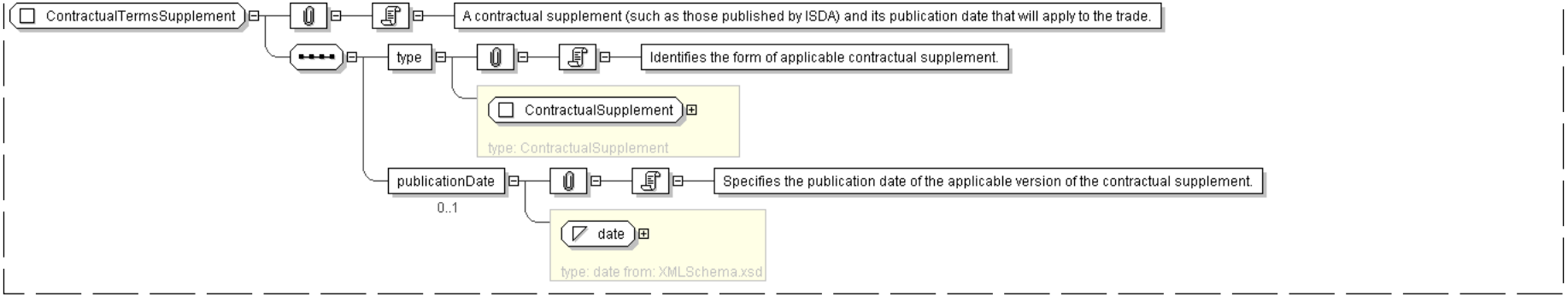
Super-types:	None
Sub-types:	None

Name	ContractualTermsSupplement
Used by (from the same schema document)	Complex Type **Documentation**
Abstract	no
Documentation	A contractual supplement (such as those published by ISDA) and its publication date that will apply to the trade.

XML Instance Representation

```
<...>
   <type> ContractualSupplement </type> [1]
   'Identifies the form of applicable contractual supplement.'

   <publicationDate> xsd:date </publicationDate> [0..1]
   'Specifies the publication date of the applicable version of the contractual supplement.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="ContractualTermsSupplement">
   <xsd:sequence>
      <xsd:element name="type" type=" ContractualSupplement "/>
      <xsd:element name="publicationDate" type=" xsd:date " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: CorrespondentInformation

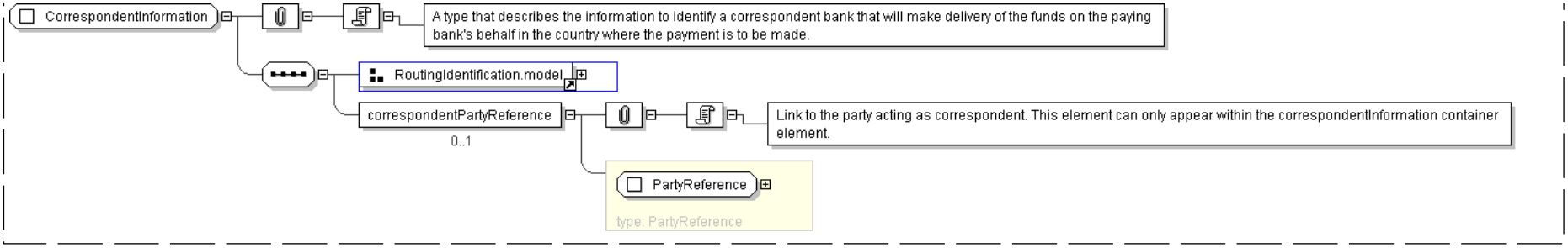
Super-types:	None
Sub-types:	None

Name	CorrespondentInformation
Used by (from the same schema document)	Complex Type **SettlementInstruction**
Abstract	no
Documentation	A type that describes the information to identify a correspondent bank that will make delivery of the funds on the paying bank's behalf in the country where the payment is to be made.

XML Instance Representation

```
<...>
Start Choice [1]
      <routingIds> RoutingIds </routingIds> [1]
      'A set of unique identifiers for a party, eachone identifying the party within a
      payment system. The assumption is that each party will not have more than one identifier
      within the same payment system.'

      <routingExplicitDetails> RoutingExplicitDetails </routingExplicitDetails> [1]
      'A set of details that is used to identify a party involved in the routing of a payment
      when the party does not have a code that identifies it within one of the recognized
      payment systems.'

      <routingIdsAndExplicitDetails> RoutingIdsAndExplicitDetails </routingIdsAndExplicitDetails> [1]
      'A combination of coded payment system identifiers and details for physical addressing for
      a party involved in the routing of a payment.'

End Choice
   <correspondentPartyReference> PartyReference </correspondentPartyReference> [0..1]
   'Link to the party acting as correspondent. This element can only appear within
   the correspondentInformation container element.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="CorrespondentInformation">
   <xsd:sequence>
      <xsd:group ref=" RoutingIdentification.model "/>
      <xsd:element name="correspondentPartyReference" type=" PartyReference " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: Country

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

Name	Country
Used by (from the same schema document)	Complex Type **Address**
Abstract	no

XML Instance Representation

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

Diagram

Schema Component Representation

```
<xsd:complexType name="Country">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="countryScheme" type=" xsd:anyURI " default="http://www.fpml.org/
         ext/iso3166"/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: CreditSeniority

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

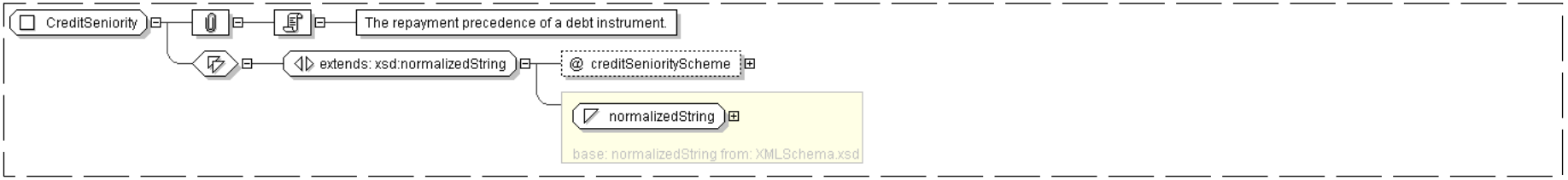
Sub-types:	None

Name	CreditSeniority
Abstract	no
Documentation	The repayment precedence of a debt instrument.

XML Instance Representation

```
<...
creditSeniorityScheme=" xsd:anyURI [0..1]
'creditSeniorityTradingScheme overrides creditSeniorityScheme when the underlyer defines
the reference obligation used in a single name credit default swap trade.'

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

Diagram



Schema Component Representation

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

top

Complex Type: Currency

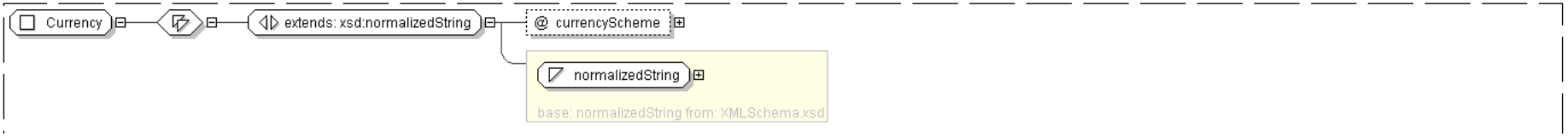
Super-types:	xsd:normalizedString < **Currency** (by extension)
Sub-types:	None

Name	Currency
Used by (from the same schema document)	Complex Type **AmountSchedule** , Complex Type **DividendConditions** , Complex Type **FxCashSettlement** , Complex Type **Money** , Complex Type **PaymentCurrency** , Complex Type **PricingStructure** , Complex Type **QuotedCurrencyPair** , Complex Type **QuotedCurrencyPair** , Model Group **SettlementAmountOrCurrency.model**
Abstract	no

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="Currency">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="currencyScheme" type=" xsd:anyURI " default="http://www.fpml.org/
         ext/iso4217-2001-08-15"/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: DateList

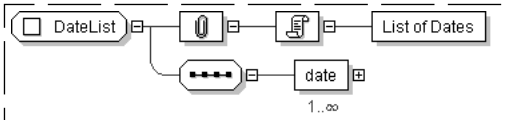
Super-types:	None
Sub-types:	None

Name	DateList
Abstract	no
Documentation	List of Dates

XML Instance Representation

```
<...>
   <date> xsd:date </date> [1..*]
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="DateList">
   <xsd:sequence>
      <xsd:element name="date" type=" xsd:date " maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: DateOffset

Super-types:	Interval < Offset (by extension) < **DateOffset** (by extension)
Sub-types:	None

Name	DateOffset
Used by (from the same schema document)	Complex Type **RelativeDateSequence**
Abstract	no
Documentation	A type defining an offset used in calculating a date when this date is defined in reference to another date through a date offset. The type includes the convention for adjusting the date and an optional sequence element to indicate the order in a sequence of multiple date offsets.

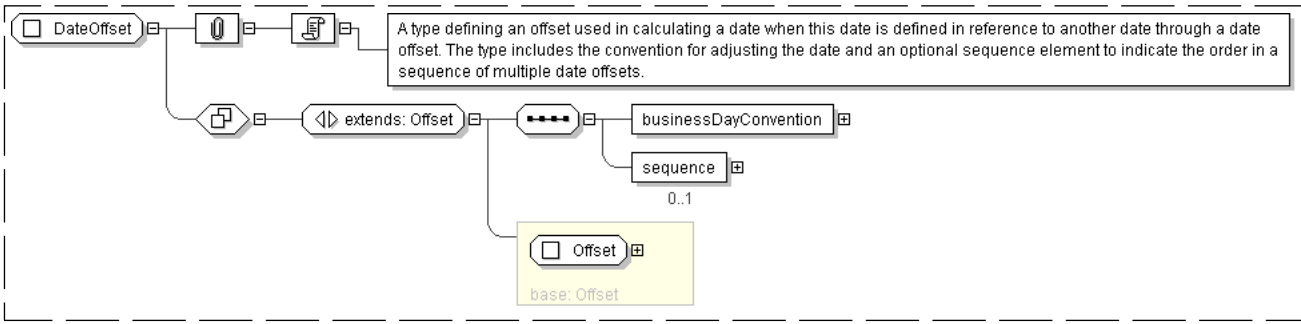
XML Instance Representation

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

'A time period multiplier, e.g. 1, 2 or 3 etc. A negative value can be used when specifying an offset relative to another date, e.g. -2 days. If the period value is T (Term) then periodMultiplier must contain the value 1.'

```
<period> PeriodEnum </period> [1]
'A time period, e.g. a day, week, month, year or term of the stream. If the
periodMultiplier value is 0 (zero) then period must contain the value D (day).'

<dayType> DayTypeEnum </dayType> [0..1]
'In the case of an offset specified as a number of days, this element defines
whether consideration is given as to whether a day is a good business day or not. If a day
type of business days is specified then non-business days are ignored when calculating
the offset. The financial business centers to use for determination of business days
are implied by the context in which this element is used. This element must only be
included when the offset is specified as a number of days. If the offset is zero days then
the dayType element should not be included.'

<businessDayConvention> BusinessDayConventionEnum </businessDayConvention> [1]
'The convention for adjusting a date if it would otherwise fall on a day that is not a
business day.'

<sequence> xsd:positiveInteger </sequence> [0..1]
'Sequence in which the reference to the time period multiplier should be applied.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="DateOffset">
   <xsd:complexContent>
      <xsd:extension base=" Offset ">
         <xsd:sequence>
            <xsd:element name="businessDayConvention" type=" BusinessDayConventionEnum "/>
            <xsd:element name="sequence" type=" xsd:positiveInteger " minOccurs="0"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: DateRange

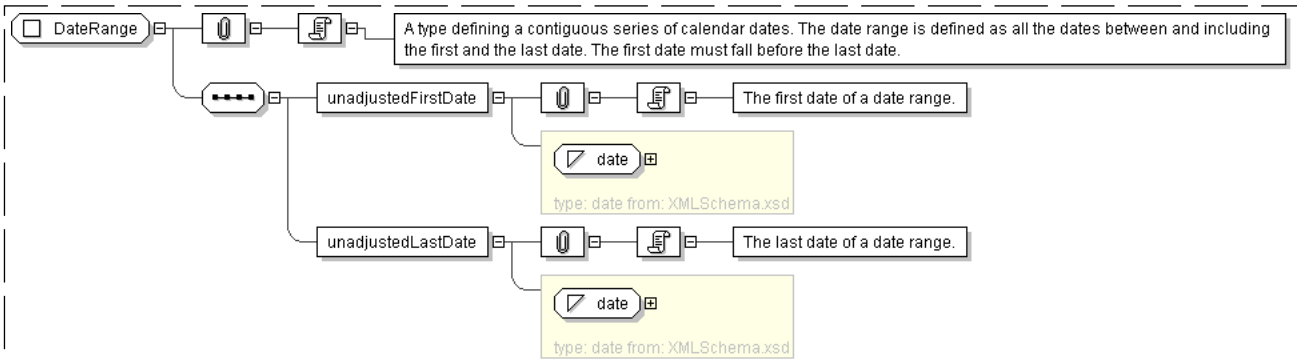
Super-types:	None
Sub-types:	• BusinessDateRange (by extension)

Name	DateRange
Used by (from the same schema document)	Complex Type **RelativeDates**
Abstract	no
Documentation	A type defining a contiguous series of calendar dates. The date range is defined as all the dates between and including the first and the last date. The first date must fall before the last date.

XML Instance Representation

```
<...>
   <unadjustedFirstDate> xsd:date </unadjustedFirstDate> [1]
   'The first date of a date range.'

   <unadjustedLastDate> xsd:date </unadjustedLastDate> [1]
   'The last date of a date range.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="DateRange">
   <xsd:sequence>
      <xsd:element name="unadjustedFirstDate" type=" xsd:date "/>
      <xsd:element name="unadjustedLastDate" type=" xsd:date "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: DateReference

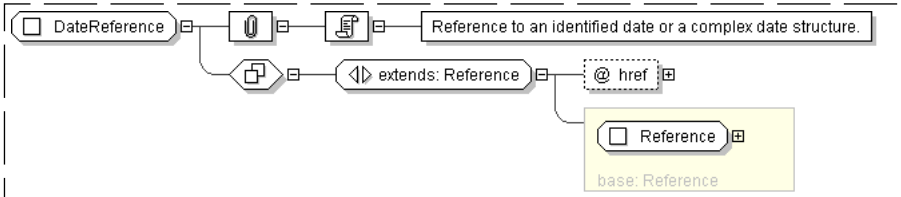
Super-types:	Reference < **DateReference** (by extension)
Sub-types:	None

Name	DateReference
Used by (from the same schema document)	Complex Type **DividendConditions** , Complex Type **DividendConditions** , Complex Type **RelativeDateOffset** , Complex Type **RelativeDateSequence**
Abstract	no
Documentation	Reference to an identified date or a complex date structure.

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: DateTimeList

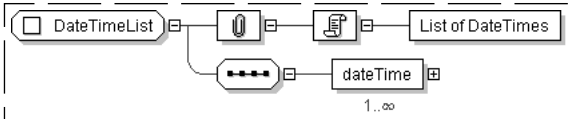
Super-types:	None
Sub-types:	None

Name	DateTimeList
Abstract	no
Documentation	List of DateTimes

XML Instance Representation

```
<...>
   <dateTime> xsd:dateTime </dateTime> [1..*]
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="DateTimeList">
   <xsd:sequence>
      <xsd:element name="dateTime" type=" xsd:dateTime " maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: DayCountFraction

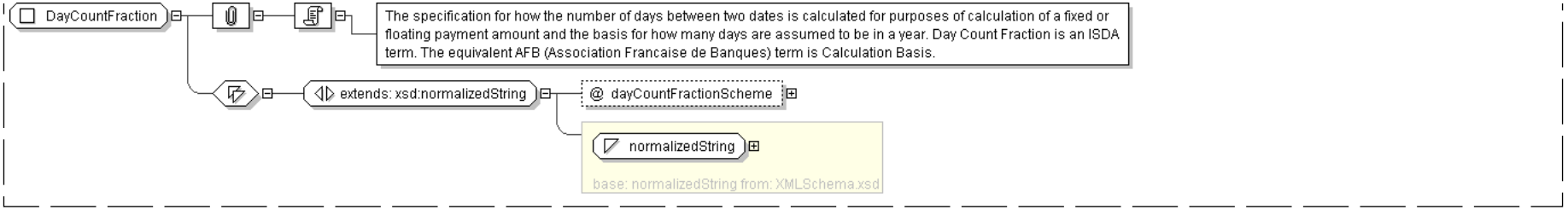
Super-types:	xsd:normalizedString < **DayCountFraction** (by extension)
Sub-types:	None

Name	DayCountFraction
Abstract	no
Documentation	The specification for how the number of days between two dates is calculated for purposes of calculation of a fixed or floating payment amount and the basis for how many days are assumed to be in a year. Day Count Fraction is an ISDA term. The equivalent AFB (Association Francaise de Banques) term is Calculation Basis.

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="DayCountFraction">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="dayCountFractionScheme" type=" xsd:anyURI " default="http://www.fpml.
         org/coding-scheme/day-count-fraction-2-1"/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: DeterminationMethod

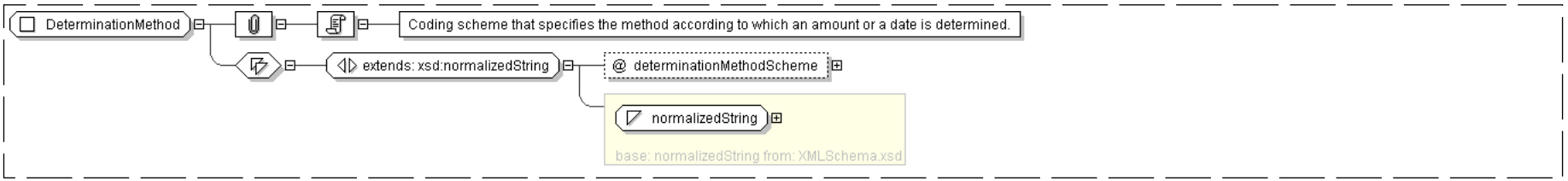
Super-types:	xsd:normalizedString < **DeterminationMethod** (by extension)
Sub-types:	None

Name	DeterminationMethod
Used by (from the same schema document)	Complex Type **DividendConditions** , Complex Type **PaymentCurrency**
Abstract	no
Documentation	Coding scheme that specifies the method according to which an amount or a date is determined.

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: DividendConditions

Super-types:	None
Sub-types:	None

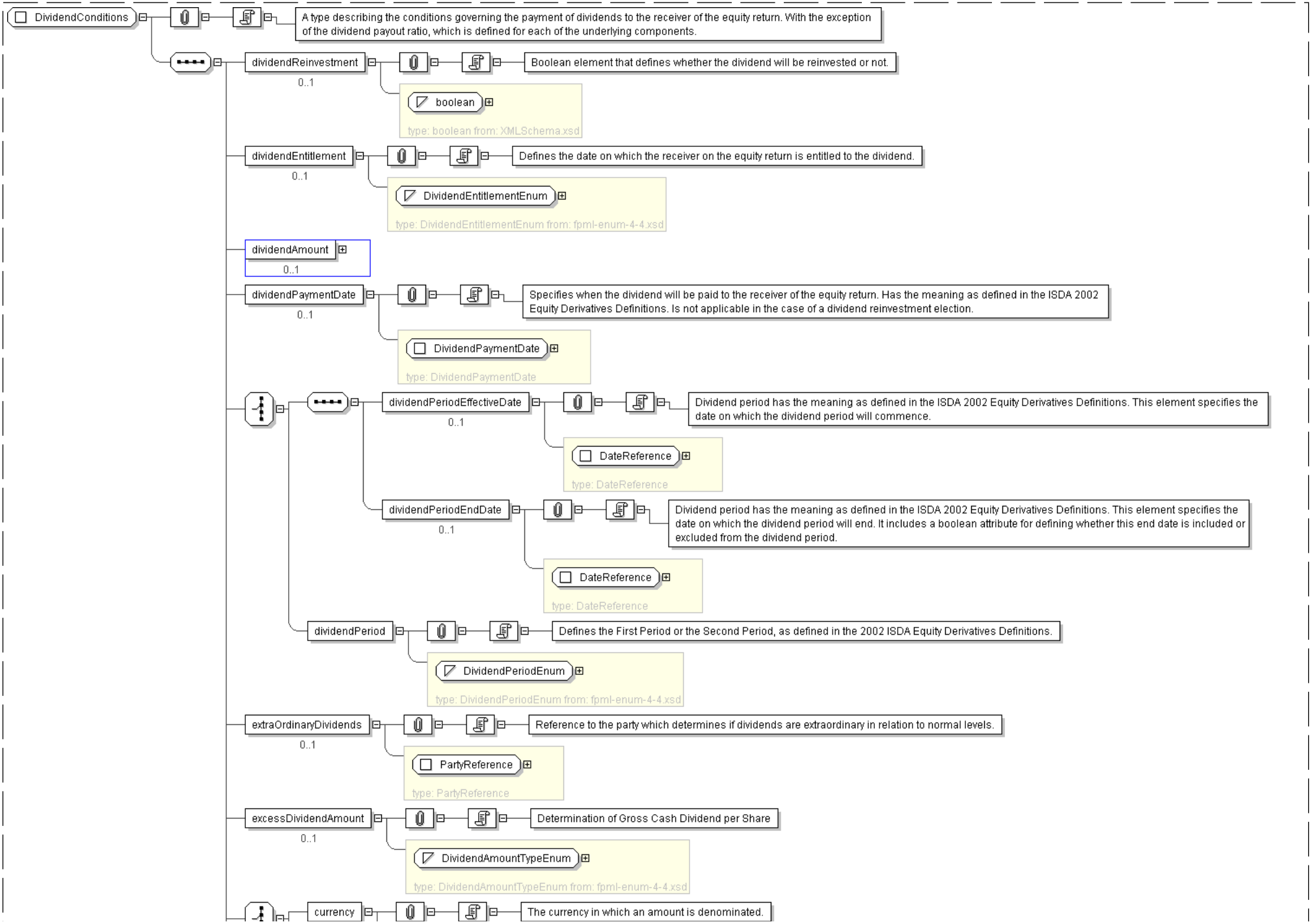
Name	DividendConditions
Abstract	no
Documentation	A type describing the conditions governing the payment of dividends to the receiver of the equity return. With the exception of the dividend payout ratio, which is defined for each of the underlying components.

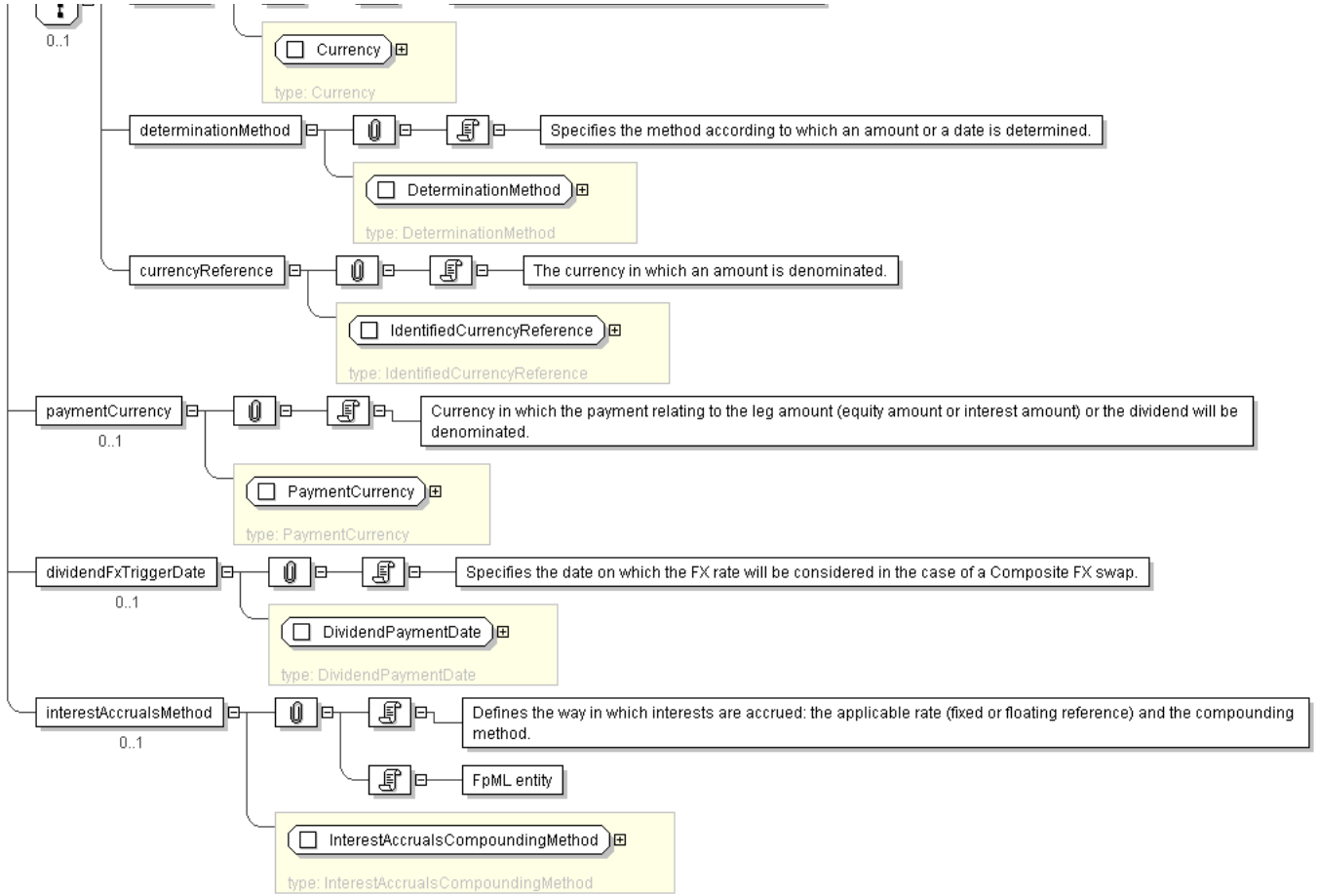
XML Instance Representation

```
<...>
   <dividendReinvestment> xsd:boolean </dividendReinvestment> [0..1]
   'Boolean element that defines whether the dividend will be reinvested or not.'

   <dividendEntitlement> DividendEntitlementEnum </dividendEntitlement> [0..1]
   'Defines the date on which the receiver on the equity return is entitled to the dividend.'

   <dividendAmount> DividendAmountTypeEnum </dividendAmount> [0..1]
   <dividendPaymentDate> DividendPaymentDate </dividendPaymentDate> [0..1]
   'Specifies when the dividend will be paid to the receiver of the equity return. Has the
   meaning as defined in the ISDA 2002 Equity Derivatives Definitions. Is not applicable in
   the case of a dividend reinvestment election.'

Start Choice [1]
      <dividendPeriodEffectiveDate> DateReference </dividendPeriodEffectiveDate> [0..1]
      'Dividend period has the meaning as defined in the ISDA 2002 Equity Derivatives
      Definitions. This element specifies the date on which the dividend period will commence.'

      <dividendPeriodEndDate> DateReference </dividendPeriodEndDate> [0..1]
      'Dividend period has the meaning as defined in the ISDA 2002 Equity Derivatives
      Definitions. This element specifies the date on which the dividend period will end. It
      includes a boolean attribute for defining whether this end date is included or excluded
      from the dividend period.'

      <dividendPeriod> DividendPeriodEnum </dividendPeriod> [1]
      'Defines the First Period or the Second Period, as defined in the 2002 ISDA Equity
      Derivatives Definitions.'

End Choice
   <extraOrdinaryDividends> PartyReference </extraOrdinaryDividends> [0..1]
   'Reference to the party which determines if dividends are extraordinary in relation to
   normal levels.'

   <excessDividendAmount> DividendAmountTypeEnum </excessDividendAmount> [0..1]
   'Determination of Gross Cash Dividend per Share'

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

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

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

End Choice
   <paymentCurrency> PaymentCurrency </paymentCurrency> [0..1]
   'Currency in which the payment relating to the leg amount (equity amount or interest amount)
   or the dividend will be denominated.'

   <dividendFxTriggerDate> DividendPaymentDate </dividendFxTriggerDate> [0..1]
   'Specifies the date on which the FX rate will be considered in the case of a Composite FX swap.'

   <interestAccrualsMethod> InterestAccrualsCompoundingMethod </interestAccrualsMethod> [0..1]
   'Defines the way in which interests are accrued: the applicable rate (fixed or
```

```
floating reference) and the compounding method.','FpML entity'

</...>
```

Diagram





Schema Component Representation

```
<xsd:complexType name="DividendConditions">
   <xsd:sequence>
      <xsd:element name="dividendReinvestment" type=" xsd:boolean " minOccurs="0"/>
      <xsd:element name="dividendEntitlement" type=" DividendEntitlementEnum " minOccurs="0"/>
      <xsd:element name="dividendAmount" type=" DividendAmountTypeEnum " minOccurs="0"/>
      <xsd:element name="dividendPaymentDate" type=" DividendPaymentDate " minOccurs="0"/>
      <xsd:choice>
         <xsd:sequence>
            <xsd:element name="dividendPeriodEffectiveDate" type=" DateReference " minOccurs="0"/>
            <xsd:element name="dividendPeriodEndDate" type=" DateReference " minOccurs="0"/>
         </xsd:sequence>
         <xsd:element name="dividendPeriod" type=" DividendPeriodEnum "/>
      </xsd:choice>
      <xsd:element name="extraOrdinaryDividends" type=" PartyReference " minOccurs="0"/>
      <xsd:element name="excessDividendAmount" type=" DividendAmountTypeEnum " minOccurs="0"/>
      <xsd:choice minOccurs="0">
         <xsd:element name="currency" type=" Currency "/>
         <xsd:element name="determinationMethod" type=" DeterminationMethod "/>
         <xsd:element name="currencyReference" type=" IdentifiedCurrencyReference "/>
      </xsd:choice>
      <xsd:element name="paymentCurrency" type=" PaymentCurrency " minOccurs="0"
      deprecated="true" deprecatedReason="The model is wrong since it has an intradocument
      reference that is not clear. Current PaymentCurrency model and elements using this type
      are deprecated. Instead, the choice above between currency, determinationMethod,
      and currencyReference (of type CurrencyReference) should be used."/>
```

```
      <xsd:element name="dividendFxTriggerDate" type=" DividendPaymentDate " minOccurs="0"/>
      <xsd:element name="interestAccrualsMethod" type=" InterestAccrualsCompoundingMethod
      " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: DividendPaymentDate

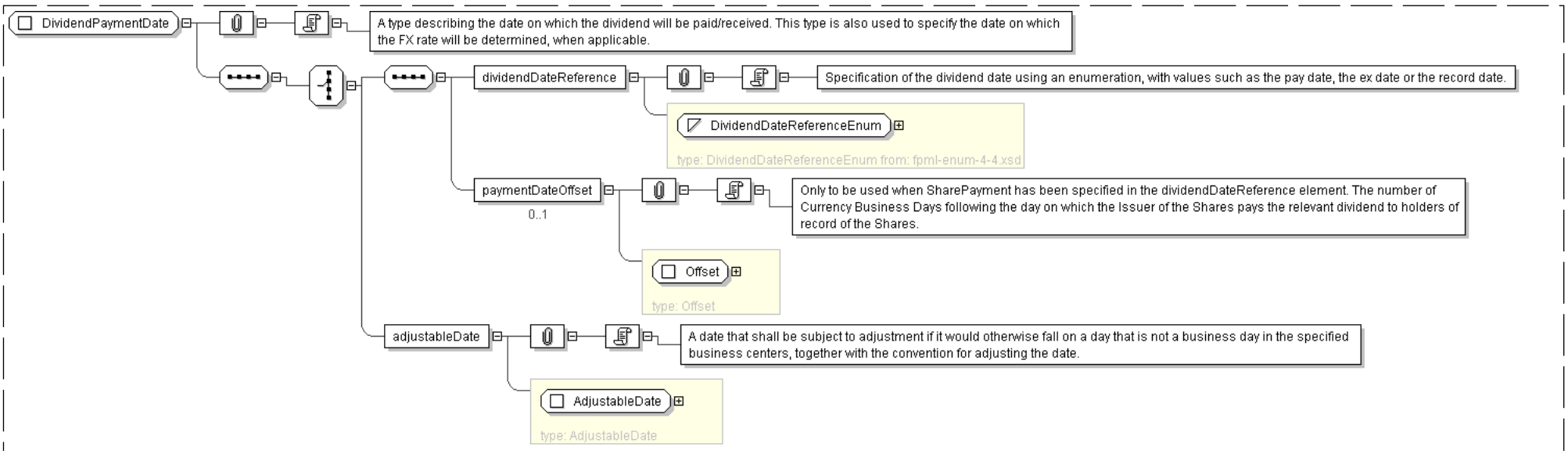
Super-types:	None
Sub-types:	None

Name	DividendPaymentDate
Used by (from the same schema document)	Complex Type **DividendConditions** , Complex Type **DividendConditions**
Abstract	no
Documentation	A type describing the date on which the dividend will be paid/received. This type is also used to specify the date on which the FX rate will be determined, when applicable.

XML Instance Representation

```
<...>
Start Choice [1]
      <dividendDateReference> DividendDateReferenceEnum </dividendDateReference> [1]
      'Specification of the dividend date using an enumeration, with values such as the pay date,
      the ex date or the record date.'

      <paymentDateOffset> Offset </paymentDateOffset> [0..1]
      'Only to be used when SharePayment has been specified in the dividendDateReference element.
      The number of Currency Business Days following the day on which the Issuer of the Shares
      pays the relevant dividend to holders of record of the Shares.'

      <adjustableDate> AdjustableDate </adjustableDate> [1]
      'A date that shall be subject to adjustment if it would otherwise fall on a day that is not
      a business day in the specified business centers, together with the convention for
      adjusting the date.'

End Choice
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="DividendPaymentDate">
   <xsd:sequence>
      <xsd:choice>
         <xsd:sequence>
            <xsd:element name="dividendDateReference" type=" DividendDateReferenceEnum "/>
            <xsd:element name="paymentDateOffset" type=" Offset " minOccurs="0"/>
         </xsd:sequence>
         <xsd:element name="adjustableDate" type=" AdjustableDate "/>
      </xsd:choice>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: Documentation

Super-types:	None
Sub-types:	None

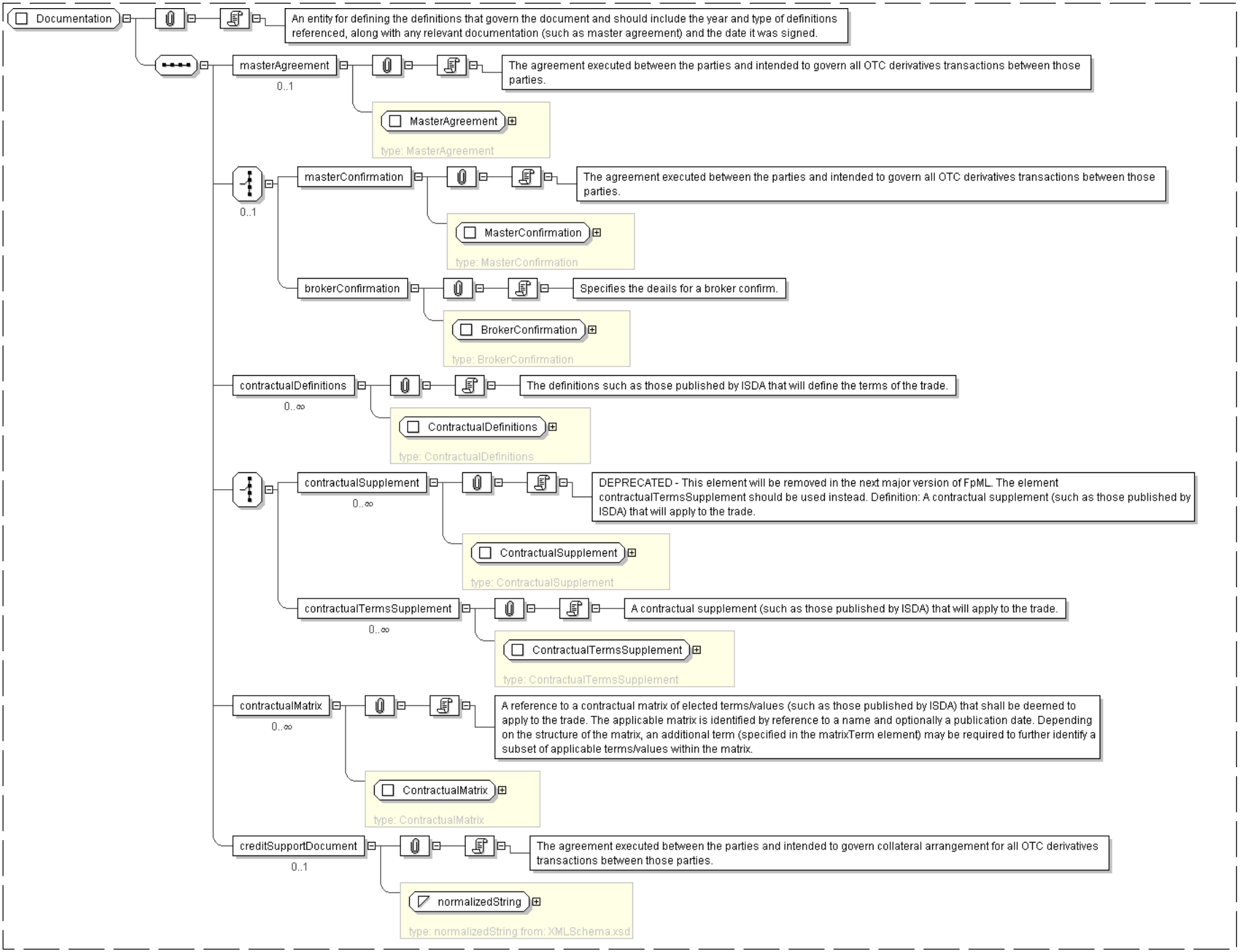
Name	Documentation
Abstract	no
Documentation	An entity for defining the definitions that govern the document and should include the year and type of definitions referenced, along with any relevant documentation (such as master agreement) and the date it was signed.

XML Instance Representation

```
<...>
   <masterAgreement> MasterAgreement </masterAgreement> [0..1]
   'The agreement executed between the parties and intended to govern all OTC
   derivatives transactions between those parties.'

Start Choice [0..1]
      <masterConfirmation> MasterConfirmation </masterConfirmation> [1]
      'The agreement executed between the parties and intended to govern all OTC
      derivatives transactions between those parties.'

      <brokerConfirmation> BrokerConfirmation </brokerConfirmation> [1]
      'Specifies the deails for a broker confirm.'

End Choice
   <contractualDefinitions> ContractualDefinitions </contractualDefinitions> [0..*]
   'The definitions such as those published by ISDA that will define the terms of the trade.'

Start Choice [1]
      <contractualSupplement> ContractualSupplement </contractualSupplement> [0..*]
      'DEPRECATED - This element will be removed in the next major version of FpML. The
      element contractualTermsSupplement should be used instead. Definition: A contractual
      supplement (such as those published by ISDA) that will apply to the trade.'

      <contractualTermsSupplement> ContractualTermsSupplement </contractualTermsSupplement> [0..*]
      'A contractual supplement (such as those published by ISDA) that will apply to the trade.'

End Choice
   <contractualMatrix> ContractualMatrix </contractualMatrix> [0..*]
   'A reference to a contractual matrix of elected terms/values (such as those published by
   ISDA) that shall be deemed to apply to the trade. The applicable matrix is identified
   by reference to a name and optionally a publication date. Depending on the structure of
   the matrix, an additional term (specified in the matrixTerm element) may be required to
   further identify a subset of applicable terms/values within the matrix.'

   <creditSupportDocument> xsd:normalizedString </creditSupportDocument> [0..1]
   'The agreement executed between the parties and intended to govern collateral arrangement
   for all OTC derivatives transactions between those parties.'
```

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

Diagram



Schema Component Representation

```
<xsd:complexType name="Documentation">
   <xsd:sequence>
      <xsd:element name="masterAgreement" type=" MasterAgreement " minOccurs="0"/>
      <xsd:choice minOccurs="0">
         <xsd:element name="masterConfirmation" type=" MasterConfirmation "/>
         <xsd:element name="brokerConfirmation" type=" BrokerConfirmation "/>
      </xsd:choice>
      <xsd:element name="contractualDefinitions" type=" ContractualDefinitions "
      minOccurs="0" maxOccurs="unbounded"/>
      <xsd:choice>
         <xsd:element name="contractualSupplement" type=" ContractualSupplement "
         minOccurs="0" maxOccurs="unbounded" deprecated="true"
         deprecatedReason="The contractualTermsSupplement includes the publication date, which was
         not present in the contractualSupplement"/>
         <xsd:element name="contractualTermsSupplement" type=" ContractualTermsSupplement
         " minOccurs="0" maxOccurs="unbounded"/>
      </xsd:choice>
      <xsd:element name="contractualMatrix" type=" ContractualMatrix "
      minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="creditSupportDocument" type=" xsd:normalizedString " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: Empty

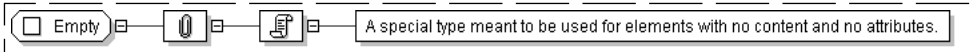
Super-types:	None
Sub-types:	None

Name	Empty
Abstract	no
Documentation	A special type meant to be used for elements with no content and no attributes.

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="Empty"/>
```

top

Complex Type: EntityId

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

Name	EntityId
Used by (from the same schema document)	Complex Type **LegalEntity** , Complex Type **LegalEntity**
Abstract	no

XML Instance Representation

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

Diagram

Schema Component Representation

```
<xsd:complexType name="EntityId">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="entityIdScheme" type=" xsd:anyURI " default="http://www.fpml.org/
         spec/2003/entity-id-RED-1-0"/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: EntityName

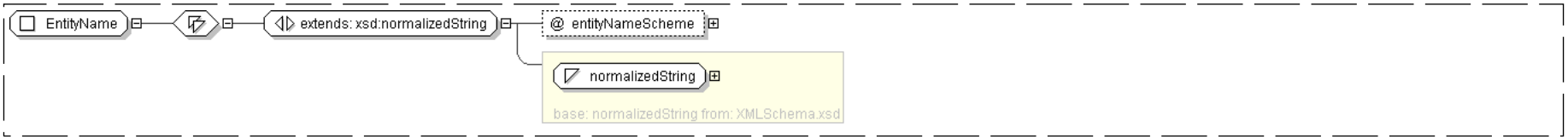
Super-types:	xsd:normalizedString < **EntityName** (by extension)
Sub-types:	None

Name	EntityName
Used by (from the same schema document)	Complex Type **LegalEntity**
Abstract	no

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="EntityName">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="entityNameScheme" type=" xsd:anyURI " default="http://www.fpml.
         org/spec/2003/entity-name-RED-1-0"/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: EuropeanExercise

Super-types:	Exercise < **EuropeanExercise** (by extension)
Sub-types:	None

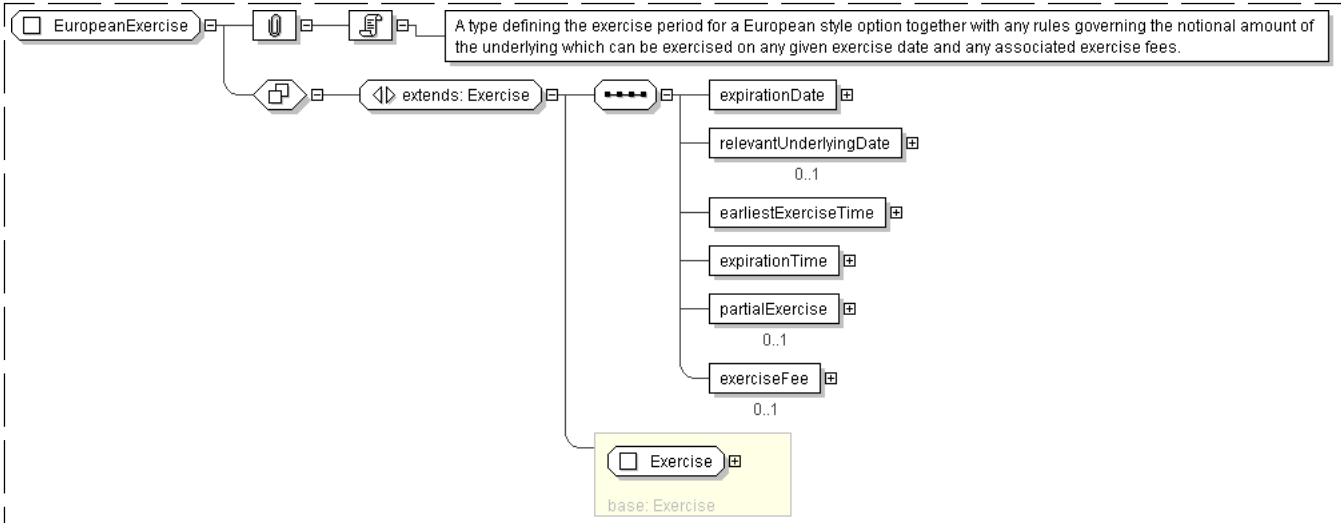
Name	EuropeanExercise
Used by (from the same schema document)	Element **europeanExercise**

Abstract	no
Documentation	A type defining the exercise period for a European style option together with any rules governing the notional amount of the underlying which can be exercised on any given exercise date and any associated exercise fees.

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
   <expirationDate> AdjustableOrRelativeDate </expirationDate> [1]
   'The last day within an exercise period for an American style option. For a European
   style option it is the only day within the exercise period.'

   <relevantUnderlyingDate> AdjustableOrRelativeDates </relevantUnderlyingDate> [0..1]
   'The daye on the underlying set by the exercise of an option. What this date is depends on
   the option (e.g. in a swaption it is the effective date, in an extendible/cancelable
   provision it is the termination date).'

   <earliestExerciseTime> BusinessCenterTime </earliestExerciseTime> [1]
   'The earliest time at which notice of exercise can be given by the buyer to the seller
   (or seller\'s agent) i) on the expriation date, in the case of a European style option, (ii)
   on each bermuda option exercise date and the expiration date, in the case of a Bermuda
   style option the commencement date to, and including, the expiration date , in the case of
   an American option.'

   <expirationTime> BusinessCenterTime </expirationTime> [1]
   'The latest time for exercise on expirationDate.'

   <partialExercise> PartialExercise </partialExercise> [0..1]
   'As defined in the 2000 ISDA Definitions, Section 12.3. Partial Exercise, the buyer of
   the option has the right to exercise all or less than all the notional amount of the
   underlying swap on the expiration date, but may not exercise less than the minimum
   notional amount, and if an integral multiple amount is specified, the notional amount
   exercised must be equal to, or be an integral multiple of, the integral multiple amount.'

   <exerciseFee> ExerciseFee </exerciseFee> [0..1]
   'A fee to be paid on exercise. This could be represented as an amount or a rate and
   notional reference on which to apply the rate.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="EuropeanExercise">
```

```
   <xsd:complexContent>
      <xsd:extension base=" Exercise ">
         <xsd:sequence>
            <xsd:element name="expirationDate" type=" AdjustableOrRelativeDate "/>
            <xsd:element name="relevantUnderlyingDate" type=" AdjustableOrRelativeDates " minOccurs="0"/>
            <xsd:element name="earliestExerciseTime" type=" BusinessCenterTime "/>
            <xsd:element name="expirationTime" type=" BusinessCenterTime "/>
            <xsd:element name="partialExercise" type=" PartialExercise " minOccurs="0"/>
            <xsd:element name="exerciseFee" type=" ExerciseFee " minOccurs="0"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: ExchangeId

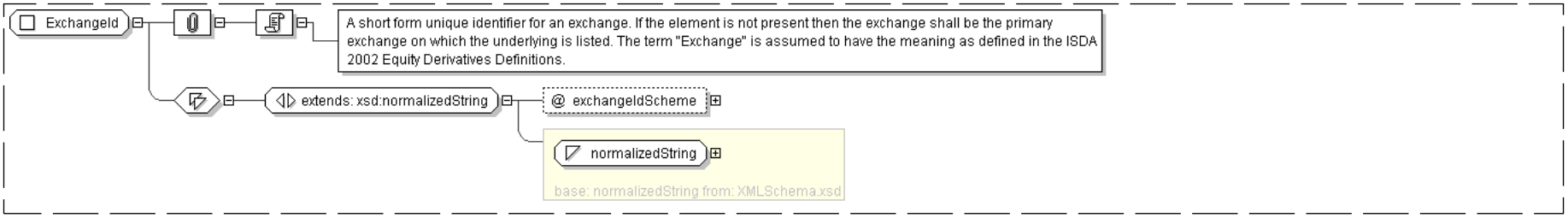
Super-types:	xsd:normalizedString < **ExchangeId** (by extension)
Sub-types:	None

Name	ExchangeId
Abstract	no
Documentation	A short form unique identifier for an exchange. If the element is not present then the exchange shall be the primary exchange on which the underlying is listed. The term "Exchange" is assumed to have the meaning as defined in the ISDA 2002 Equity Derivatives Definitions.

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="ExchangeId">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="exchangeIdScheme" type=" xsd:anyURI " default="http://www.fpml.
         org/spec/2002/exchange-id-MIC-1-0"/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: Exercise

Super-types:	None

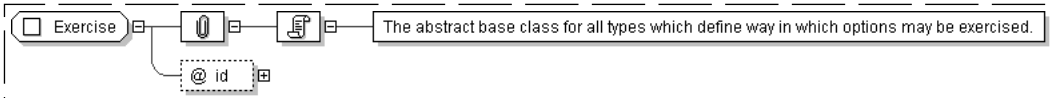
Sub-types:	• AmericanExercise (by extension) • BermudaExercise (by extension) • EuropeanExercise (by extension) • SharedAmericanExercise (by extension)

Name	Exercise
Used by (from the same schema document)	Element **exercise**
Abstract	no
Documentation	The abstract base class for all types which define way in which options may be exercised.

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="Exercise">
   <xsd:attribute name="id" type=" xsd:ID "/>
</xsd:complexType>
```

top

Complex Type: ExerciseFee

Super-types:	None
Sub-types:	None

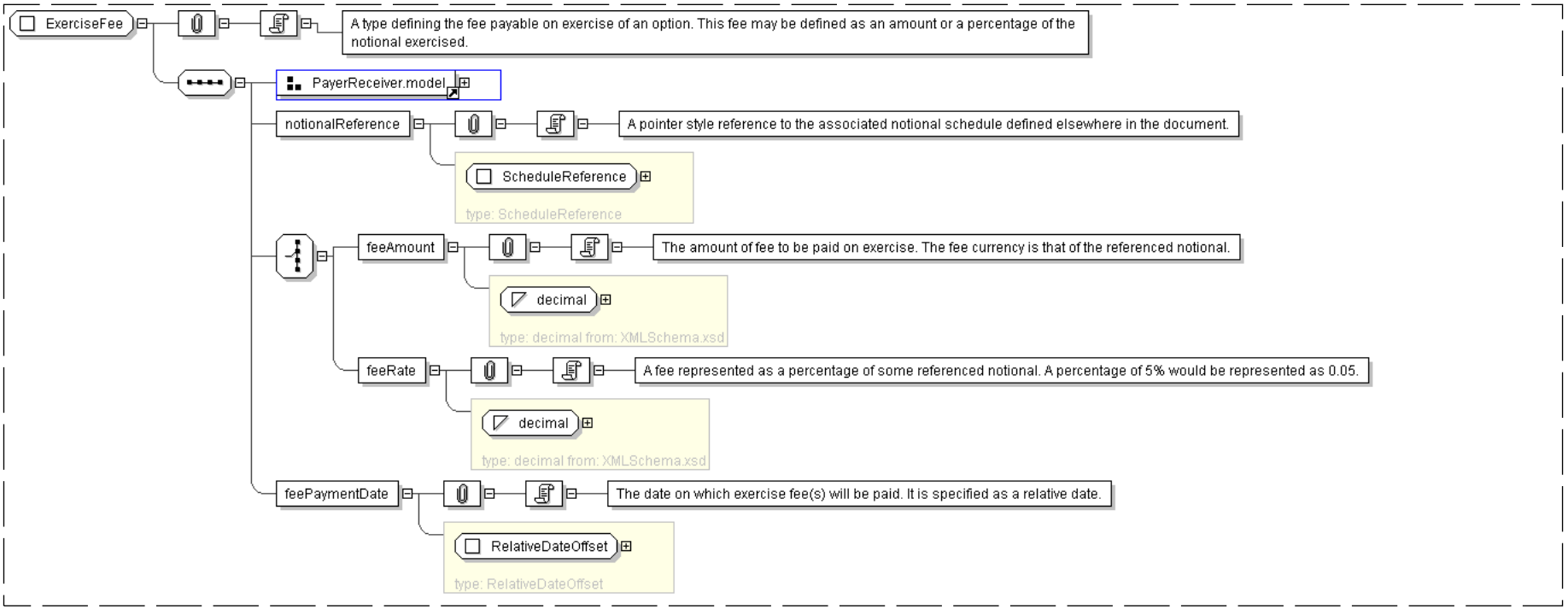
Name	ExerciseFee
Used by (from the same schema document)	Complex Type **EuropeanExercise**
Abstract	no
Documentation	A type defining the fee payable on exercise of an option. This fee may be defined as an amount or a percentage of the notional exercised.

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.'

   <notionalReference> ScheduleReference </notionalReference> [1]
   'A pointer style reference to the associated notional schedule defined elsewhere in
   the document.'

Start Choice [1]
      <feeAmount> xsd:decimal </feeAmount> [1]
      'The amount of fee to be paid on exercise. The fee currency is that of the referenced notional.'

      <feeRate> xsd:decimal </feeRate> [1]
      'A fee represented as a percentage of some referenced notional. A percentage of 5% would
      be represented as 0.05.'

End Choice
   <feePaymentDate> RelativeDateOffset </feePaymentDate> [1]
   'The date on which exercise fee(s) will be paid. It is specified as a relative date.'
```

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

Diagram



Schema Component Representation

```
<xsd:complexType name="ExerciseFee">
   <xsd:sequence>
      <xsd:group ref=" PayerReceiver.model "/>
      <xsd:element name="notionalReference" type=" ScheduleReference "/>
      <xsd:choice>
         <xsd:element name="feeAmount" type=" xsd:decimal "/>
         <xsd:element name="feeRate" type=" xsd:decimal "/>
      </xsd:choice>
      <xsd:element name="feePaymentDate" type=" RelativeDateOffset "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: ExerciseFeeSchedule

Super-types:	None
Sub-types:	None

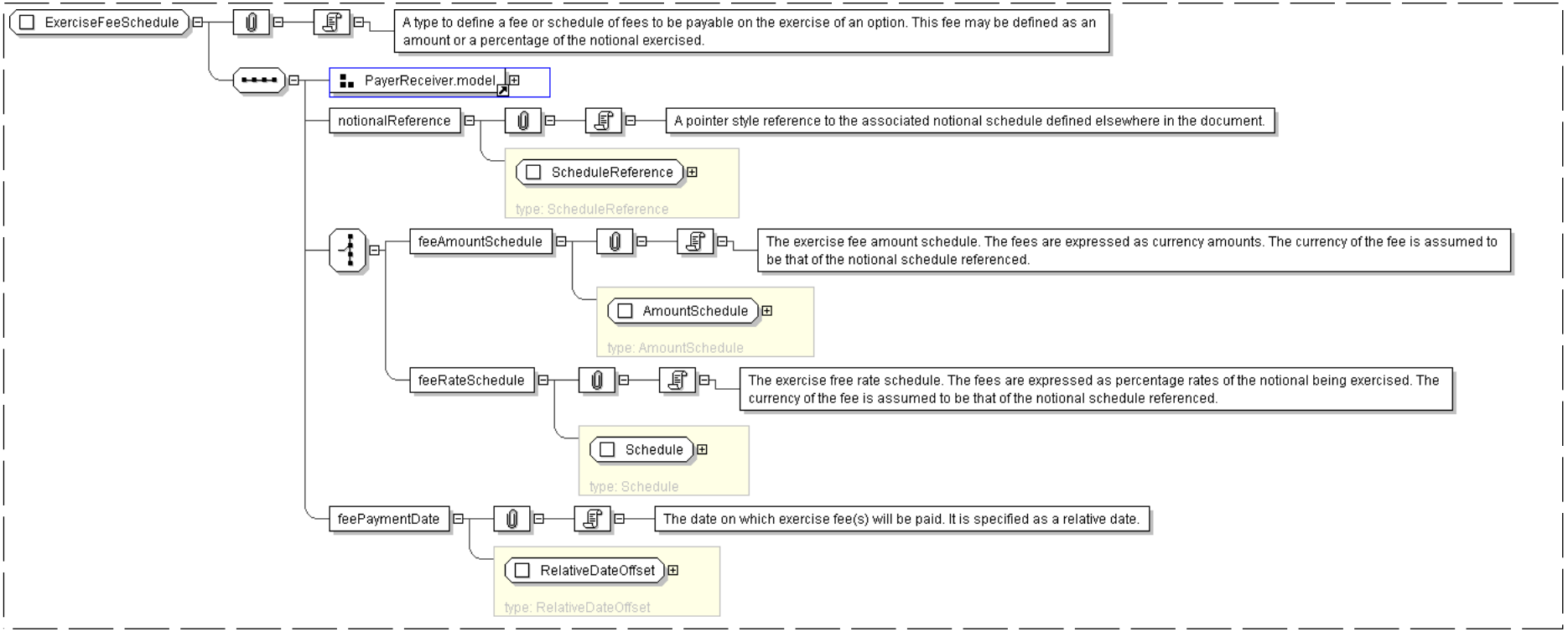
Name	ExerciseFeeSchedule
Used by (from the same schema document)	Complex Type **AmericanExercise** , Complex Type **BermudaExercise**
Abstract	no
Documentation	A type to define a fee or schedule of fees to be payable on the exercise of an option. This fee may be defined as an amount or a percentage of the notional exercised.

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.'

<notionalReference> ScheduleReference </notionalReference> [1]
'A pointer style reference to the associated notional schedule defined elsewhere in
the document.'

Start Choice [1]
    <feeAmountSchedule> AmountSchedule </feeAmountSchedule> [1]
    'The exercise fee amount schedule. The fees are expressed as currency amounts. The currency
    of the fee is assumed to be that of the notional schedule referenced.'

    <feeRateSchedule> Schedule </feeRateSchedule> [1]
    'The exercise free rate schedule. The fees are expressed as percentage rates of the
    notional being exercised. The currency of the fee is assumed to be that of the
    notional schedule referenced.'

End Choice
<feePaymentDate> RelativeDateOffset </feePaymentDate> [1]
'The date on which exercise fee(s) will be paid. It is specified as a relative date.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="ExerciseFeeSchedule">
   <xsd:sequence>
      <xsd:group ref=" PayerReceiver.model "/>
      <xsd:element name="notionalReference" type=" ScheduleReference "/>
      <xsd:choice>
         <xsd:element name="feeAmountSchedule" type=" AmountSchedule "/>
```

```
        <xsd:element name="feeRateSchedule" type=" Schedule "/>
      </xsd:choice>
      <xsd:element name="feePaymentDate" type=" RelativeDateOffset "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: ExerciseNotice

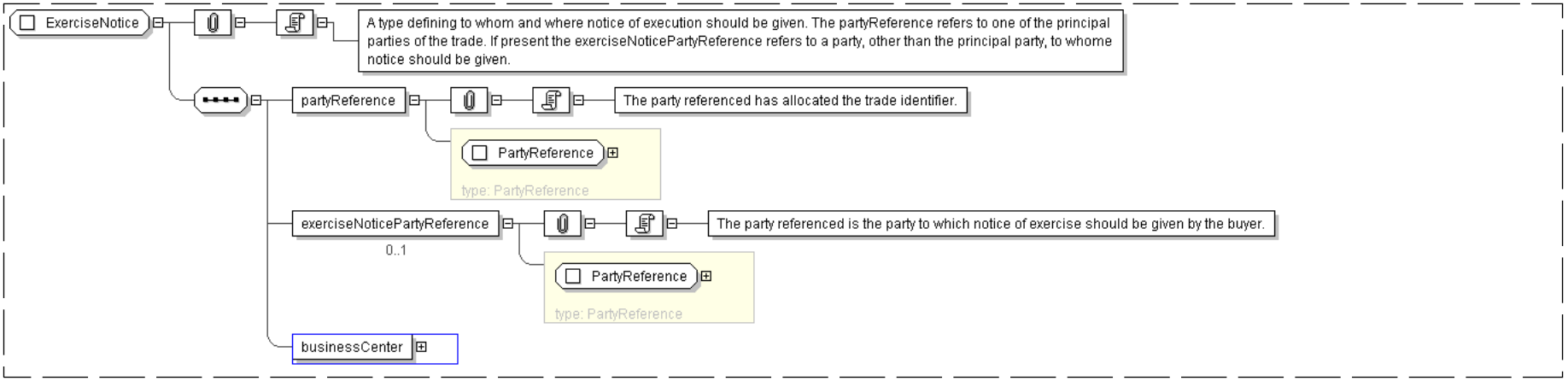
Super-types:	None
Sub-types:	None

Name	ExerciseNotice
Used by (from the same schema document)	Complex Type **ManualExercise**
Abstract	no
Documentation	A type defining to whom and where notice of execution should be given. The partyReference refers to one of the principal parties of the trade. If present the exerciseNoticePartyReference refers to a party, other than the principal party, to whome notice should be given.

XML Instance Representation

```
<...>
   <partyReference> PartyReference </partyReference> [1]
   'The party referenced has allocated the trade identifier.'

   <exerciseNoticePartyReference> PartyReference </exerciseNoticePartyReference> [0..1]
   'The party referenced is the party to which notice of exercise should be given by the buyer.'

   <businessCenter> BusinessCenter </businessCenter> [1]
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="ExerciseNotice">
   <xsd:sequence>
      <xsd:element name="partyReference" type=" PartyReference "/>
      <xsd:element name="exerciseNoticePartyReference" type=" PartyReference " minOccurs="0"/>
      <xsd:element name="businessCenter" type=" BusinessCenter "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: ExerciseProcedure

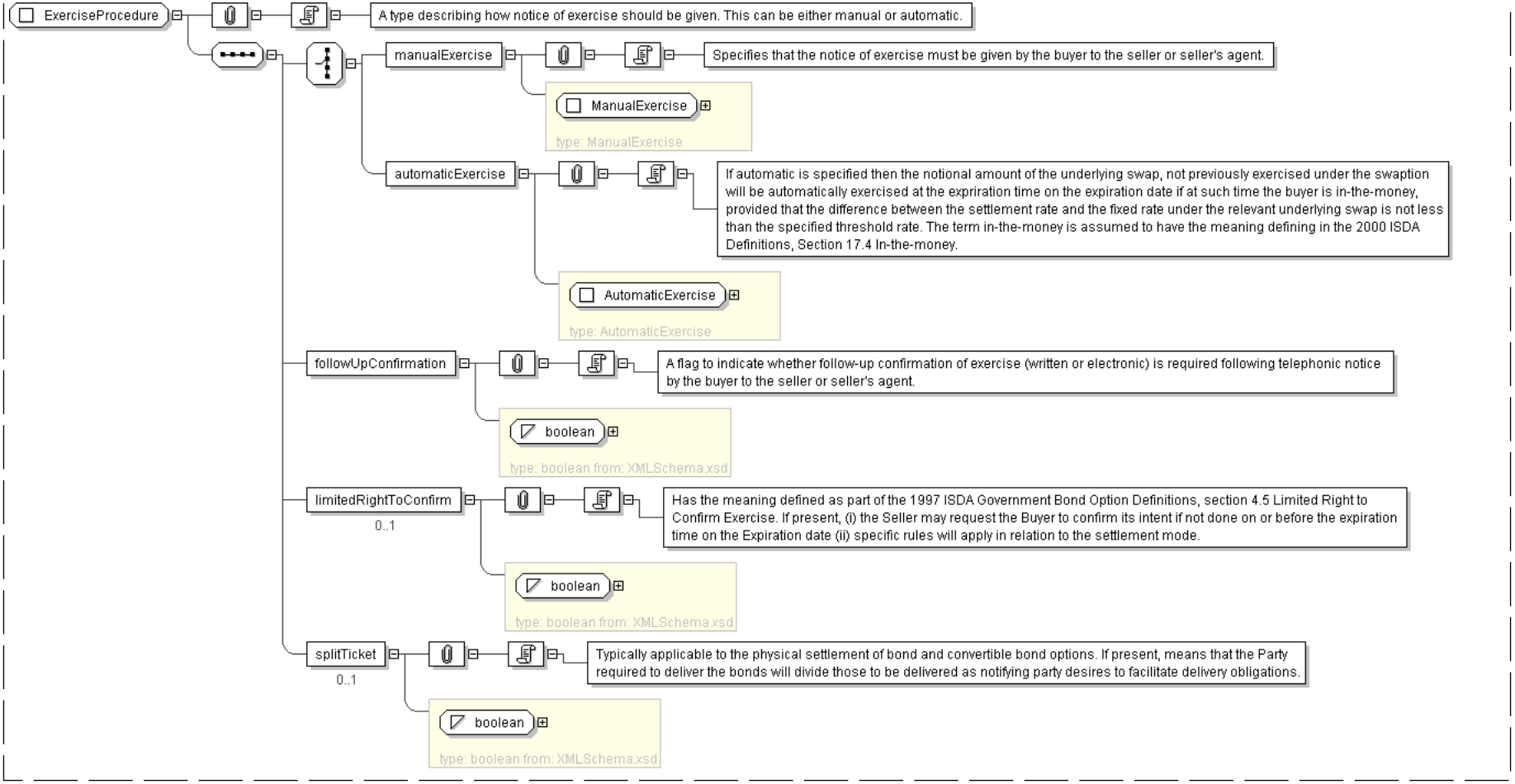
Super-types:	None
Sub-types:	None

Name	ExerciseProcedure
Abstract	no
Documentation	A type describing how notice of exercise should be given. This can be either manual or automatic.

XML Instance Representation

```
<...>
Start Choice [1]
      <manualExercise> ManualExercise </manualExercise> [1]
      'Specifies that the notice of exercise must be given by the buyer to the seller or seller
      \'s agent.'

      <automaticExercise> AutomaticExercise </automaticExercise> [1]
      'If automatic is specified then the notional amount of the underlying swap, not
      previously exercised under the swaption will be automatically exercised at the expriration
      time on the expiration date if at such time the buyer is in-the-money, provided that
      the difference between the settlement rate and the fixed rate under the relevant
      underlying swap is not less than the specified threshold rate. The term in-the-money is
      assumed to have the meaning defining in the 2000 ISDA Definitions, Section 17.4 In-the-money.'

End Choice
   <followUpConfirmation> xsd:boolean </followUpConfirmation> [1]
   'A flag to indicate whether follow-up confirmation of exercise (written or electronic)
   is required following telephonic notice by the buyer to the seller or seller\'s agent.'

   <limitedRightToConfirm> xsd:boolean </limitedRightToConfirm> [0..1]
   'Has the meaning defined as part of the 1997 ISDA Government Bond Option Definitions,
   section 4.5 Limited Right to Confirm Exercise. If present, (i) the Seller may request the
   Buyer to confirm its intent if not done on or before the expiration time on the Expiration
   date (ii) specific rules will apply in relation to the settlement mode.'

   <splitTicket> xsd:boolean </splitTicket> [0..1]
   'Typically applicable to the physical settlement of bond and convertible bond options.
   If present, means that the Party required to deliver the bonds will divide those to
   be delivered as notifying party desires to facilitate delivery obligations.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="ExerciseProcedure">
   <xsd:sequence>
      <xsd:choice>
         <xsd:element name="manualExercise" type=" ManualExercise "/>
         <xsd:element name="automaticExercise" type=" AutomaticExercise "/>
      </xsd:choice>
      <xsd:element name="followUpConfirmation" type=" xsd:boolean "/>
      <xsd:element name="limitedRightToConfirm" type=" xsd:boolean " minOccurs="0"/>
      <xsd:element name="splitTicket" type=" xsd:boolean " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: FloatingRate

Super-types:	Rate < **FloatingRate** (by extension)
Sub-types:	• FloatingRateCalculation (by extension)

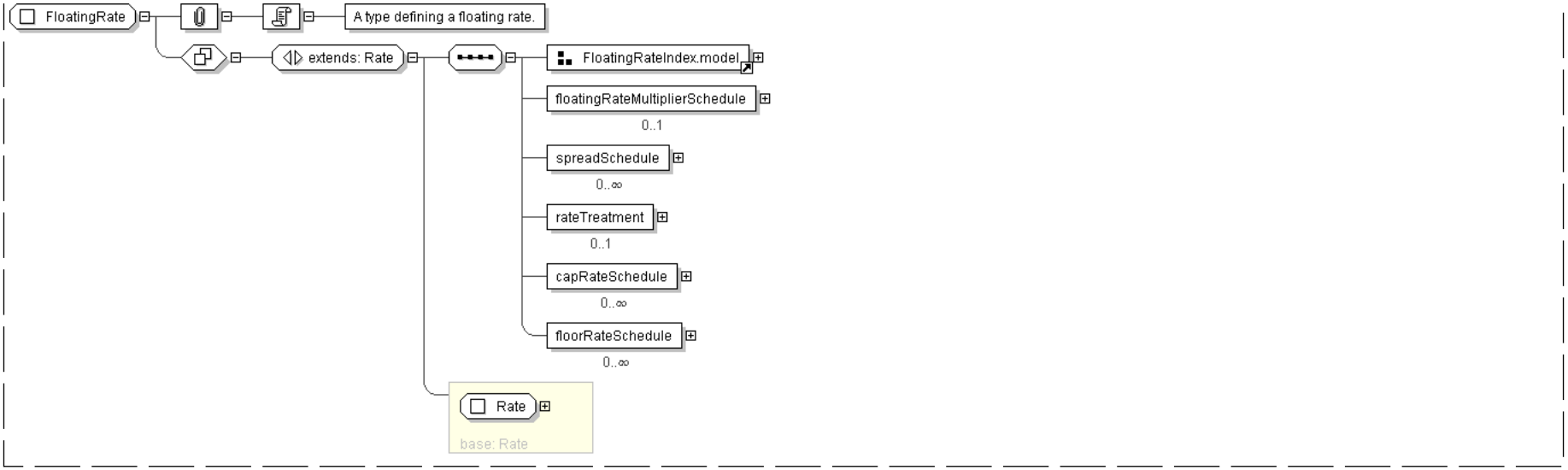
Name	FloatingRate
Used by (from the same schema document)	Complex Type **StubValue**

Abstract	no
Documentation	A type defining a floating rate.

XML Instance Representation

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

   <floatingRateMultiplierSchedule> Schedule </floatingRateMultiplierSchedule> [0..1]
   'A rate multiplier or multiplier schedule to apply to the floating rate. A multiplier
   schedule is expressed as explicit multipliers and dates. In the case of a schedule, the
   step dates may be subject to adjustment in accordance with any adjustments specified in
   the calculationPeriodDatesAdjustments. The multiplier can be a positive or negative
   decimal. This element should only be included if the multiplier is not equal to 1 (one) for
   the term of the stream.'

   <spreadSchedule> SpreadSchedule </spreadSchedule> [0..*]
   'The ISDA Spread or a Spread schedule expressed as explicit spreads and dates. In the case of
   a schedule, the step dates may be subject to adjustment in accordance with any
   adjustments specified in calculationPeriodDatesAdjustments. The spread is a per annum
   rate, expressed as a decimal. For purposes of determining a calculation period amount,
   if positive the spread will be added to the floating rate and if negative the spread will
   be subtracted from the floating rate. A positive 10 basis point (0.1%) spread would
   be represented as 0.001.'

   <rateTreatment> RateTreatmentEnum </rateTreatment> [0..1]
   'The specification of any rate conversion which needs to be applied to the observed rate
   before being used in any calculations. The two common conversions are for securities quoted
   on a bank discount basis which will need to be converted to either a Money Market Yield or
   Bond Equivalent Yield. See the Annex to the 2000 ISDA Definitions, Section 7.3. Certain
   General Definitions Relating to Floating Rate Options, paragraphs (g) and (h) for
   definitions of these terms.'

   <capRateSchedule> StrikeSchedule </capRateSchedule> [0..*]
   'The cap rate or cap rate schedule, if any, which applies to the floating rate. The cap
   rate (strike) is only required where the floating rate on a swap stream is capped at a
   certain level. A cap rate schedule is expressed as explicit cap rates and dates and the
   step dates may be subject to adjustment in accordance with any adjustments specified
   in calculationPeriodDatesAdjustments. The cap rate is assumed to be exclusive of any spread
   and is a per annum rate, expressed as a decimal. A cap rate of 5% would be represented as 0.05.'

   <floorRateSchedule> StrikeSchedule </floorRateSchedule> [0..*]
   'The floor rate or floor rate schedule, if any, which applies to the floating rate. The
   floor rate (strike) is only required where the floating rate on a swap stream is floored at
   a certain strike level. A floor rate schedule is expressed as explicit floor rates and
   dates and the step dates may be subject to adjustment in accordance with any
   adjustments specified in calculationPeriodDatesAdjustments. The floor rate is assumed to
   be exclusive of any spread and is a per annum rate, expressed as a decimal. A floor rate of
   5% would be represented as 0.05.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="FloatingRate">
   <xsd:complexContent>
      <xsd:extension base=" Rate ">
         <xsd:sequence>
            <xsd:group ref=" FloatingRateIndex.model "/>
            <xsd:element name="floatingRateMultiplierSchedule" type=" Schedule " minOccurs="0"/>
            <xsd:element name="spreadSchedule" type=" SpreadSchedule " minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="rateTreatment" type=" RateTreatmentEnum " minOccurs="0"/>
            <xsd:element name="capRateSchedule" type=" StrikeSchedule " minOccurs="0"
            maxOccurs="unbounded"/>
            <xsd:element name="floorRateSchedule" type=" StrikeSchedule "
            minOccurs="0" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: FloatingRateCalculation

Super-types:	Rate < FloatingRate (by extension) < **FloatingRateCalculation** (by extension)
Sub-types:	None

Name	FloatingRateCalculation
Used by (from the same schema document)	Complex Type **InterestAccrualsMethod**
Abstract	no
Documentation	A type defining the floating rate and definitions relating to the calculation of floating rate amounts.

XML Instance Representation

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

   <floatingRateMultiplierSchedule> Schedule </floatingRateMultiplierSchedule> [0..1]
   'A rate multiplier or multiplier schedule to apply to the floating rate. A multiplier
   schedule is expressed as explicit multipliers and dates. In the case of a schedule, the
```

step dates may be subject to adjustment in accordance with any adjustments specified in the calculationPeriodDatesAdjustments. The multiplier can be a positive or negative decimal. This element should only be included if the multiplier is not equal to 1 (one) for the term of the stream.'

<spreadSchedule> SpreadSchedule </spreadSchedule> [0..*]

'The ISDA Spread or a Spread schedule expressed as explicit spreads and dates. In the case of a schedule, the step dates may be subject to adjustment in accordance with any adjustments specified in calculationPeriodDatesAdjustments. The spread is a per annum rate, expressed as a decimal. For purposes of determining a calculation period amount, if positive the spread will be added to the floating rate and if negative the spread will be subtracted from the floating rate. A positive 10 basis point (0.1%) spread would be represented as 0.001.'

<rateTreatment> RateTreatmentEnum </rateTreatment> [0..1]

'The specification of any rate conversion which needs to be applied to the observed rate before being used in any calculations. The two common conversions are for securities quoted on a bank discount basis which will need to be converted to either a Money Market Yield or Bond Equivalent Yield. See the Annex to the 2000 ISDA Definitions, Section 7.3. Certain General Definitions Relating to Floating Rate Options, paragraphs (g) and (h) for definitions of these terms.'

<capRateSchedule> StrikeSchedule </capRateSchedule> [0..*]

'The cap rate or cap rate schedule, if any, which applies to the floating rate. The cap rate (strike) is only required where the floating rate on a swap stream is capped at a certain level. A cap rate schedule is expressed as explicit cap rates and dates and the step dates may be subject to adjustment in accordance with any adjustments specified in calculationPeriodDatesAdjustments. The cap rate is assumed to be exclusive of any spread and is a per annum rate, expressed as a decimal. A cap rate of 5% would be represented as 0.05.'

<floorRateSchedule> StrikeSchedule </floorRateSchedule> [0..*]

'The floor rate or floor rate schedule, if any, which applies to the floating rate. The floor rate (strike) is only required where the floating rate on a swap stream is floored at a certain strike level. A floor rate schedule is expressed as explicit floor rates and dates and the step dates may be subject to adjustment in accordance with any adjustments specified in calculationPeriodDatesAdjustments. The floor rate is assumed to be exclusive of any spread and is a per annum rate, expressed as a decimal. A floor rate of 5% would be represented as 0.05.'

<initialRate> xsd:decimal **</initialRate>** [0..1]

'The initial floating rate reset agreed between the principal parties involved in the trade. This is assumed to be the first required reset rate for the first regular calculation period. It should only be included when the rate is not equal to the rate published on the source implied by the floating rate index. An initial rate of 5% would be represented as 0.05.'

<finalRateRounding> Rounding **</finalRateRounding>** [0..1]

'The rounding convention to apply to the final rate used in determination of a calculation period amount.'

<averagingMethod> AveragingMethodEnum **</averagingMethod>** [0..1]

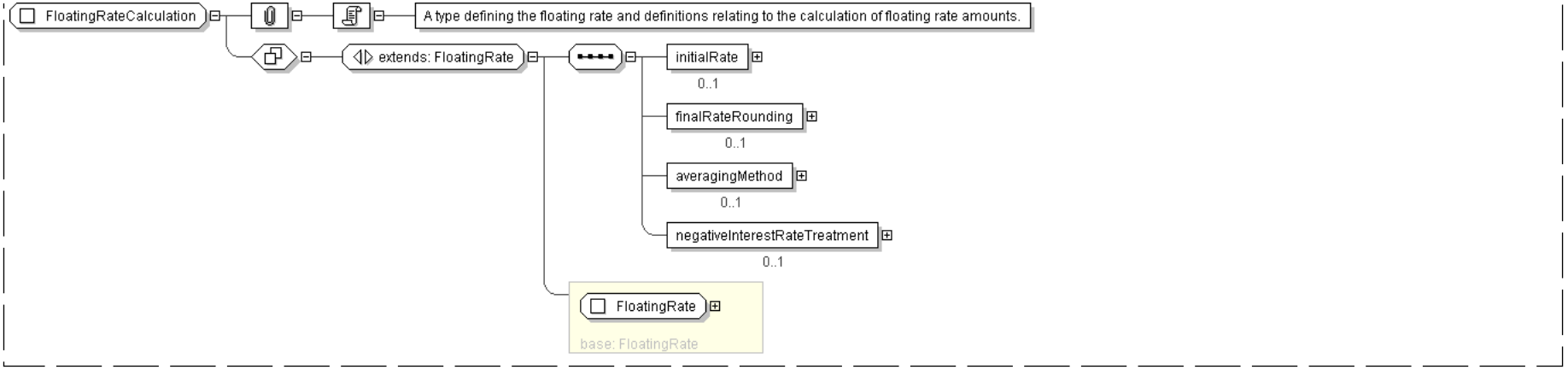
'If averaging is applicable, this component specifies whether a weighted or unweighted average method of calculation is to be used. The component must only be included when averaging applies.'

<negativeInterestRateTreatment> NegativeInterestRateTreatmentEnum **</negativeInterestRateTreatment>** [0..1]

'The specification of any provisions for calculating payment obligations when a floating rate is negative (either due to a quoted negative floating rate or by operation of a spread that is subtracted from the floating rate).'

</...>

Diagram



Schema Component Representation

```
<xsd:complexType name="FloatingRateCalculation">
   <xsd:complexContent>
      <xsd:extension base=" FloatingRate ">
         <xsd:sequence>
            <xsd:element name="initialRate" type=" xsd:decimal " minOccurs="0"/>
            <xsd:element name="finalRateRounding" type=" Rounding " minOccurs="0"/>
            <xsd:element name="averagingMethod" type=" AveragingMethodEnum " minOccurs="0"/>
            <xsd:element name="negativeInterestRateTreatment" type=" NegativeInterestRateTreatmentEnum
            " minOccurs="0"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: FloatingRateIndex

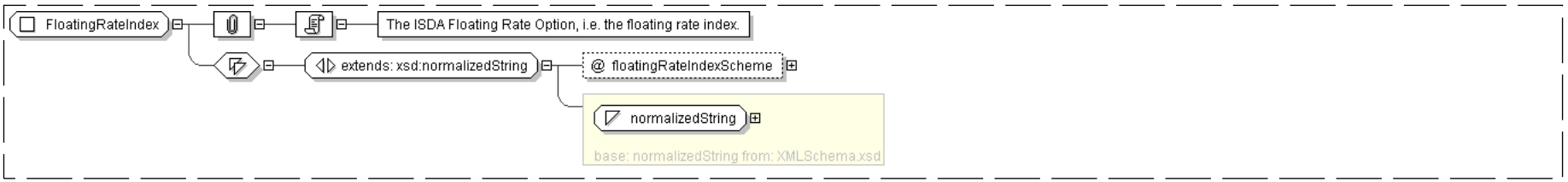
Super-types:	xsd:normalizedString < **FloatingRateIndex** (by extension)
Sub-types:	None

Name	FloatingRateIndex
Used by (from the same schema document)	Complex Type **ForecastRateIndex** , Model Group **FloatingRateIndex.model**
Abstract	no
Documentation	The ISDA Floating Rate Option, i.e. the floating rate index.

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="FloatingRateIndex">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="floatingRateIndexScheme" type=" xsd:anyURI " default="http://www.fpml.
         org/coding-scheme/floating-rate-index-2-0"/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: ForecastRateIndex

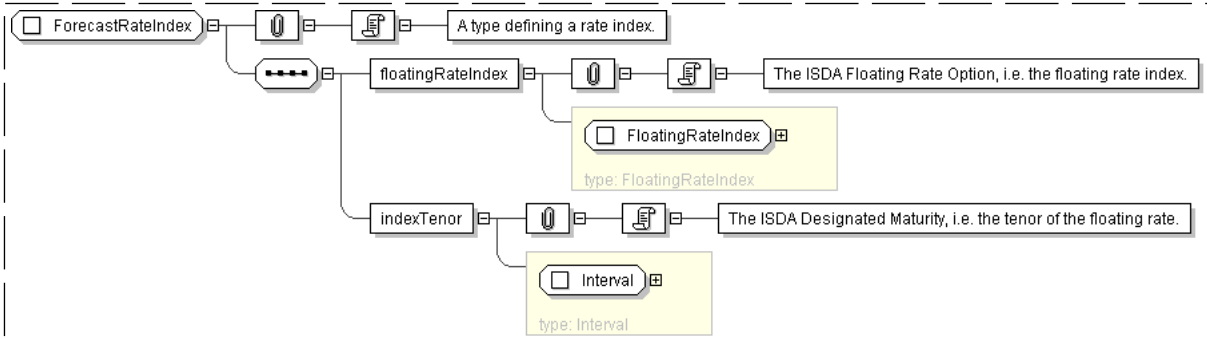
Super-types:	None
Sub-types:	None

Name	ForecastRateIndex
Abstract	no
Documentation	A type defining a rate index.

XML Instance Representation

```
<...>
   <floatingRateIndex> FloatingRateIndex </floatingRateIndex> [1]
   'The ISDA Floating Rate Option, i.e. the floating rate index.'

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

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="ForecastRateIndex">
   <xsd:sequence>
      <xsd:element name="floatingRateIndex" type=" FloatingRateIndex "/>
      <xsd:element name="indexTenor" type=" Interval "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: Formula

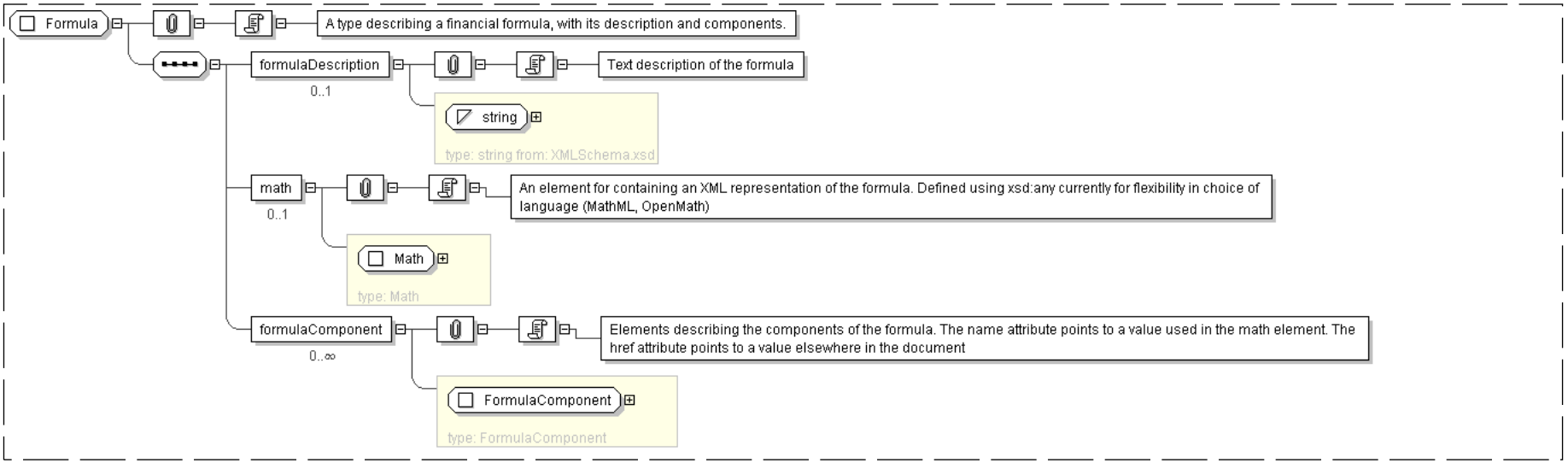
Super-types:	None
Sub-types:	None

Name	Formula
Used by (from the same schema document)	Complex Type **FormulaComponent**
Abstract	no
Documentation	A type describing a financial formula, with its description and components.

XML Instance Representation

```
<...>
   <formulaDescription> xsd:string </formulaDescription> [0..1]
   'Text description of the formula'

   <math> Math </math> [0..1]
   'An element for containing an XML representation of the formula. Defined using xsd:
   any currently for flexibility in choice of language (MathML, OpenMath)'

   <formulaComponent> FormulaComponent </formulaComponent> [0..*]
   'Elements describing the components of the formula. The name attribute points to a value
   used in the math element. The href attribute points to a value elsewhere in the document'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Formula">
   <xsd:sequence>
      <xsd:element name="formulaDescription" type=" xsd:string " minOccurs="0"/>
      <xsd:element name="math" type=" Math " minOccurs="0"/>
      <xsd:element name="formulaComponent" type=" FormulaComponent "
      minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: FormulaComponent

Super-types:	None
Sub-types:	None

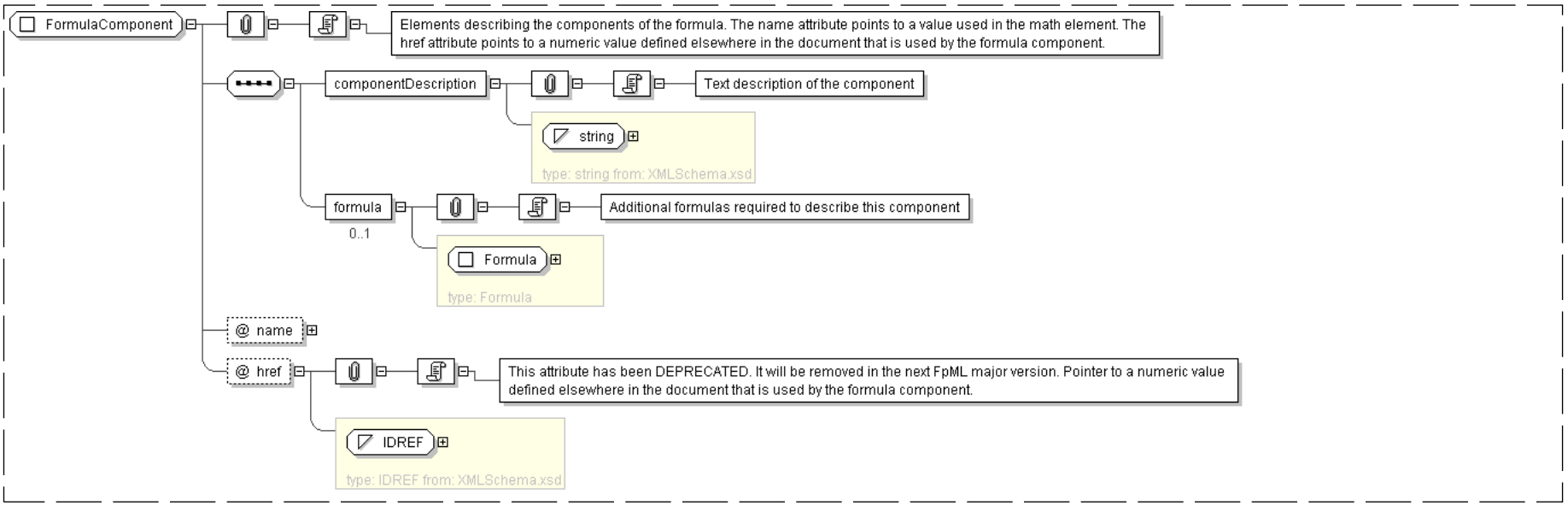
Name	FormulaComponent

Used by (from the same schema document)	Complex Type **Formula**
Abstract	no
Documentation	Elements describing the components of the formula. The name attribute points to a value used in the math element. The href attribute points to a numeric value defined elsewhere in the document that is used by the formula component.

XML Instance Representation

```
<...
name=" xsd:normalizedString [0..1]"
href=" xsd:IDREF [0..1]
 'This attribute has been DEPRECATED. It will be removed in the next FpML major version.
 Pointer to a numeric value defined elsewhere in the document that is used by the
 formula component.'

">
   <componentDescription> xsd:string </componentDescription> [1]
    'Text description of the component'

   <formula> Formula </formula> [0..1]
    'Additional formulas required to describe this component'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="FormulaComponent">
   <xsd:sequence>
      <xsd:element name="componentDescription" type=" xsd:string "/>
      <xsd:element name="formula" type=" Formula " minOccurs="0"/>
   </xsd:sequence>
   <xsd:attribute name="name" type=" xsd:normalizedString "/>
   <xsd:attribute name="href" type=" xsd:IDREF " deprecated="true" deprecatedReason="There was
   no definition on where this attribute should point at and no clear usage."/>
</xsd:complexType>
```

top

Complex Type: FxCashSettlement

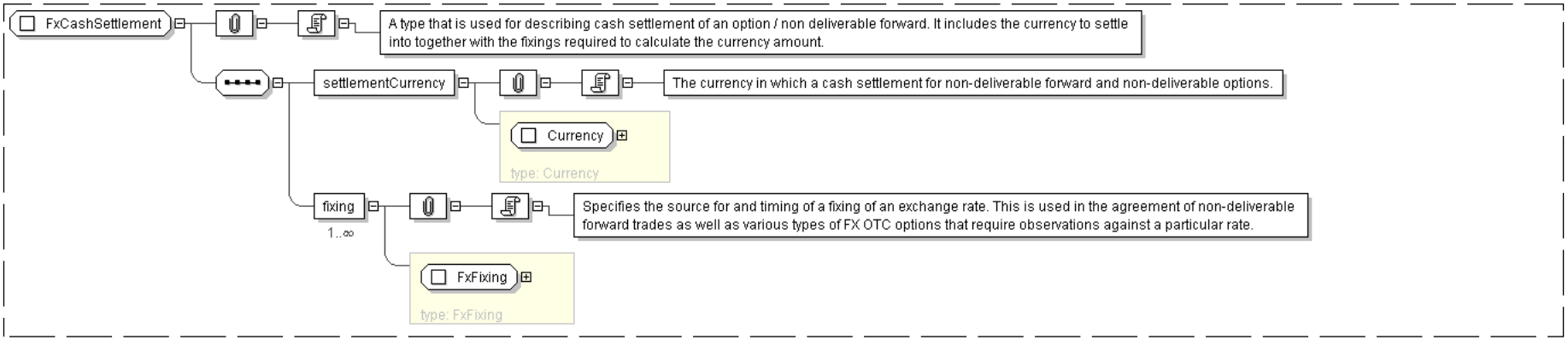
Super-types:	None
Sub-types:	None

Name	FxCashSettlement
Abstract	no
Documentation	A type that is used for describing cash settlement of an option / non deliverable forward. It includes the currency to settle into together with the fixings required to calculate the currency amount.

XML Instance Representation

```
<...>
   <settlementCurrency> Currency </settlementCurrency> [1]
   'The currency in which a cash settlement for non-deliverable forward and non-
   deliverable options.'

   <fixing> FxFixing </fixing> [1..*]
   'Specifies the source for and timing of a fixing of an exchange rate. This is used in
   the agreement of non-deliverable forward trades as well as various types of FX OTC options
   that require observations against a particular rate.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="FxCashSettlement">
   <xsd:sequence>
      <xsd:element name="settlementCurrency" type=" Currency "/>
      <xsd:element name="fixing" type=" FxFixing " maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: FxFixing

Super-types:	FxSpotRateSource < **FxFixing** (by extension)
Sub-types:	None

Name	FxFixing
Used by (from the same schema document)	Complex Type **FxCashSettlement**
Abstract	no
Documentation	A type that specifies the source for and timing of a fixing of an exchange rate. This is used in the agreement of non-deliverable forward trades as well as various types of FX OTC options that require observations against a particular rate.

XML Instance Representation

```
<...>
```

```
<primaryRateSource> InformationSource </primaryRateSource> [1]
```

'The primary source for where the rate observation will occur. Will typically be either a page or a reference bank published rate.'

```
<secondaryRateSource> InformationSource </secondaryRateSource> [0..1]
```

'An alternative, or secondary, source for where the rate observation will occur. Will typically be either a page or a reference bank published rate.'

```
<fixingTime> BusinessCenterTime </fixingTime> [1]
```

'The time at which the spot currency exchange rate will be observed. It is specified as a time in a specific business center, e.g. 11:00am London time.'

```
<quotedCurrencyPair> QuotedCurrencyPair </quotedCurrencyPair> [1]
```

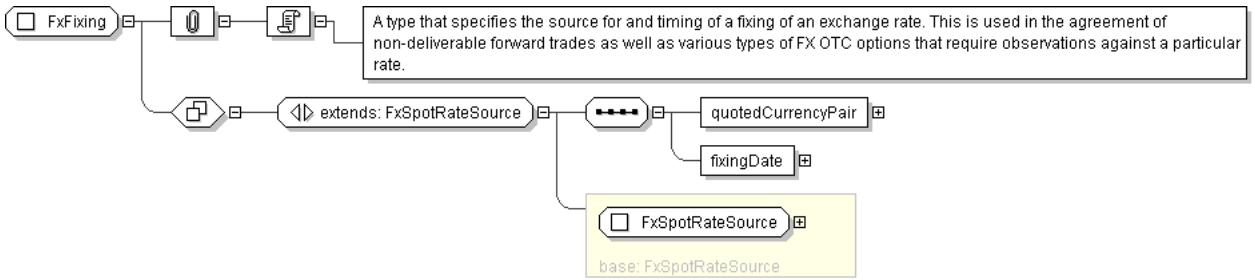
'Defines the two currencies for an FX trade and the quotation relationship between the two currencies.'

```
<fixingDate> xsd:date </fixingDate> [1]
```

'Describes the specific date when a non-deliverable forward or non-deliverable option will \"fix\" against a particular rate, which will be used to compute the ultimate cash settlement.'

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

Diagram



Schema Component Representation

```
<xsd:complexType name="FxFixing">
   <xsd:complexContent>
      <xsd:extension base=" FxSpotRateSource ">
         <xsd:sequence>
            <xsd:element name="quotedCurrencyPair" type=" QuotedCurrencyPair "/>
            <xsd:element name="fixingDate" type=" xsd:date "/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: FxRate

Super-types:	None
Sub-types:	None

Name	FxRate
Abstract	no
Documentation	A type describing the rate of a currency conversion: pair of currency, quotation mode and exchange rate.

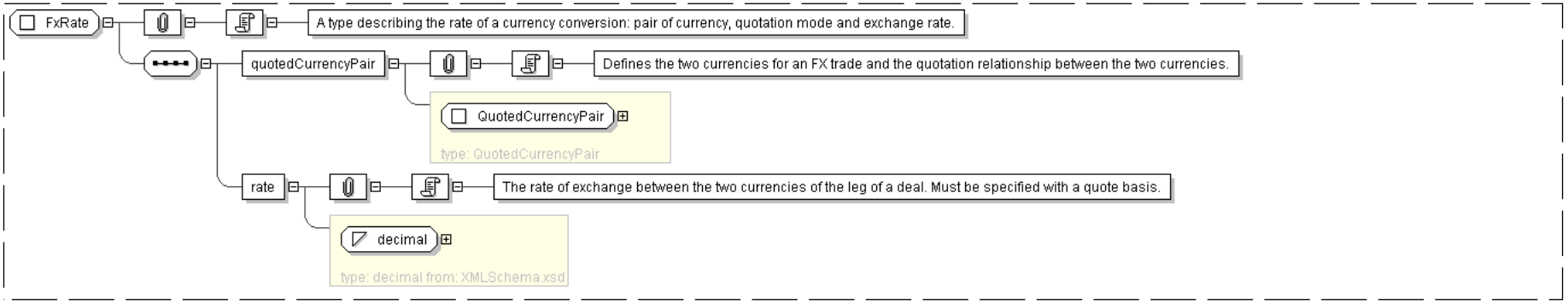
XML Instance Representation

```
<...>
   <quotedCurrencyPair> QuotedCurrencyPair </quotedCurrencyPair> [1]
```

'Defines the two currencies for an FX trade and the quotation relationship between the

```
    two currencies.'

    <rate> xsd:decimal </rate> [1]
    'The rate of exchange between the two currencies of the leg of a deal. Must be specified with
    a quote basis.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="FxRate">
   <xsd:sequence>
      <xsd:element name="quotedCurrencyPair" type=" QuotedCurrencyPair "/>
      <xsd:element name="rate" type=" xsd:decimal "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: FxSpotRateSource

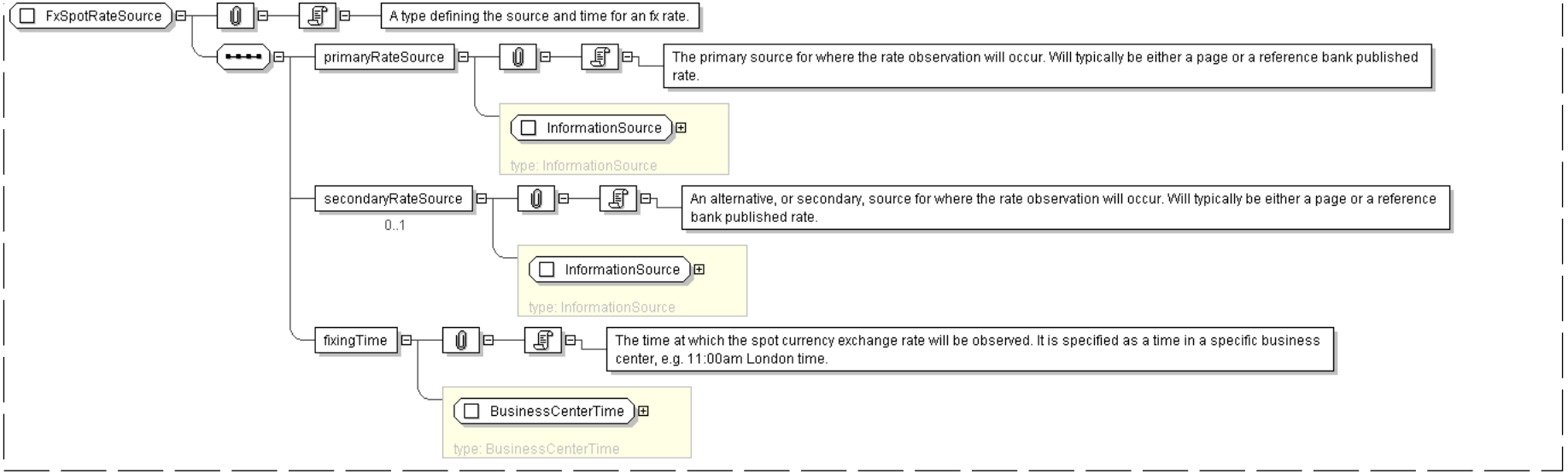
Super-types:	None
Sub-types:	• FxFixing (by extension)

Name	FxSpotRateSource
Abstract	no
Documentation	A type defining the source and time for an fx rate.

XML Instance Representation

```
<...>
    <primaryRateSource> InformationSource </primaryRateSource> [1]
    'The primary source for where the rate observation will occur. Will typically be either a
    page or a reference bank published rate.'

    <secondaryRateSource> InformationSource </secondaryRateSource> [0..1]
    'An alternative, or secondary, source for where the rate observation will occur. Will
    typically be either a page or a reference bank published rate.'

    <fixingTime> BusinessCenterTime </fixingTime> [1]
    'The time at which the spot currency exchange rate will be observed. It is specified as a
    time in a specific business center, e.g. 11:00am London time.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="FxSpotRateSource">
   <xsd:sequence>
      <xsd:element name="primaryRateSource" type=" InformationSource "/>
      <xsd:element name="secondaryRateSource" type=" InformationSource " minOccurs="0"/>
      <xsd:element name="fixingTime" type=" BusinessCenterTime "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: GoverningLaw

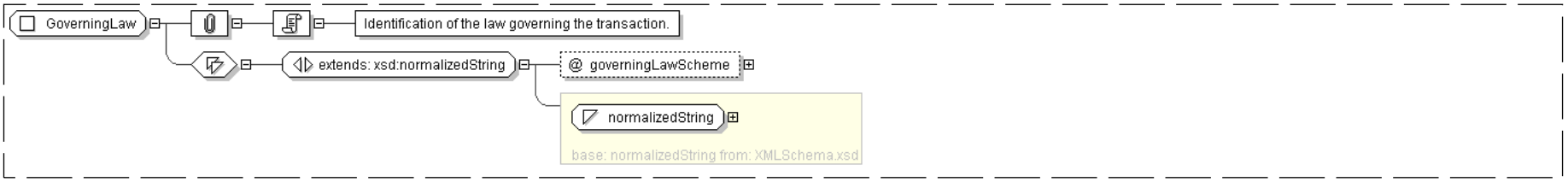
Super-types:	xsd:normalizedString < **GoverningLaw** (by extension)
Sub-types:	None

Name	GoverningLaw
Abstract	no
Documentation	Identification of the law governing the transaction.

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="GoverningLaw">
```

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

top

Complex Type: IdentifiedCurrency

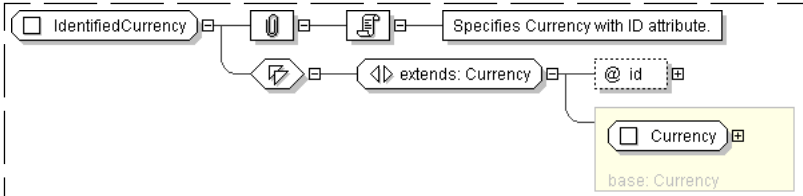
Super-types:	xsd:normalizedString < Currency (by extension) < **IdentifiedCurrency** (by extension)
Sub-types:	None

Name	IdentifiedCurrency
Abstract	no
Documentation	Specifies Currency with ID attribute.

XML Instance Representation

```
<...
currencyScheme=" xsd:anyURI [0..1]"
id=" xsd:ID [0..1]">
Currency
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="IdentifiedCurrency">
   <xsd:simpleContent>
      <xsd:extension base=" Currency ">
         <xsd:attribute name="id" type=" xsd:ID "/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: IdentifiedCurrencyReference

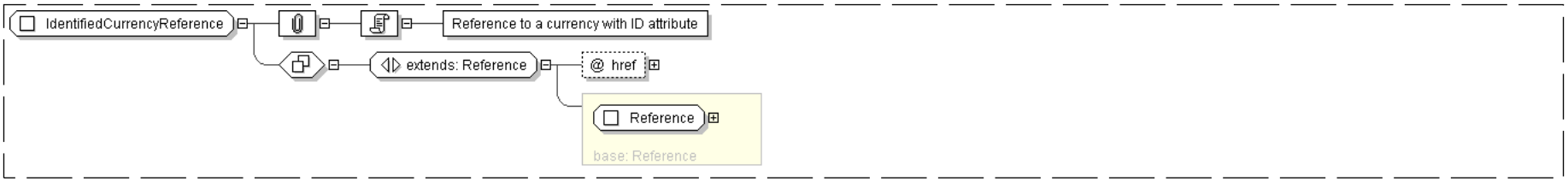
Super-types:	Reference < **IdentifiedCurrencyReference** (by extension)
Sub-types:	None

Name	IdentifiedCurrencyReference
Used by (from the same schema document)	Complex Type **DividendConditions**
Abstract	no
Documentation	Reference to a currency with ID attribute

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: IdentifiedDate

Super-types:	xsd:date < **IdentifiedDate** (by extension)
Sub-types:	None

Name	IdentifiedDate
Used by (from the same schema document)	Complex Type **AdjustableDate** , Complex Type **AdjustableDate2** , Complex Type **AdjustableDates** , Complex Type **AdjustableOrRelativeAndAdjustedDate** , Complex Type **Payment** , Model Group **VersionHistory.model**
Abstract	no
Documentation	A date which can be referenced elsewhere.

XML Instance Representation

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

Diagram

Schema Component Representation

```
<xsd:complexType name="IdentifiedDate">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:date ">
         <xsd:attribute name="id" type=" xsd:ID "/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: IdentifiedPayerReceiver

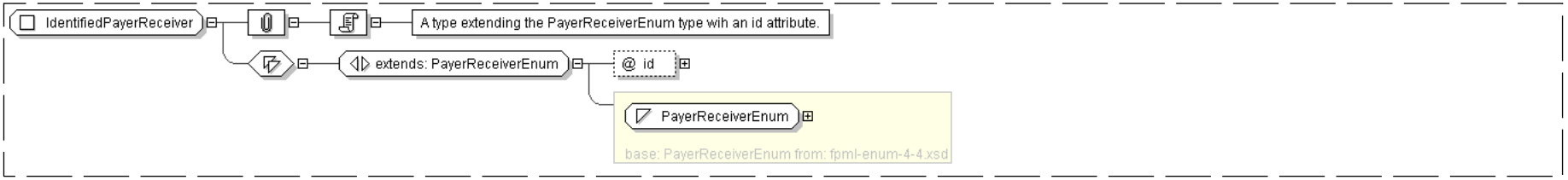
Super-types:	PayerReceiverEnum < **IdentifiedPayerReceiver** (by extension)
Sub-types:	None

Name	IdentifiedPayerReceiver
Used by (from the same schema document)	Complex Type **Strike** , Complex Type **Strike** , Complex Type **StrikeSchedule** , Complex Type **StrikeSchedule**
Abstract	no
Documentation	A type extending the PayerReceiverEnum type wih an id attribute.

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="IdentifiedPayerReceiver">
   <xsd:simpleContent>
      <xsd:extension base=" PayerReceiverEnum ">
         <xsd:attribute name="id" type=" xsd:ID "/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: InformationProvider

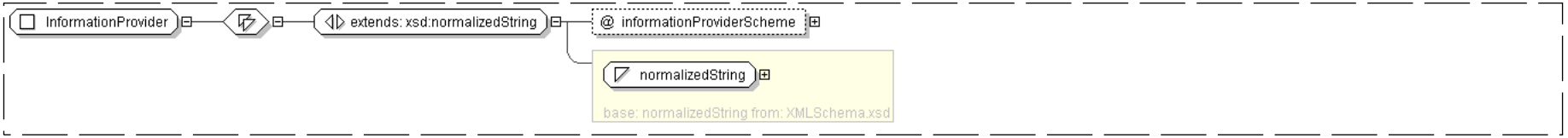
Super-types:	xsd:normalizedString < **InformationProvider** (by extension)
Sub-types:	None

Name	InformationProvider
Used by (from the same schema document)	Complex Type **InformationSource**
Abstract	no

XML Instance Representation

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

Diagram



Schema Component Representation

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

```
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="informationProviderScheme" type=" xsd:anyURI " default="http://www.
         fpml.org/coding-scheme/information-provider-2-0"/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: InformationSource

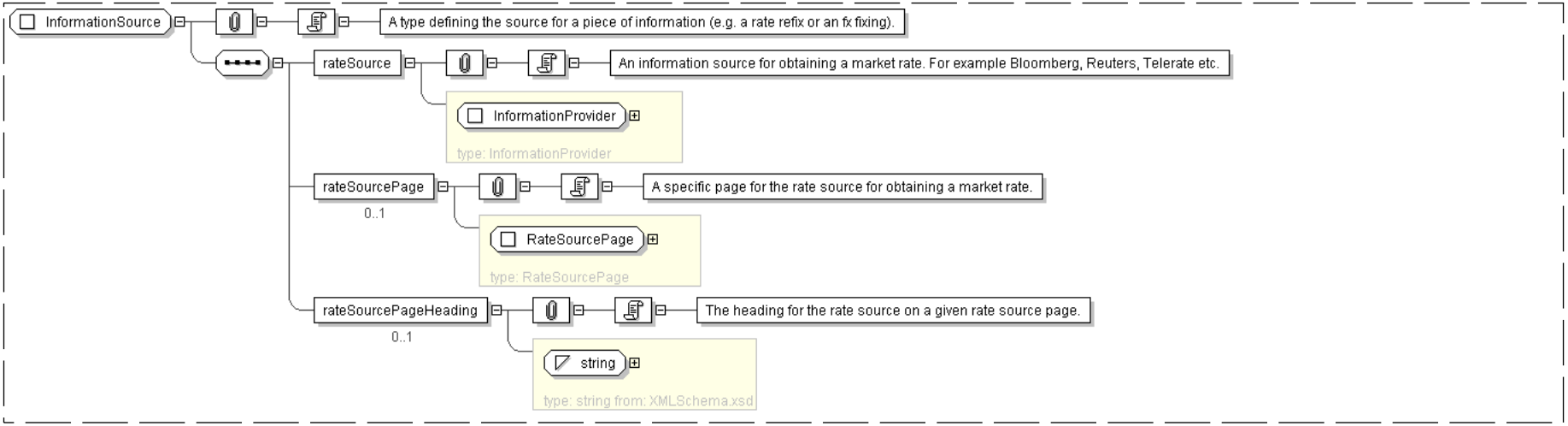
Super-types:	None
Sub-types:	None

Name	InformationSource
Used by (from the same schema document)	Complex Type **FxSpotRateSource** , Complex Type **FxSpotRateSource** , Complex Type **SettlementRateSource**
Abstract	no
Documentation	A type defining the source for a piece of information (e.g. a rate refix or an fx fixing).

XML Instance Representation

```
<...>
   <rateSource> InformationProvider </rateSource> [1]
   'An information source for obtaining a market rate. For example Bloomberg, Reuters,
   Telerate etc.'

   <rateSourcePage> RateSourcePage </rateSourcePage> [0..1]
   'A specific page for the rate source for obtaining a market rate.'

   <rateSourcePageHeading> xsd:string </rateSourcePageHeading> [0..1]
   'The heading for the rate source on a given rate source page.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="InformationSource">
   <xsd:sequence>
      <xsd:element name="rateSource" type=" InformationProvider "/>
      <xsd:element name="rateSourcePage" type=" RateSourcePage " minOccurs="0"/>
      <xsd:element name="rateSourcePageHeading" type=" xsd:string " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: InstrumentId

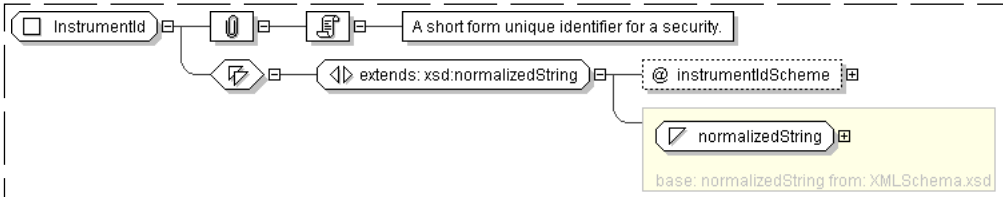
Super-types:	xsd:normalizedString < **InstrumentId** (by extension)
Sub-types:	None

Name	InstrumentId
Abstract	no
Documentation	A short form unique identifier for a security.

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: InterestAccrualsCompoundingMethod

Super-types:	InterestAccrualsMethod < **InterestAccrualsCompoundingMethod** (by extension)
Sub-types:	None

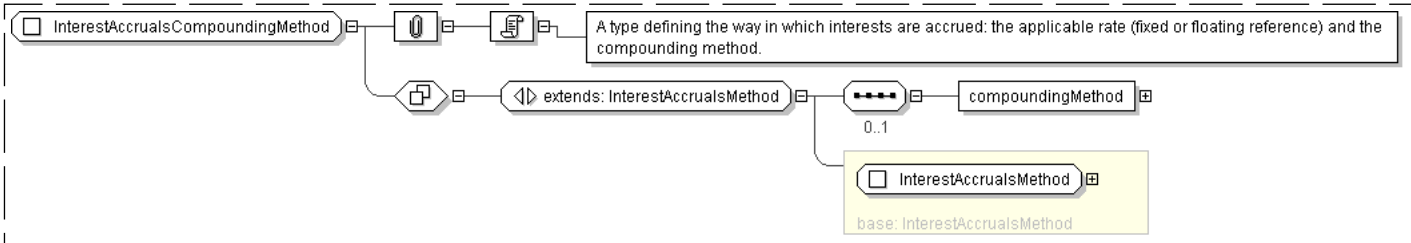
Name	InterestAccrualsCompoundingMethod
Used by (from the same schema document)	Complex Type **DividendConditions**
Abstract	no
Documentation	A type defining the way in which interests are accrued: the applicable rate (fixed or floating reference) and the compounding method.

XML Instance Representation

```
<...>
Start Choice [1]
      <floatingRateCalculation> FloatingRateCalculation </floatingRateCalculation> [1]
      'The floating rate calculation definitions'

      <fixedRate> xsd:decimal </fixedRate> [1]
      'The calculation period fixed rate. A per annum rate, expressed as a decimal. A fixed rate
      of 5% would be represented as 0.05.'

End Choice
```

```
Start Sequence [0..1]
      <compoundingMethod> CompoundingMethodEnum </compoundingMethod> [1]
      'If more that one calculation period contributes to a single payment amount this
      element specifies whether compounding is applicable, and if so, what compounding method is
      to be used. This element must only be included when more that one calculation
      period contributes to a single payment amount.'

End Sequence
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="InterestAccrualsCompoundingMethod">
   <xsd:complexContent>
      <xsd:extension base=" InterestAccrualsMethod ">
         <xsd:sequence minOccurs="0">
            <xsd:element name="compoundingMethod" type=" CompoundingMethodEnum "/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: InterestAccrualsMethod

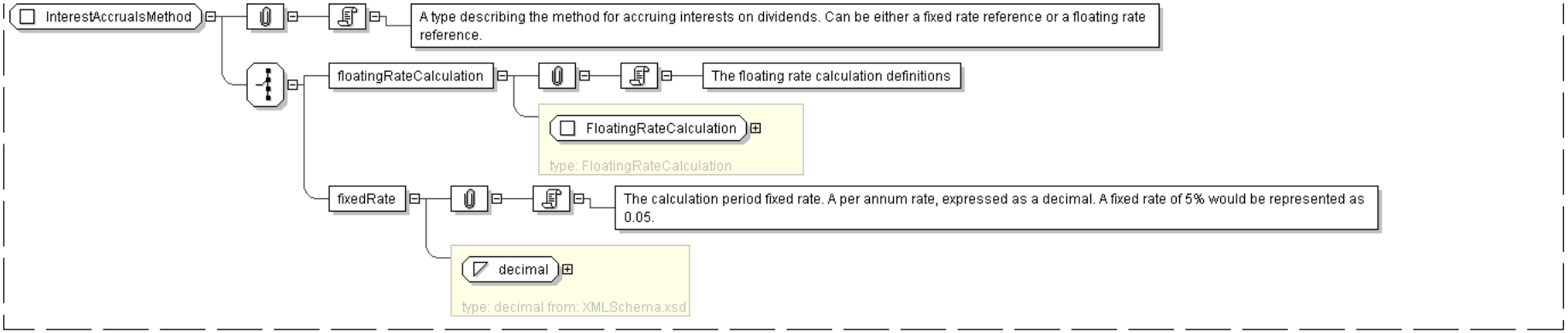
Super-types:	None
Sub-types:	• InterestAccrualsCompoundingMethod (by extension)

Name	InterestAccrualsMethod
Abstract	no
Documentation	A type describing the method for accruing interests on dividends. Can be either a fixed rate reference or a floating rate reference.

XML Instance Representation

```
<...>
Start Choice [1]
      <floatingRateCalculation> FloatingRateCalculation </floatingRateCalculation> [1]
      'The floating rate calculation definitions'

      <fixedRate> xsd:decimal </fixedRate> [1]
      'The calculation period fixed rate. A per annum rate, expressed as a decimal. A fixed rate
      of 5% would be represented as 0.05.'

End Choice
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="InterestAccrualsMethod">
   <xsd:choice>
      <xsd:element name="floatingRateCalculation" type=" FloatingRateCalculation "/>
      <xsd:element name="fixedRate" type=" xsd:decimal "/>
   </xsd:choice>
</xsd:complexType>
```

top

Complex Type: IntermediaryInformation

Super-types:	None
Sub-types:	None

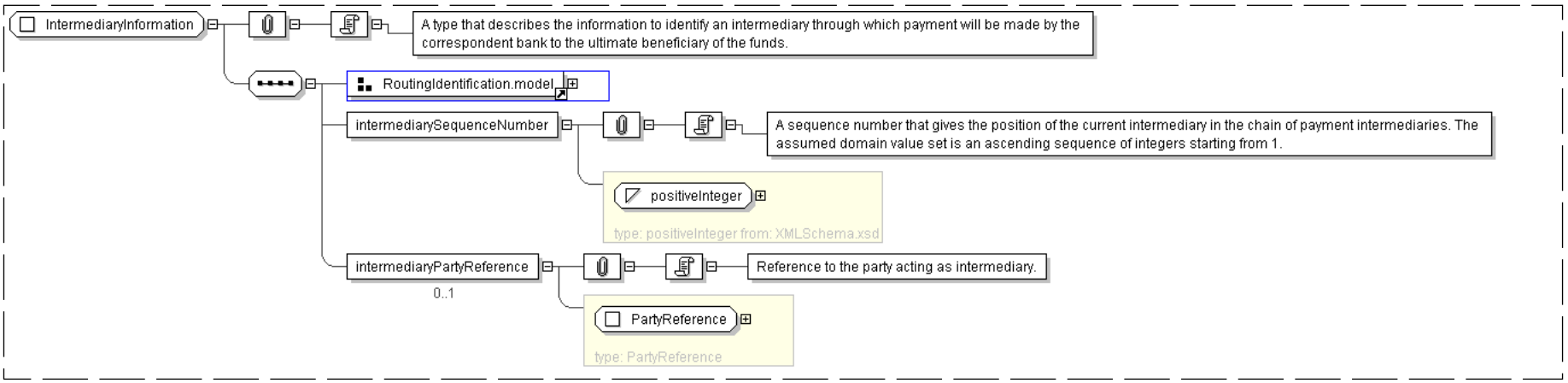
Name	IntermediaryInformation
Used by (from the same schema document)	Complex Type **SettlementInstruction**
Abstract	no
Documentation	A type that describes the information to identify an intermediary through which payment will be made by the correspondent bank to the ultimate beneficiary of the funds.

XML Instance Representation

```
<...>
Start Choice [1]
      <routingIds> RoutingIds </routingIds> [1]
      'A set of unique identifiers for a party, eachone identifying the party within a
      payment system. The assumption is that each party will not have more than one identifier
      within the same payment system.'

      <routingExplicitDetails> RoutingExplicitDetails </routingExplicitDetails> [1]
      'A set of details that is used to identify a party involved in the routing of a payment
      when the party does not have a code that identifies it within one of the recognized
      payment systems.'

      <routingIdsAndExplicitDetails> RoutingIdsAndExplicitDetails </routingIdsAndExplicitDetails> [1]
      'A combination of coded payment system identifiers and details for physical addressing for
      a party involved in the routing of a payment.'

End Choice
   <intermediarySequenceNumber> xsd:positiveInteger </intermediarySequenceNumber> [1]
   'A sequence number that gives the position of the current intermediary in the chain of
   payment intermediaries. The assumed domain value set is an ascending sequence of
   integers starting from 1.'

   <intermediaryPartyReference> PartyReference </intermediaryPartyReference> [0..1]
   'Reference to the party acting as intermediary.'
```

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

Diagram



Schema Component Representation

```
<xsd:complexType name="IntermediaryInformation">
   <xsd:sequence>
      <xsd:group ref=" RoutingIdentification.model "/>
      <xsd:element name="intermediarySequenceNumber" type=" xsd:positiveInteger "/>
      <xsd:element name="intermediaryPartyReference" type=" PartyReference " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: Interval

Super-types:	None

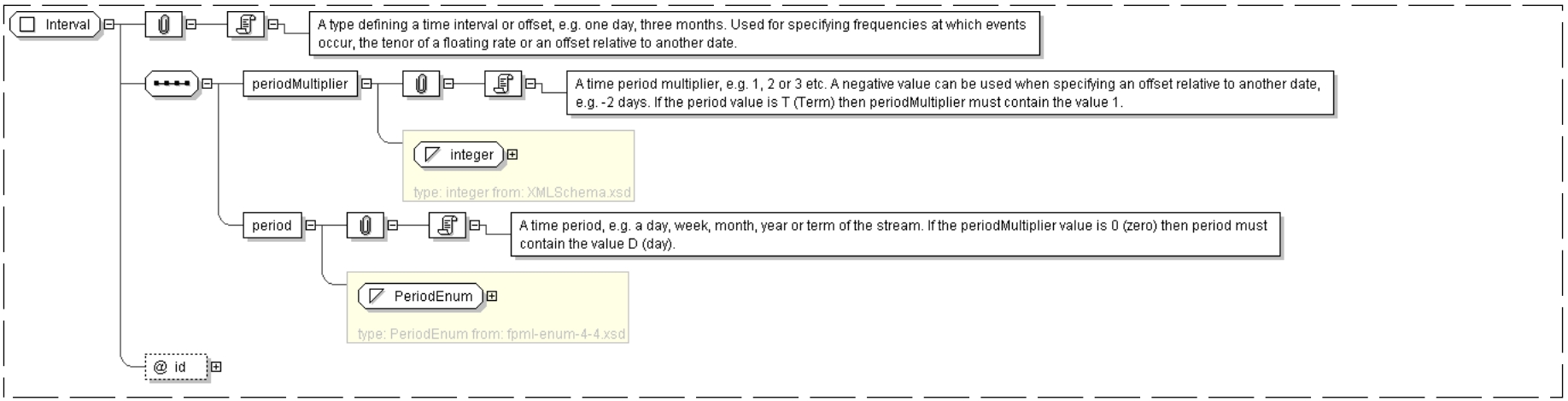
Sub-types:	• CalculationPeriodFrequency (by extension)
	• Offset (by extension)
	  ○ DateOffset (by extension)
	  ○ RelativeDateOffset (by extension)
	  ▪ AdjustedRelativeDateOffset (by extension)
	  ▪ RelativeDates (by extension)
	• ResetFrequency (by extension)

Name	Interval
Used by (from the same schema document)	Complex Type **ForecastRateIndex** , Model Group **FloatingRateIndex.model**
Abstract	no
Documentation	A type defining a time interval or offset, e.g. one day, three months. Used for specifying frequencies at which events occur, the tenor of a floating rate or an offset relative to another date.

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
   <periodMultiplier> xsd:integer </periodMultiplier> [1]
   'A time period multiplier, e.g. 1, 2 or 3 etc. A negative value can be used when specifying
   an offset relative to another date, e.g. -2 days. If the period value is T (Term)
   then periodMultiplier must contain the value 1.'

   <period> PeriodEnum </period> [1]
   'A time period, e.g. a day, week, month, year or term of the stream. If the
   periodMultiplier value is 0 (zero) then period must contain the value D (day).'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Interval">
   <xsd:sequence>
      <xsd:element name="periodMultiplier" type=" xsd:integer "/>
      <xsd:element name="period" type=" PeriodEnum "/>
   </xsd:sequence>
   <xsd:attribute name="id" type=" xsd:ID "/>
</xsd:complexType>
```

top

Complex Type: Leg

Super-types:	None
Sub-types:	None

Name	Leg
Abstract	yes
Documentation	A supertype of leg. All swap legs extend this type.

XML Instance Representation

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

Diagram

Schema Component Representation

```
<xsd:complexType name="Leg" abstract="true"/>
```

top

Complex Type: LegalEntity

Super-types:	None
Sub-types:	None

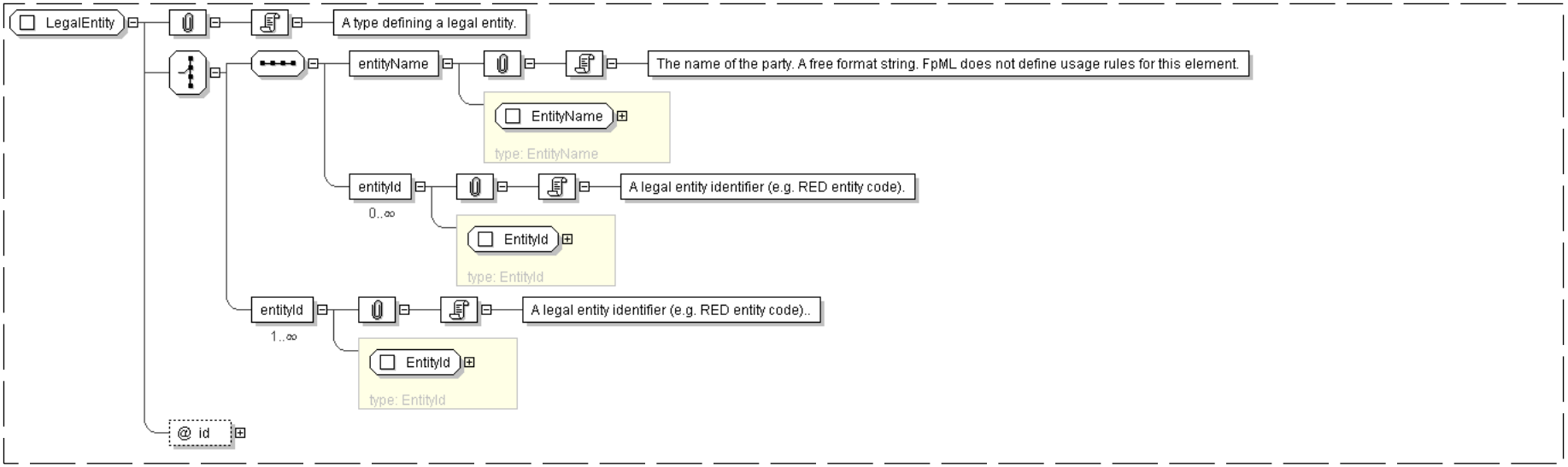
Name	LegalEntity

Abstract	no
Documentation	A type defining a legal entity.

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
Start Choice [1]
      <entityName> EntityName </entityName> [1]
      'The name of the party. A free format string. FpML does not define usage rules for
      this element.'

      <entityId> EntityId </entityId> [0..*]
      'A legal entity identifier (e.g. RED entity code).'

      <entityId> EntityId </entityId> [1..*]
      'A legal entity identifier (e.g. RED entity code)..'

End Choice
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="LegalEntity">
   <xsd:choice>
      <xsd:sequence>
         <xsd:element name="entityName" type=" EntityName "/>
         <xsd:element name="entityId" type=" EntityId " minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:element name="entityId" type=" EntityId " maxOccurs="unbounded"/>
   </xsd:choice>
   <xsd:attribute name="id" type=" xsd:ID "/>
</xsd:complexType>
```

top

Complex Type: LegalEntityReference

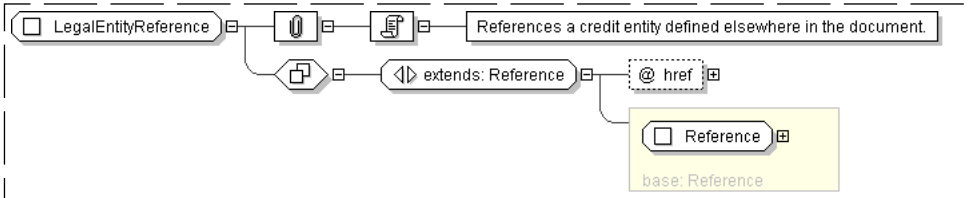
Super-types:	Reference < **LegalEntityReference** (by extension)
Sub-types:	None

Name	LegalEntityReference
Abstract	no
Documentation	References a credit entity defined elsewhere in the document.

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: MainPublication

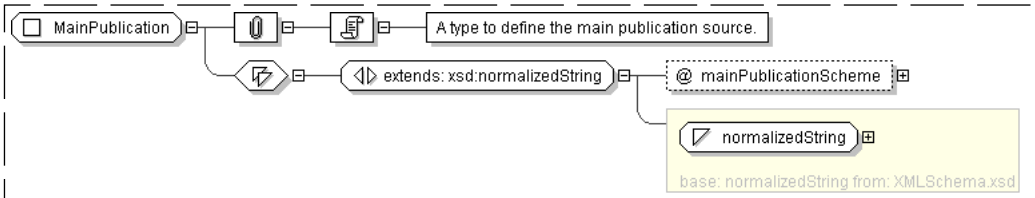
Super-types:	xsd:normalizedString < **MainPublication** (by extension)
Sub-types:	None

Name	MainPublication
Abstract	no
Documentation	A type to define the main publication source.

XML Instance Representation

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

Diagram



Schema Component Representation

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

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

top

Complex Type: ManualExercise

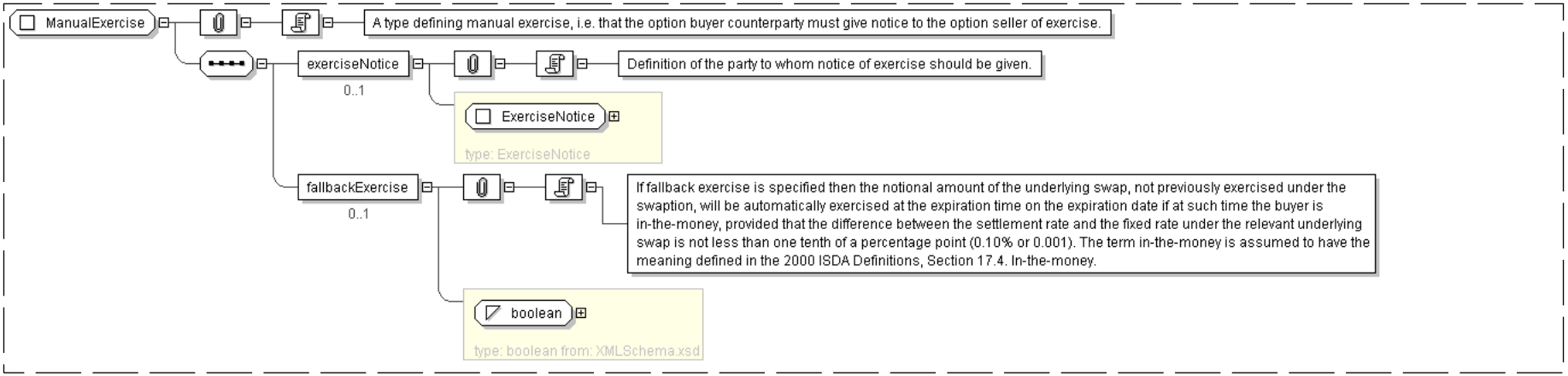
Super-types:	None
Sub-types:	None

Name	ManualExercise
Used by (from the same schema document)	Complex Type **ExerciseProcedure**
Abstract	no
Documentation	A type defining manual exercise, i.e. that the option buyer counterparty must give notice to the option seller of exercise.

XML Instance Representation

```
<...>
   <exerciseNotice> ExerciseNotice </exerciseNotice> [0..1]
   'Definition of the party to whom notice of exercise should be given.'

   <fallbackExercise> xsd:boolean </fallbackExercise> [0..1]
   'If fallback exercise is specified then the notional amount of the underlying swap,
   not previously exercised under the swaption, will be automatically exercised at the
   expiration time on the expiration date if at such time the buyer is in-the-money, provided
   that the difference between the settlement rate and the fixed rate under the
   relevant underlying swap is not less than one tenth of a percentage point (0.10% or 0.001).
   The term in-the-money is assumed to have the meaning defined in the 2000 ISDA
   Definitions, Section 17.4. In-the-money.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="ManualExercise">
   <xsd:sequence>
      <xsd:element name="exerciseNotice" type=" ExerciseNotice " minOccurs="0"/>
      <xsd:element name="fallbackExercise" type=" xsd:boolean " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: MasterAgreement

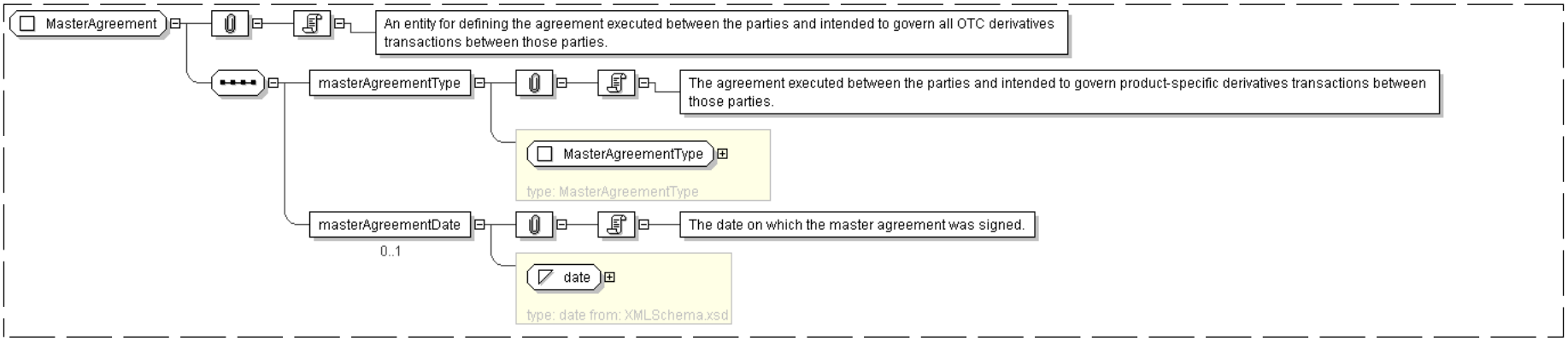
Super-types:	None
Sub-types:	None

Name	MasterAgreement
Used by (from the same schema document)	Complex Type **Documentation**
Abstract	no
Documentation	An entity for defining the agreement executed between the parties and intended to govern all OTC derivatives transactions between those parties.

XML Instance Representation

```
<...>
   <masterAgreementType> MasterAgreementType </masterAgreementType> [1]
   'The agreement executed between the parties and intended to govern product-specific
   derivatives transactions between those parties.'

   <masterAgreementDate> xsd:date </masterAgreementDate> [0..1]
   'The date on which the master agreement was signed.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="MasterAgreement">
   <xsd:sequence>
      <xsd:element name="masterAgreementType" type=" MasterAgreementType "/>
      <xsd:element name="masterAgreementDate" type=" xsd:date " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: MasterAgreementType

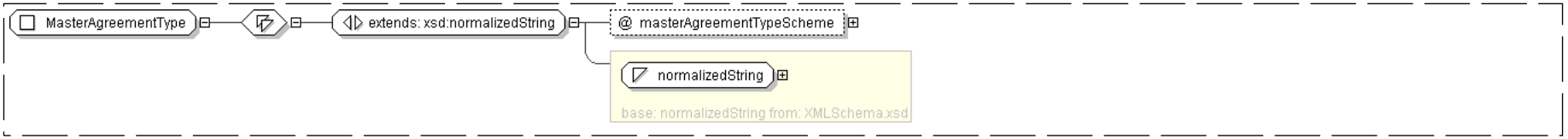
Super-types:	xsd:normalizedString < **MasterAgreementType** (by extension)
Sub-types:	None

Name	MasterAgreementType
Used by (from the same schema document)	Complex Type **MasterAgreement**
Abstract	no

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: MasterConfirmation

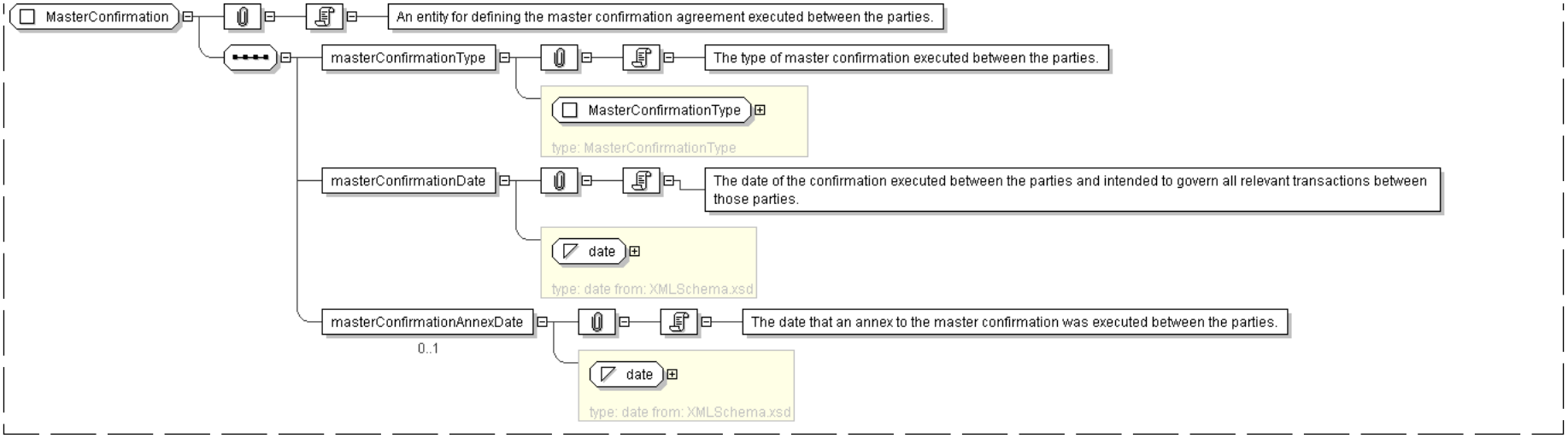
Super-types:	None
Sub-types:	None

Name	MasterConfirmation
Used by (from the same schema document)	Complex Type **Documentation**
Abstract	no
Documentation	An entity for defining the master confirmation agreement executed between the parties.

XML Instance Representation

```
<...>
   <masterConfirmationType> MasterConfirmationType </masterConfirmationType> [1]
   'The type of master confirmation executed between the parties.'

   <masterConfirmationDate> xsd:date </masterConfirmationDate> [1]
   'The date of the confirmation executed between the parties and intended to govern all
   relevant transactions between those parties.'

   <masterConfirmationAnnexDate> xsd:date </masterConfirmationAnnexDate> [0..1]
   'The date that an annex to the master confirmation was executed between the parties.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="MasterConfirmation">
   <xsd:sequence>
      <xsd:element name="masterConfirmationType" type=" MasterConfirmationType "/>
      <xsd:element name="masterConfirmationDate" type=" xsd:date "/>
      <xsd:element name="masterConfirmationAnnexDate" type=" xsd:date " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: MasterConfirmationType

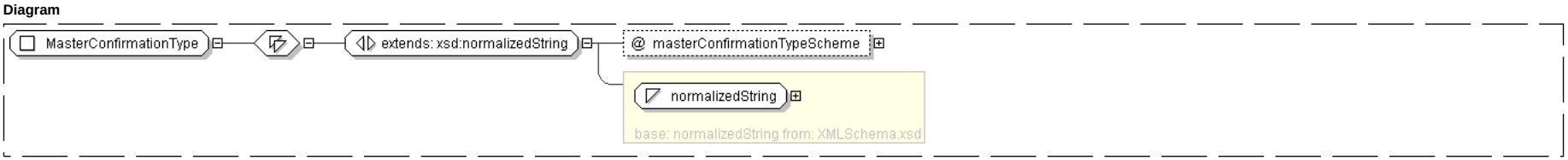
Super-types:	xsd:normalizedString < **MasterConfirmationType** (by extension)
Sub-types:	None

Name	MasterConfirmationType
Used by (from the same schema document)	Complex Type **MasterConfirmation**
Abstract	no

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="MasterConfirmationType">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="masterConfirmationTypeScheme" type=" xsd:anyURI " default="http://www.
         fpml.org/coding-scheme/master-confirmation-type-5-5"/>
```

```
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: Math

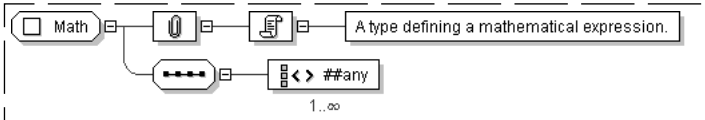
Super-types:	None
Sub-types:	None

Name	Math
Used by (from the same schema document)	Complex Type **Formula**
Abstract	no
Documentation	A type defining a mathematical expression.

XML Instance Representation

```
<...>
<!-- Mixed content -->
   Allow any elements from any namespace (skip validation). [1..*]
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Math" mixed="true">
   <xsd:sequence>
      <xsd:any namespace="##any" processContents="skip" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: MatrixTerm

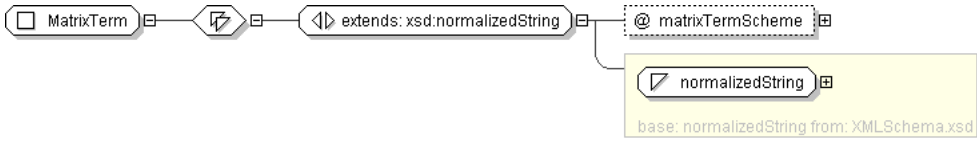
Super-types:	xsd:normalizedString < **MatrixTerm** (by extension)
Sub-types:	None

Name	MatrixTerm
Used by (from the same schema document)	Complex Type **ContractualMatrix**
Abstract	no

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="MatrixTerm">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="matrixTermScheme" type=" xsd:anyURI " default="http://www.fpml.org/
         coding-scheme/credit-matrix-transaction-type-2-2"/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: MatrixType

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

Name	MatrixType
Used by (from the same schema document)	Complex Type **ContractualMatrix**
Abstract	no

XML Instance Representation

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

Diagram

Schema Component Representation

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

top

Complex Type: MimeType

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

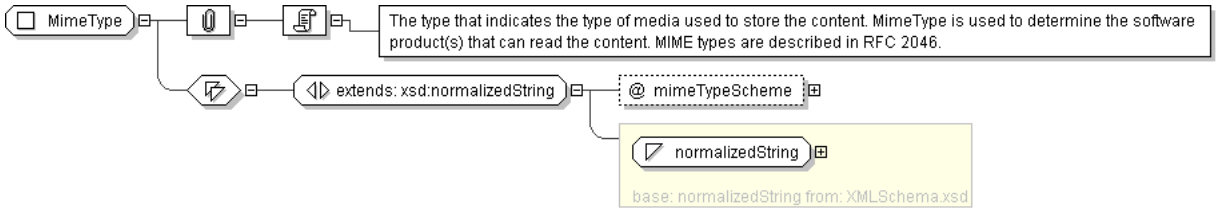
Name	MimeType
Abstract	no
Documentation	The type that indicates the type of media used to store the content. MimeType is used to determine the software product(s) that can read the content. MIME types are described in RFC 2046.

XML Instance Representation

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

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

Diagram



Schema Component Representation

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

top

Complex Type: Money

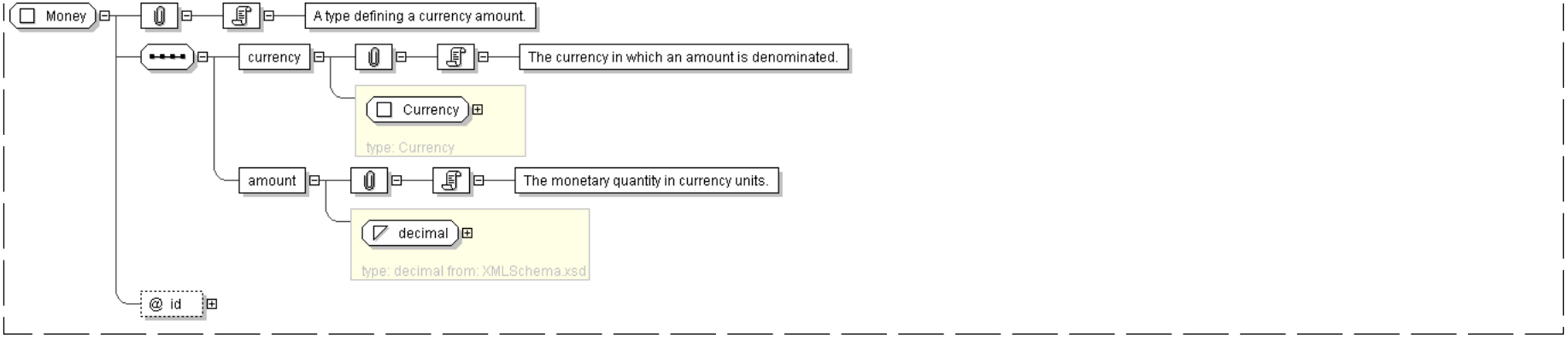
Super-types:	None
Sub-types:	None

Name	Money
Used by (from the same schema document)	Complex Type **Payment** , Complex Type **Payment** , Complex Type **SimplePayment** , Complex Type **SplitSettlement** , Complex Type **StubValue** , Model Group **PaymentDiscounting.model** , Model Group **Premium.model** , Model Group **SettlementAmountOrCurrency.model**
Abstract	no
Documentation	A type defining a currency amount.

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
   <currency> Currency </currency> [1]
   'The currency in which an amount is denominated.'

   <amount> xsd:decimal </amount> [1]
   'The monetary quantity in currency units.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Money">
   <xsd:sequence>
      <xsd:element name="currency" type=" Currency "/>
      <xsd:element name="amount" type=" xsd:decimal "/>
   </xsd:sequence>
   <xsd:attribute name="id" type=" xsd:ID "/>
</xsd:complexType>
```

top

Complex Type: MultipleExercise

Super-types:	None
Sub-types:	None

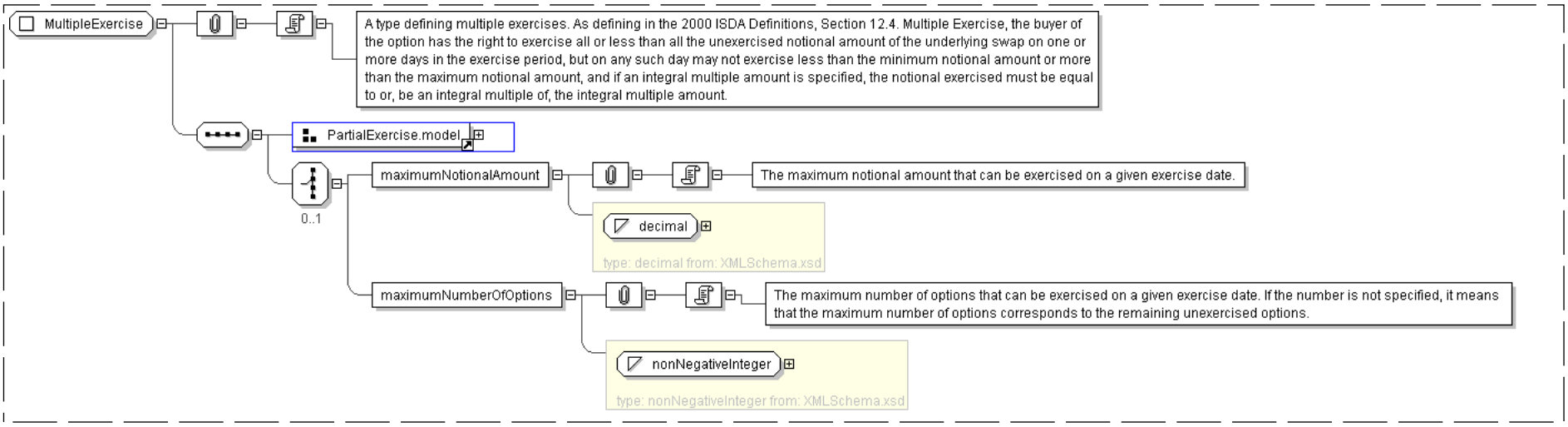
Name	MultipleExercise
Used by (from the same schema document)	Complex Type **AmericanExercise** , Complex Type **BermudaExercise**
Abstract	no
Documentation	A type defining multiple exercises. As defining in the 2000 ISDA Definitions, Section 12.4. Multiple Exercise, the buyer of the option has the right to exercise all or less than all the unexercised notional amount of the underlying swap on one or more days in the exercise period, but on any such day may not exercise less than the minimum notional amount or more than the maximum notional amount, and if an integral multiple amount is specified, the notional exercised must be equal to or, be an integral multiple of, the integral multiple amount.

XML Instance Representation

```
<...>
   <notionalReference> ScheduleReference </notionalReference> [0..*]
   'A pointer style reference to the associated notional schedule defined elsewhere in
   the document. This element has been made optional as part of its integration in
   the OptionBaseExtended, because not required for the options on securities.'

   <integralMultipleAmount> xsd:decimal </integralMultipleAmount> [0..1]
   'A notional amount which restricts the amount of notional that can be exercised when
   partial exercise or multiple exercise is applicable. The integral multiple amount defines
   a lower limit of notional that can be exercised and also defines a unit multiple of
   notional that can be exercised, i.e. only integer multiples of this amount can be exercised.'

Start Choice [1]
      <minimumNotionalAmount> xsd:decimal </minimumNotionalAmount> [1]
      'The minimum notional amount that can be exercised on a given exercise date.
      See multipleExercise.'

      <minimumNumberOfOptions> xsd:nonNegativeInteger </minimumNumberOfOptions> [1]
      'The minimum number of options that can be exercised on a given exercise date.'

End Choice
Start Choice [0..1]
```

```
<maximumNotionalAmount> xsd:decimal </maximumNotionalAmount> [1]
'The maximum notional amount that can be exercised on a given exercise date.'

<maximumNumberOfOptions> xsd:nonNegativeInteger </maximumNumberOfOptions> [1]
'The maximum number of options that can be exercised on a given exercise date. If the number
is not specified, it means that the maximum number of options corresponds to the
remaining unexercised options.'

End Choice
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="MultipleExercise">
   <xsd:sequence>
      <xsd:group ref=" PartialExercise.model "/>
      <xsd:choice minOccurs="0">
         <xsd:element name="maximumNotionalAmount" type=" xsd:decimal "/>
         <xsd:element name="maximumNumberOfOptions" type=" xsd:nonNegativeInteger "/>
      </xsd:choice>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: NotionalAmountReference

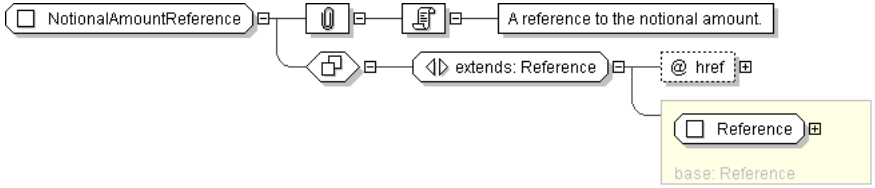
Super-types:	Reference < **NotionalAmountReference** (by extension)
Sub-types:	None

Name	NotionalAmountReference
Abstract	no
Documentation	A reference to the notional amount.

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: Offset

Super-types:	Interval < **Offset** (by extension)

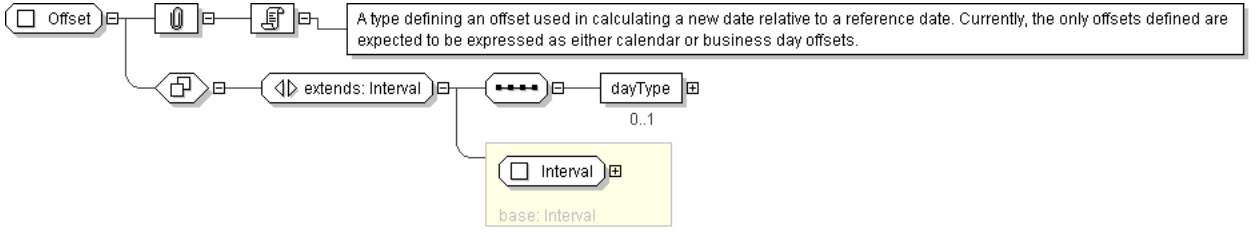
Sub-types:	• DateOffset (by extension)
	• RelativeDateOffset (by extension)
	  ○ AdjustedRelativeDateOffset (by extension)
	  ○ RelativeDates (by extension)

Name	Offset
Used by (from the same schema document)	Complex Type **DividendPaymentDate**
Abstract	no
Documentation	A type defining an offset used in calculating a new date relative to a reference date. Currently, the only offsets defined are expected to be expressed as either calendar or business day offsets.

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
   <periodMultiplier> xsd:integer </periodMultiplier> [1]
   'A time period multiplier, e.g. 1, 2 or 3 etc. A negative value can be used when specifying
   an offset relative to another date, e.g. -2 days. If the period value is T (Term)
   then periodMultiplier must contain the value 1.'

   <period> PeriodEnum </period> [1]
   'A time period, e.g. a day, week, month, year or term of the stream. If the
   periodMultiplier value is 0 (zero) then period must contain the value D (day).'

   <dayType> DayTypeEnum </dayType> [0..1]
   'In the case of an offset specified as a number of days, this element defines
   whether consideration is given as to whether a day is a good business day or not. If a day
   type of business days is specified then non-business days are ignored when calculating
   the offset. The financial business centers to use for determination of business days
   are implied by the context in which this element is used. This element must only be
   included when the offset is specified as a number of days. If the offset is zero days then
   the dayType element should not be included.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Offset">
   <xsd:complexContent>
      <xsd:extension base=" Interval ">
         <xsd:sequence>
            <xsd:element name="dayType" type=" DayTypeEnum " minOccurs="0"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: PartialExercise

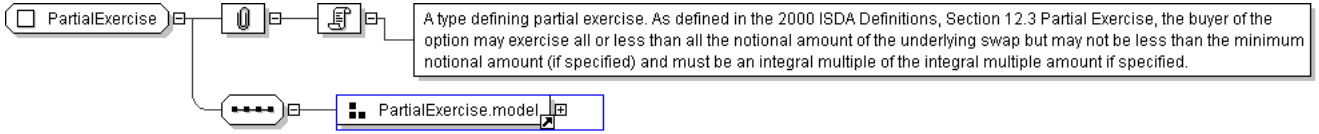
Super-types:	None
Sub-types:	None

Name	PartialExercise
Used by (from the same schema document)	Complex Type **EuropeanExercise**
Abstract	no
Documentation	A type defining partial exercise. As defined in the 2000 ISDA Definitions, Section 12.3 Partial Exercise, the buyer of the option may exercise all or less than all the notional amount of the underlying swap but may not be less than the minimum notional amount (if specified) and must be an integral multiple of the integral multiple amount if specified.

XML Instance Representation

```
<...>
   <notionalReference> ScheduleReference </notionalReference> [0..*]
   'A pointer style reference to the associated notional schedule defined elsewhere in
   the document. This element has been made optional as part of its integration in
   the OptionBaseExtended, because not required for the options on securities.'

   <integralMultipleAmount> xsd:decimal </integralMultipleAmount> [0..1]
   'A notional amount which restricts the amount of notional that can be exercised when
   partial exercise or multiple exercise is applicable. The integral multiple amount defines
   a lower limit of notional that can be exercised and also defines a unit multiple of
   notional that can be exercised, i.e. only integer multiples of this amount can be exercised.'

Start Choice [1]
      <minimumNotionalAmount> xsd:decimal </minimumNotionalAmount> [1]
      'The minimum notional amount that can be exercised on a given exercise date.
      See multipleExercise.'

      <minimumNumberOfOptions> xsd:nonNegativeInteger </minimumNumberOfOptions> [1]
      'The minimum number of options that can be exercised on a given exercise date.'

End Choice
</...>
```

Diagram



Schema Component Representation

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

top

Complex Type: Party

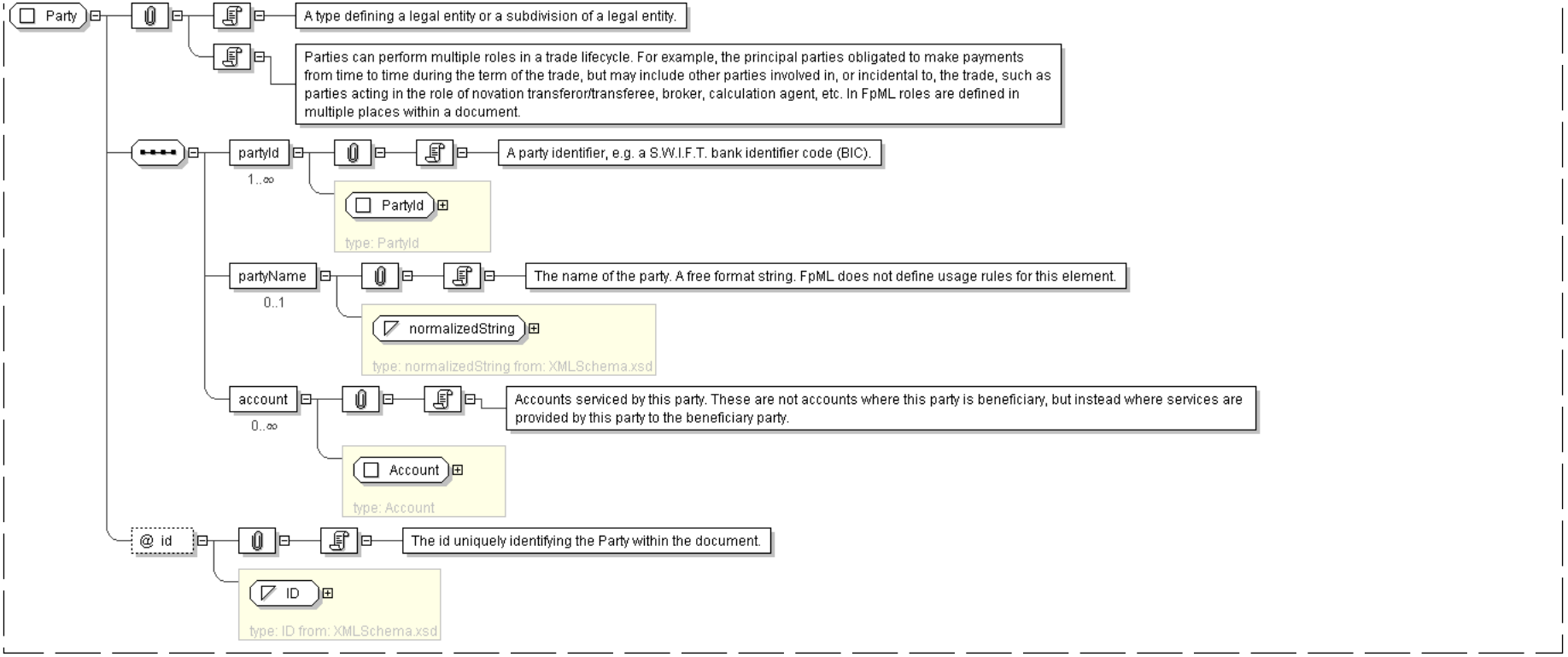
Super-types:	None
Sub-types:	None

Name	Party
Abstract	no
Documentation	A type defining a legal entity or a subdivision of a legal entity. Parties can perform multiple roles in a trade lifecycle. For example, the principal parties obligated to make payments from time to time during the term of the trade, but may include other parties involved in, or incidental to, the trade, such as parties acting in the role of novation transferor/transferee, broker, calculation agent, etc. In FpML roles are defined in multiple places within a document.

XML Instance Representation

```
<...
id=" xsd:ID [1]
 'The id uniquely identifying the Party within the document.'

">
   <partyId> PartyId </partyId> [1..*]
   'A party identifier, e.g. a S.W.I.F.T. bank identifier code (BIC).'

   <partyName> xsd:normalizedString </partyName> [0..1]
   'The name of the party. A free format string. FpML does not define usage rules for
   this element.'

   <account> Account </account> [0..*]
   'Accounts serviced by this party. These are not accounts where this party is beneficiary,
   but instead where services are provided by this party to the beneficiary party.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Party">
   <xsd:sequence>
      <xsd:element name="partyId" type=" PartyId " maxOccurs="unbounded"/>
      <xsd:element name="partyName" type=" xsd:normalizedString " minOccurs="0"/>
      <xsd:element name="account" type=" Account " minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
   <xsd:attribute name="id" type=" xsd:ID " use="required"/>
</xsd:complexType>
```

top

Complex Type: PartyId

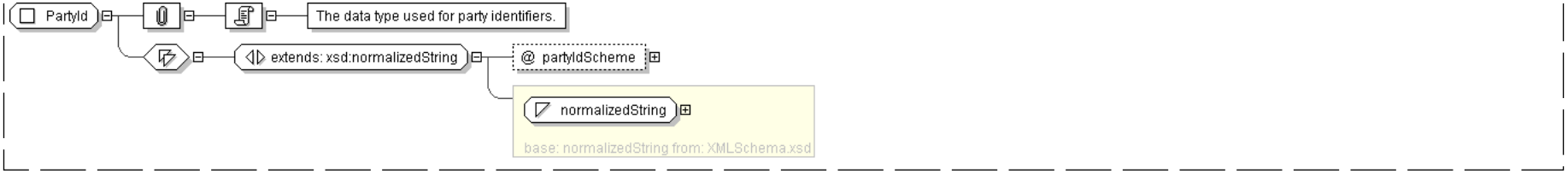
Super-types:	xsd:normalizedString < **PartyId** (by extension)
Sub-types:	None

Name	PartyId
Used by (from the same schema document)	Complex Type **Party**
Abstract	no
Documentation	The data type used for party identifiers.

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: PartyOrAccountReference

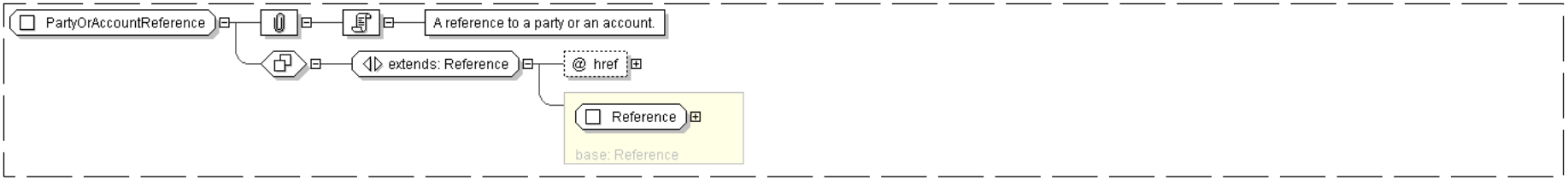
Super-types:	Reference < **PartyOrAccountReference** (by extension)
Sub-types:	None

Name	PartyOrAccountReference
Used by (from the same schema document)	Model Group **PayerReceiver.model** , Model Group **PayerReceiver.model**
Abstract	no
Documentation	A reference to a party or an account.

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: PartyOrTradeSideReference

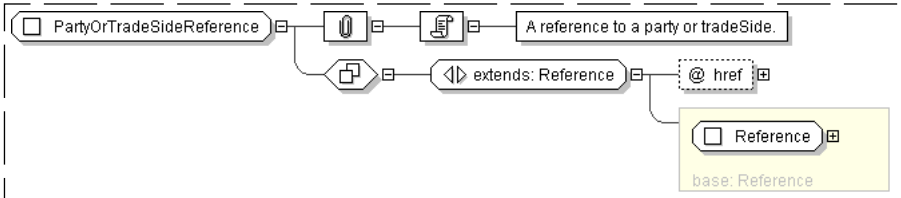
Super-types:	Reference < **PartyOrTradeSideReference** (by extension)
Sub-types:	None

Name	PartyOrTradeSideReference
Used by (from the same schema document)	Model Group **BuyerSeller.model** , Model Group **BuyerSeller.model**
Abstract	no
Documentation	A reference to a party or tradeSide.

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: PartyReference

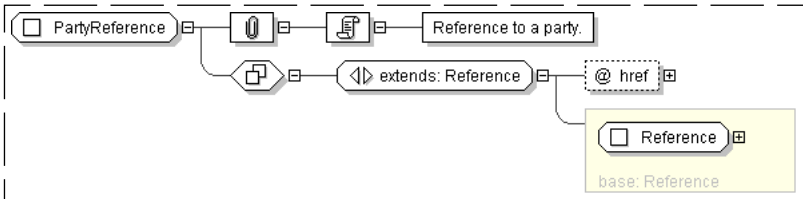
Super-types:	Reference < **PartyReference** (by extension)
Sub-types:	None

Name	PartyReference
Used by (from the same schema document)	Complex Type **Account** , Complex Type **Beneficiary** , Complex Type **CalculationAgent** , Complex Type **CorrespondentInformation** , Complex Type **DividendConditions** , Complex Type **ExerciseNotice** , Complex Type **ExerciseNotice** , Complex Type **IntermediaryInformation** , Complex Type **SettlementInstruction**
Abstract	no
Documentation	Reference to a party.

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="PartyReference">
   <xsd:complexContent>
      <xsd:extension base=" Reference ">
```

```
        <xsd:attribute name="href" type=" xsd:IDREF " use="required" reference="Party"/>
      </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: Payment

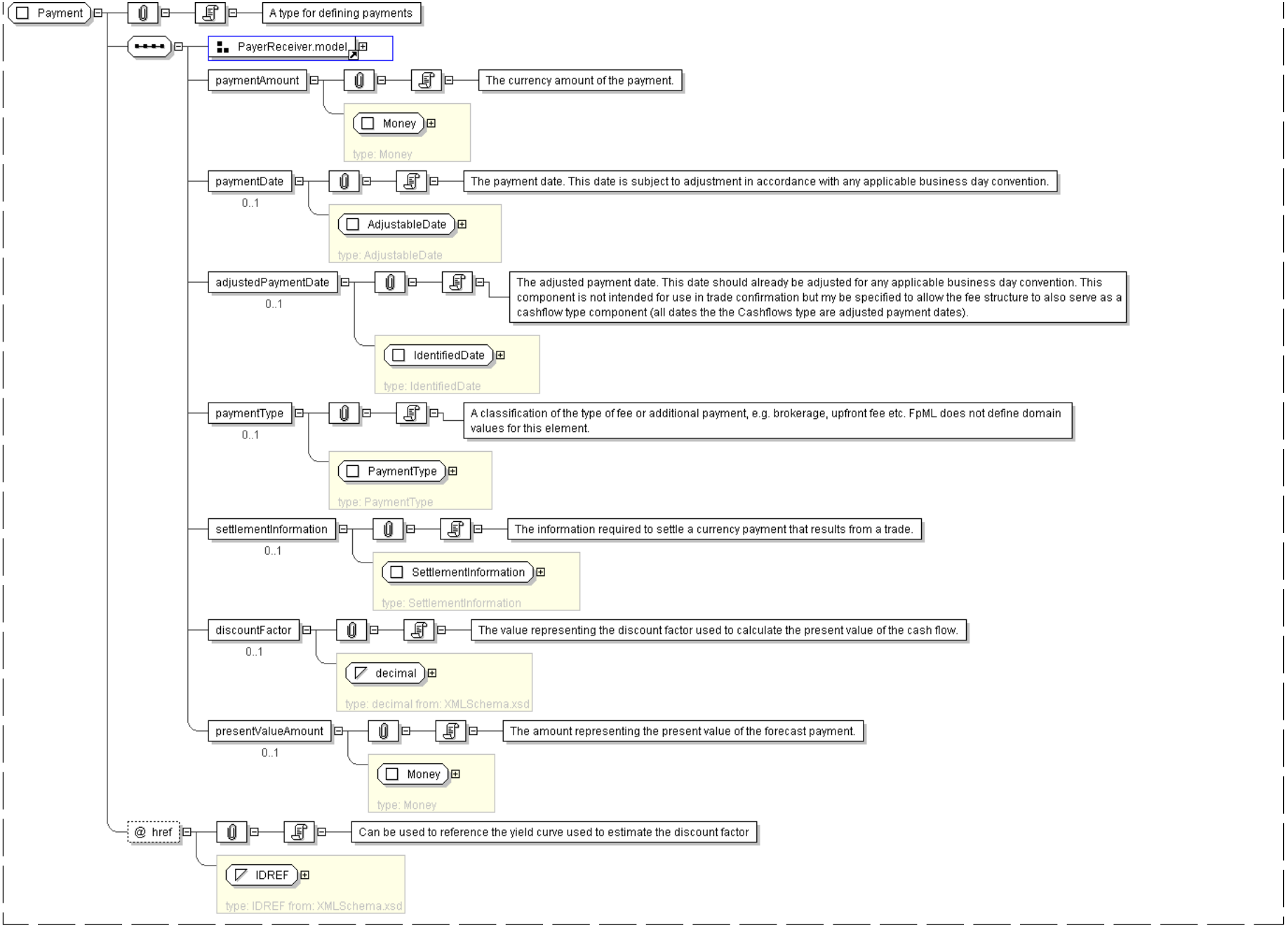
Super-types:	None
Sub-types:	None

Name	Payment
Abstract	no
Documentation	A type for defining payments

XML Instance Representation

```
<...
href=" xsd:IDREF [0..1]
 'Can be used to reference the yield curve used to estimate the discount factor'

">
   <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]
    'The currency amount of the payment.'

   <paymentDate> AdjustableDate </paymentDate> [0..1]
    'The payment date. This date is subject to adjustment in accordance with any
    applicable business day convention.'

   <adjustedPaymentDate> IdentifiedDate </adjustedPaymentDate> [0..1]
    'The adjusted payment date. This date should already be adjusted for any applicable
    business day convention. This component is not intended for use in trade confirmation but my
    be specified to allow the fee structure to also serve as a cashflow type component (all
    dates the the Cashflows type are adjusted payment dates).'

   <paymentType> PaymentType </paymentType> [0..1]
    'A classification of the type of fee or additional payment, e.g. brokerage, upfront fee
    etc. FpML does not define domain values for this element.'

   <settlementInformation> SettlementInformation </settlementInformation> [0..1]
    'The information required to settle a currency payment that results from a trade.'

   <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.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Payment">
   <xsd:sequence>
      <xsd:group ref=" PayerReceiver.model "/>
      <xsd:element name="paymentAmount" type=" Money "/>
      <xsd:element name="paymentDate" type=" AdjustableDate " minOccurs="0"/>
      <xsd:element name="adjustedPaymentDate" type=" IdentifiedDate " minOccurs="0"/>
      <xsd:element name="paymentType" type=" PaymentType " minOccurs="0"/>
```

```
      <xsd:element name="settlementInformation" type=" SettlementInformation " minOccurs="0"/>
      <xsd:element name="discountFactor" type=" xsd:decimal " minOccurs="0"/>
      <xsd:element name="presentValueAmount" type=" Money " minOccurs="0"/>
   </xsd:sequence>
   <xsd:attribute name="href" type=" xsd:IDREF " reference="PricingStructure"/>
</xsd:complexType>
```

top

Complex Type: PaymentCurrency

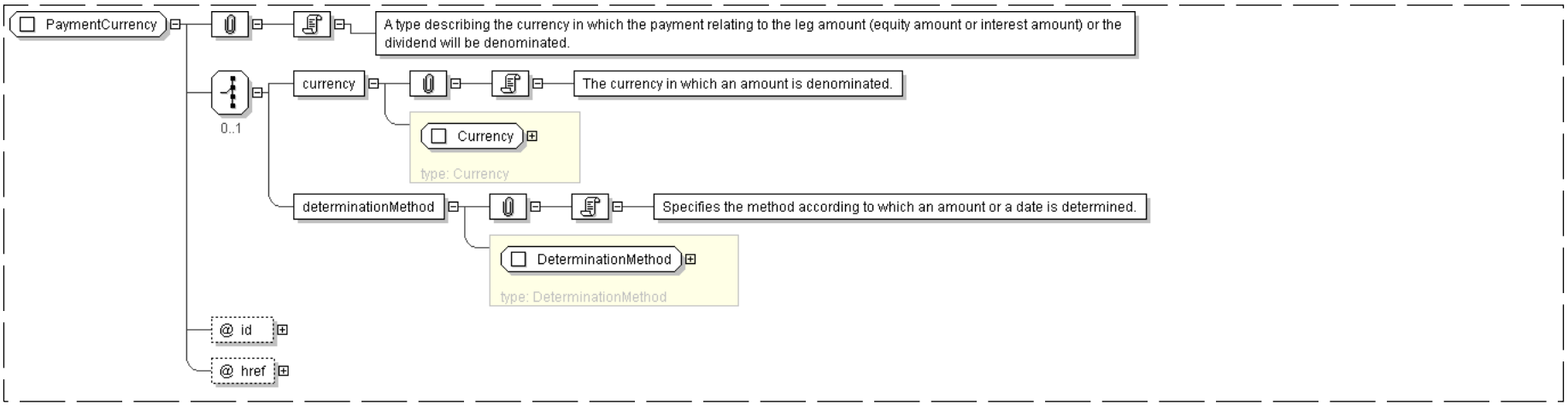
Super-types:	None
Sub-types:	None

Name	PaymentCurrency
Used by (from the same schema document)	Complex Type **DividendConditions**
Abstract	no
Documentation	A type describing the currency in which the payment relating to the leg amount (equity amount or interest amount) or the dividend will be denominated.

XML Instance Representation

```
<...
id=" xsd:ID [0..1]"
href=" xsd:IDREF [0..1]">
Start Choice [0..1]
      <currency> Currency </currency> [1]
      'The currency in which an amount is denominated.'

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

End Choice
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="PaymentCurrency" deprecated="true" deprecatedReason="The model is
wrong since it has an intradocument reference that is not clear. Current PaymentCurrency
model and elements using this type are deprecated. Instead, a choice between
currency, determinationMethod, and currencyReference (of type CurrencyReference) are
inserted at the same level as the elements using PaymentCurrency.">
   <xsd:choice minOccurs="0">
      <xsd:element name="currency" type=" Currency "/>
      <xsd:element name="determinationMethod" type=" DeterminationMethod "/>
```

```
    </xsd:choice>
    <xsd:attribute name="id" type=" xsd:ID "/>
    <xsd:attribute name="href" type=" xsd:IDREF "/>
</xsd:complexType>
```

top

Complex Type: PaymentType

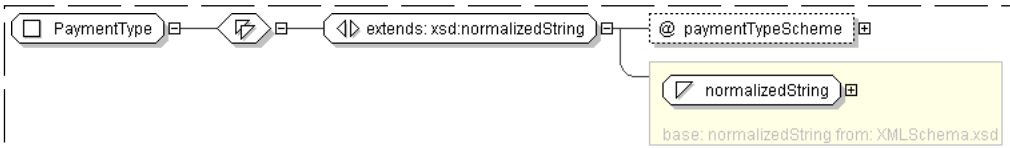
Super-types:	xsd:normalizedString < **PaymentType** (by extension)
Sub-types:	None

Name	PaymentType
Used by (from the same schema document)	Complex Type **Payment**
Abstract	no

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: PeriodicDates

Super-types:	None
Sub-types:	None

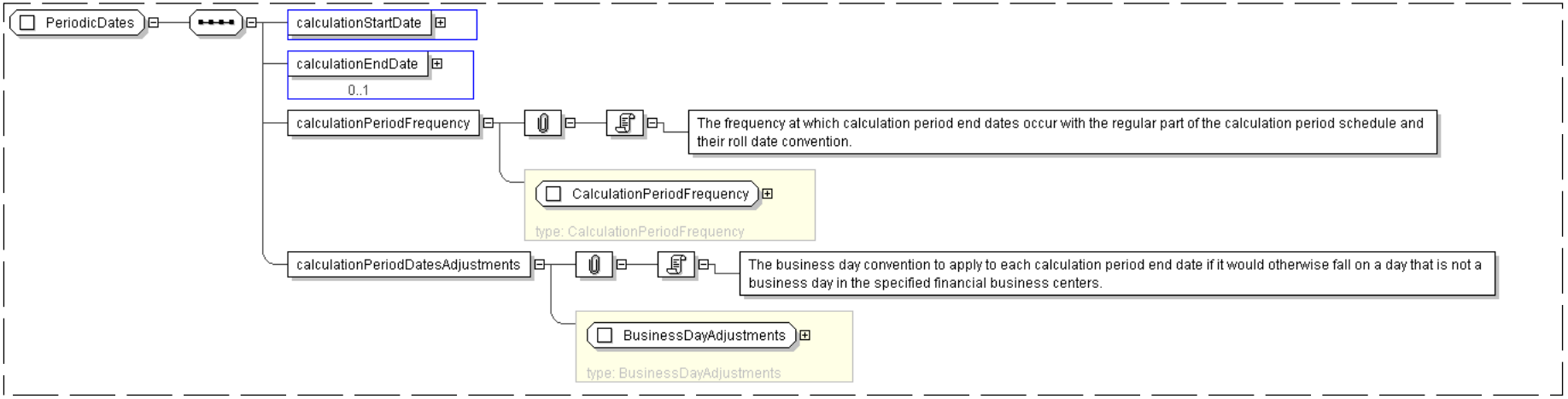
Name	PeriodicDates
Used by (from the same schema document)	Complex Type **AdjustableRelativeOrPeriodicDates**
Abstract	no

XML Instance Representation

```
<...>
   <calculationStartDate> AdjustableOrRelativeDate </calculationStartDate> [1]
   <calculationEndDate> AdjustableOrRelativeDate </calculationEndDate> [0..1]
   <calculationPeriodFrequency> CalculationPeriodFrequency </calculationPeriodFrequency> [1]
   'The frequency at which calculation period end dates occur with the regular part of
   the calculation period schedule and their roll date convention.'

   <calculationPeriodDatesAdjustments> BusinessDayAdjustments </
   calculationPeriodDatesAdjustments> [1]
   'The business day convention to apply to each calculation period end date if it would
```

```
    otherwise fall on a day that is not a business day in the specified financial business centers.'
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="PeriodicDates">
   <xsd:sequence>
      <xsd:element name="calculationStartDate" type=" AdjustableOrRelativeDate "/>
      <xsd:element name="calculationEndDate" type=" AdjustableOrRelativeDate " minOccurs="0"/>
      <xsd:element name="calculationPeriodFrequency" type=" CalculationPeriodFrequency "/>
      <xsd:element name="calculationPeriodDatesAdjustments" type=" BusinessDayAdjustments "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: PricingStructure

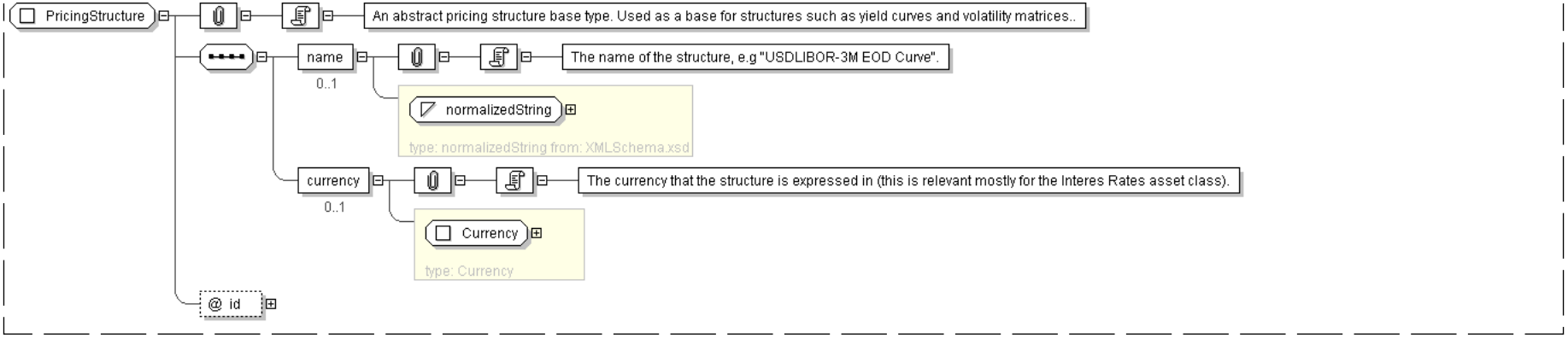
Super-types:	None
Sub-types:	None

Name	PricingStructure
Abstract	yes
Documentation	An abstract pricing structure base type. Used as a base for structures such as yield curves and volatility matrices..

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
   <name> xsd:normalizedString </name> [0..1]
   'The name of the structure, e.g \"USDLIBOR-3M EOD Curve\".'

   <currency> Currency </currency> [0..1]
   'The currency that the structure is expressed in (this is relevant mostly for the Interes
   Rates asset class).'
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="PricingStructure" abstract="true">
   <xsd:sequence>
      <xsd:element name="name" type=" xsd:normalizedString " minOccurs="0"/>
      <xsd:element name="currency" type=" Currency " minOccurs="0"/>
   </xsd:sequence>
   <xsd:attribute name="id" type=" xsd:ID "/>
</xsd:complexType>
```

top

Complex Type: PricingStructureReference

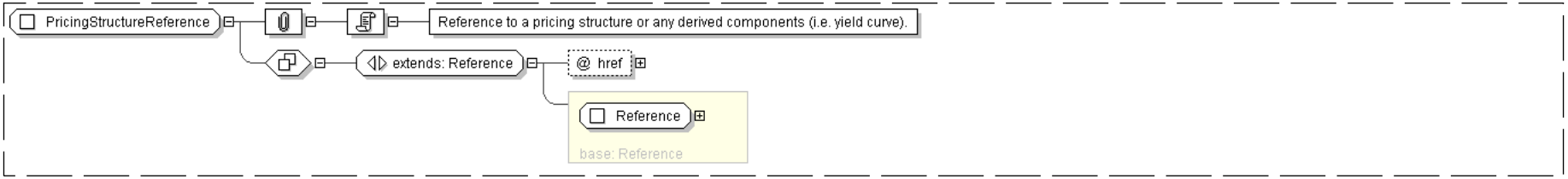
Super-types:	Reference < **PricingStructureReference** (by extension)
Sub-types:	None

Name	PricingStructureReference
Abstract	no
Documentation	Reference to a pricing structure or any derived components (i.e. yield curve).

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: PrincipalExchanges

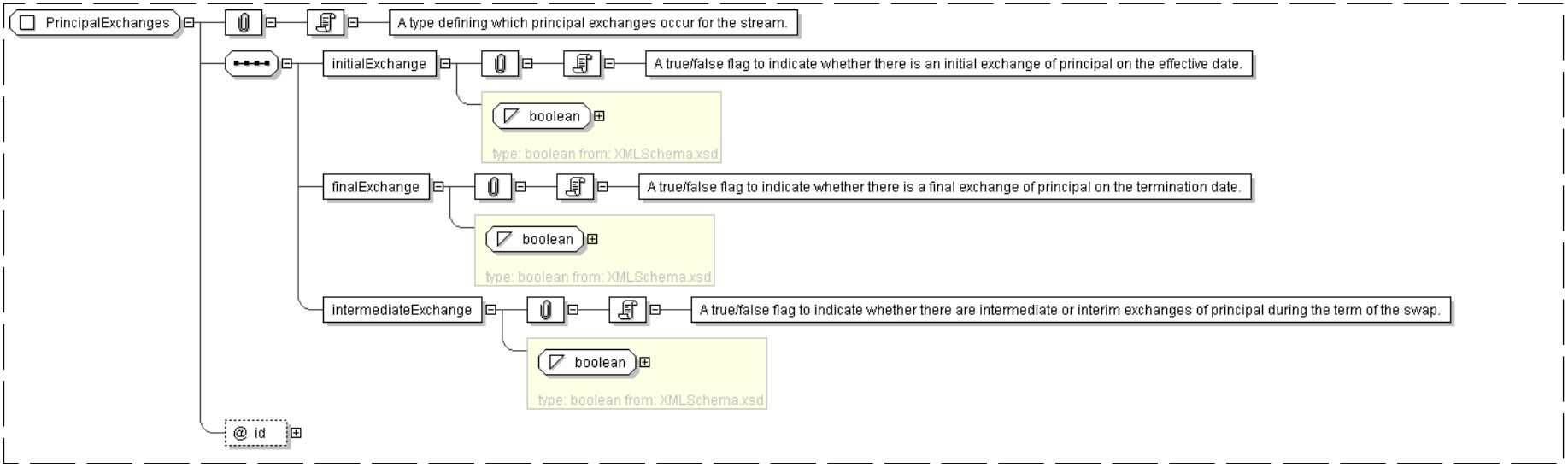
Super-types:	None
Sub-types:	None

Name	PrincipalExchanges
Abstract	no
Documentation	A type defining which principal exchanges occur for the stream.

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
   <initialExchange> xsd:boolean </initialExchange> [1]
   'A true/false flag to indicate whether there is an initial exchange of principal on
   the effective date.'

   <finalExchange> xsd:boolean </finalExchange> [1]
   'A true/false flag to indicate whether there is a final exchange of principal on
   the termination date.'

   <intermediateExchange> xsd:boolean </intermediateExchange> [1]
   'A true/false flag to indicate whether there are intermediate or interim exchanges of
   principal during the term of the swap.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="PrincipalExchanges">
   <xsd:sequence>
      <xsd:element name="initialExchange" type=" xsd:boolean "/>
      <xsd:element name="finalExchange" type=" xsd:boolean "/>
      <xsd:element name="intermediateExchange" type=" xsd:boolean "/>
   </xsd:sequence>
   <xsd:attribute name="id" type=" xsd:ID " use="optional"/>
</xsd:complexType>
```

top

Complex Type: Product

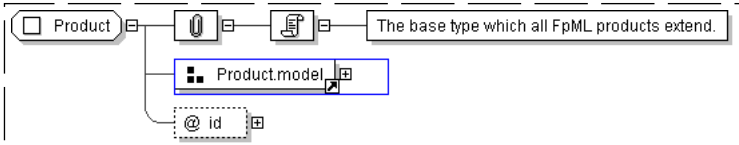
Super-types:	None
Sub-types:	None

Name	Product
Used by (from the same schema document)	Element **product**
Abstract	yes
Documentation	The base type which all FpML products extend.

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.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Product" abstract="true">
   <xsd:group ref=" Product.model "/>
   <xsd:attribute name="id" type=" xsd:ID "/>
</xsd:complexType>
```

top

Complex Type: ProductId

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

Name	ProductId
Used by (from the same schema document)	Model Group **Product.model**
Abstract	no

XML Instance Representation

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

Diagram

Schema Component Representation

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

top

Complex Type: ProductReference

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

Name	ProductReference
Abstract	no
Documentation	Reference to a full FpML product.

XML Instance Representation

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

Diagram

Schema Component Representation

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

top

Complex Type: ProductType

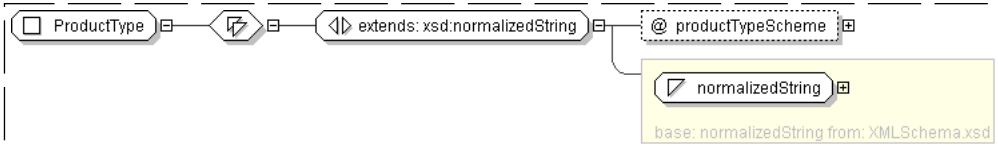
Super-types:	xsd:normalizedString < **ProductType** (by extension)
Sub-types:	None

Name	ProductType
Used by (from the same schema document)	Model Group **Product.model**
Abstract	no

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="ProductType">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:normalizedString ">
         <xsd:attribute name="productTypeScheme" type=" xsd:anyURI " default="http://www.fpml.org/
         coding-scheme/product-type-simple-1-2"/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: QuotedCurrencyPair

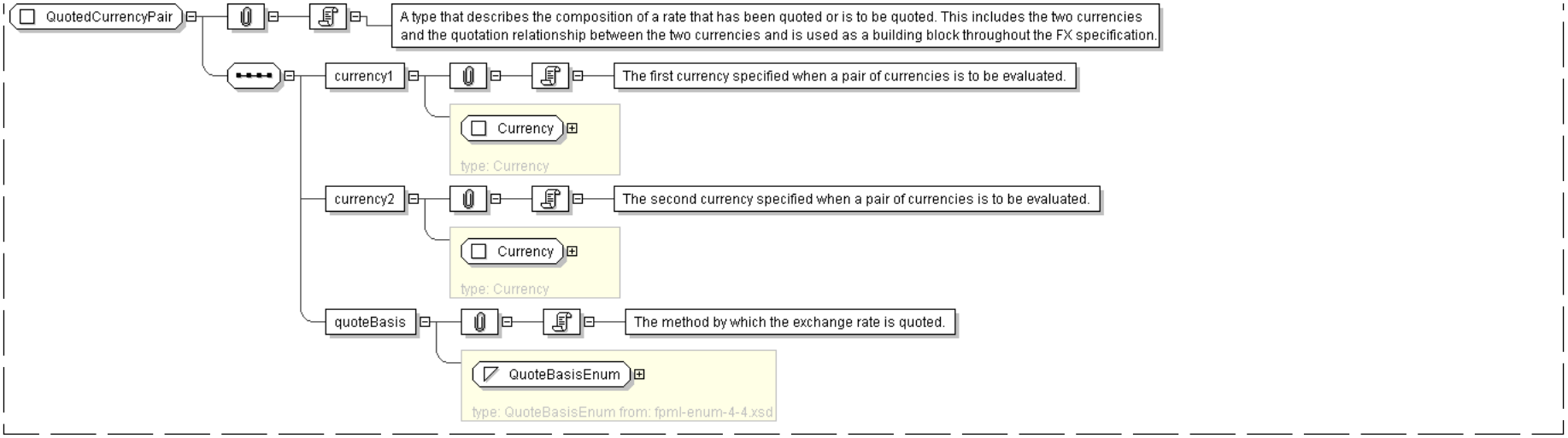
Super-types:	None
Sub-types:	None

Name	QuotedCurrencyPair
Used by (from the same schema document)	Complex Type **FxFixing** , Complex Type **FxRate**
Abstract	no
Documentation	A type that describes the composition of a rate that has been quoted or is to be quoted. This includes the two currencies and the quotation relationship between the two currencies and is used as a building block throughout the FX specification.

XML Instance Representation

```
<...>
   <currency1> Currency </currency1> [1]
   'The first currency specified when a pair of currencies is to be evaluated.'

   <currency2> Currency </currency2> [1]
   'The second currency specified when a pair of currencies is to be evaluated.'

   <quoteBasis> QuoteBasisEnum </quoteBasis> [1]
   'The method by which the exchange rate is quoted.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="QuotedCurrencyPair">
   <xsd:sequence>
      <xsd:element name="currency1" type=" Currency "/>
      <xsd:element name="currency2" type=" Currency "/>
      <xsd:element name="quoteBasis" type=" QuoteBasisEnum "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: Rate

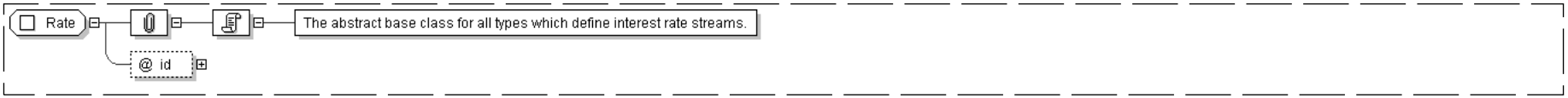
Super-types:	None
Sub-types:	• FloatingRate (by extension) ○ FloatingRateCalculation (by extension)

Name	Rate
Abstract	yes
Documentation	The abstract base class for all types which define interest rate streams.

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="Rate" abstract="true">
   <xsd:attribute name="id" type=" xsd:ID "/>
</xsd:complexType>
```

top

Complex Type: RateObservation

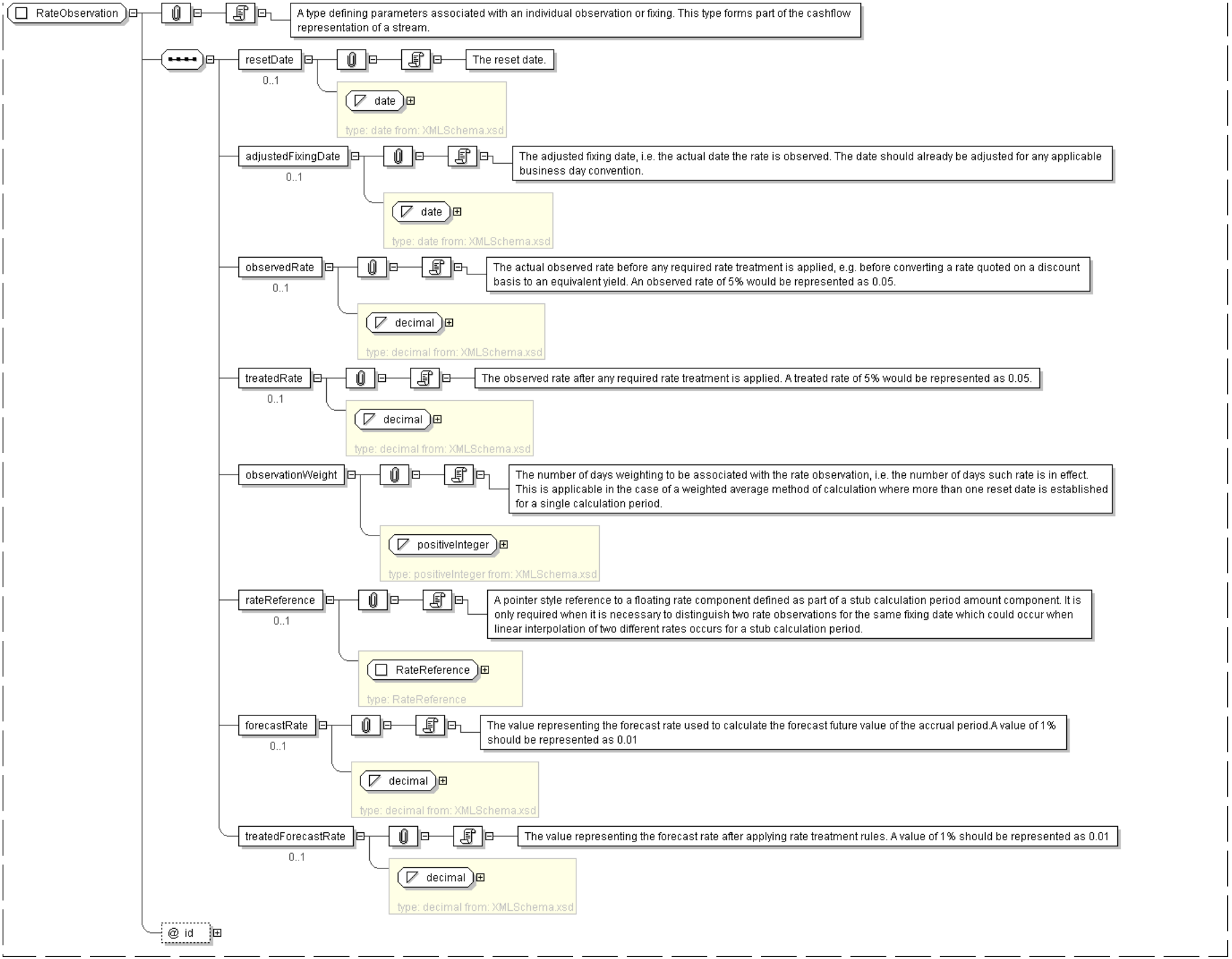
Super-types:	None
Sub-types:	None

Name	RateObservation
Abstract	no
Documentation	A type defining parameters associated with an individual observation or fixing. This type forms part of the cashflow representation of a stream.

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
   <resetDate> xsd:date </resetDate> [0..1]
   'The reset date.'

   <adjustedFixingDate> xsd:date </adjustedFixingDate> [0..1]
   'The adjusted fixing date, i.e. the actual date the rate is observed. The date should
   already be adjusted for any applicable business day convention.'

   <observedRate> xsd:decimal </observedRate> [0..1]
   'The actual observed rate before any required rate treatment is applied, e.g. before
   converting a rate quoted on a discount basis to an equivalent yield. An observed rate of
   5% would be represented as 0.05.'

   <treatedRate> xsd:decimal </treatedRate> [0..1]
   'The observed rate after any required rate treatment is applied. A treated rate of 5% would
   be represented as 0.05.'

   <observationWeight> xsd:positiveInteger </observationWeight> [1]
   'The number of days weighting to be associated with the rate observation, i.e. the number
   of days such rate is in effect. This is applicable in the case of a weighted average method
   of calculation where more than one reset date is established for a single calculation period.'

   <rateReference> RateReference </rateReference> [0..1]
   'A pointer style reference to a floating rate component defined as part of a stub
   calculation period amount component. It is only required when it is necessary to
   distinguish two rate observations for the same fixing date which could occur when
   linear interpolation of two different rates occurs for a stub calculation period.'

   <forecastRate> xsd:decimal </forecastRate> [0..1]
   'The value representing the forecast rate used to calculate the forecast future value of
   the accrual period.A value of 1% should be represented as 0.01'

   <treatedForecastRate> xsd:decimal </treatedForecastRate> [0..1]
   'The value representing the forecast rate after applying rate treatment rules. A value of
   1% should be represented as 0.01'

</...>
```

Diagram



Schema Component Representation

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

```
      <xsd:element name="resetDate" type=" xsd:date " minOccurs="0"/>
      <xsd:element name="adjustedFixingDate" type=" xsd:date " minOccurs="0"/>
      <xsd:element name="observedRate" type=" xsd:decimal " minOccurs="0"/>
      <xsd:element name="treatedRate" type=" xsd:decimal " minOccurs="0"/>
      <xsd:element name="observationWeight" type=" xsd:positiveInteger "/>
      <xsd:element name="rateReference" type=" RateReference " minOccurs="0"/>
      <xsd:element name="forecastRate" type=" xsd:decimal " minOccurs="0"/>
      <xsd:element name="treatedForecastRate" type=" xsd:decimal " minOccurs="0"/>
   </xsd:sequence>
   <xsd:attribute name="id" type=" xsd:ID "/>
</xsd:complexType>
```

top

Complex Type: RateReference

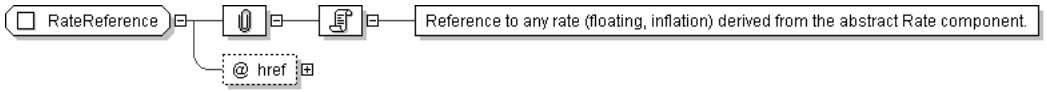
Super-types:	None
Sub-types:	None

Name	RateReference
Used by (from the same schema document)	Complex Type **RateObservation**
Abstract	no
Documentation	Reference to any rate (floating, inflation) derived from the abstract Rate component.

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="RateReference">
   <xsd:attribute name="href" type=" xsd:IDREF " use="required" reference="Rate"/>
</xsd:complexType>
```

top

Complex Type: RateSourcePage

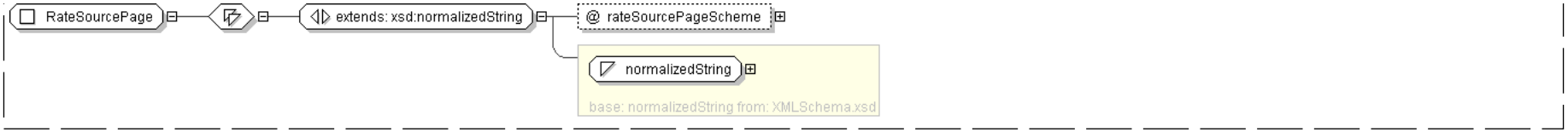
Super-types:	xsd:normalizedString < **RateSourcePage** (by extension)
Sub-types:	None

Name	RateSourcePage
Used by (from the same schema document)	Complex Type **InformationSource**
Abstract	no

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: Reference

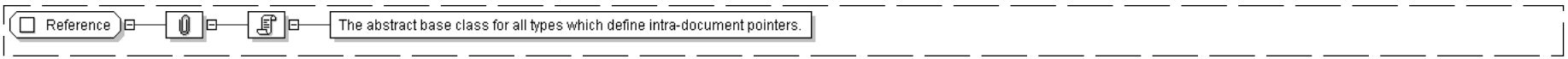
Super-types:	None
Sub-types:	• AccountReference (by extension) • AmountReference (by extension) • BusinessCentersReference (by extension) • BusinessDayAdjustmentsReference (by extension) • DateReference (by extension) • IdentifiedCurrencyReference (by extension) • LegalEntityReference (by extension) • NotionalAmountReference (by extension) • PartyOrAccountReference (by extension) • PartyOrTradeSideReference (by extension) • PartyReference (by extension) • PricingStructureReference (by extension) • ProductReference (by extension) • ScheduleReference (by extension) • SpreadScheduleReference (by extension)

Name	Reference
Abstract	yes
Documentation	The abstract base class for all types which define intra-document pointers.

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="Reference" abstract="true"/>
```

top

Complex Type: ReferenceAmount

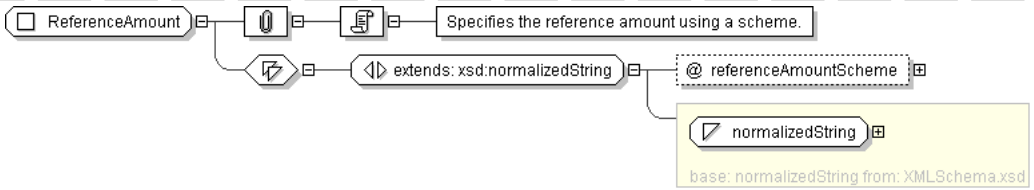
Super-types:	xsd:normalizedString < **ReferenceAmount** (by extension)
Sub-types:	None

Name	ReferenceAmount
Abstract	no
Documentation	Specifies the reference amount using a scheme.

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: ReferenceBank

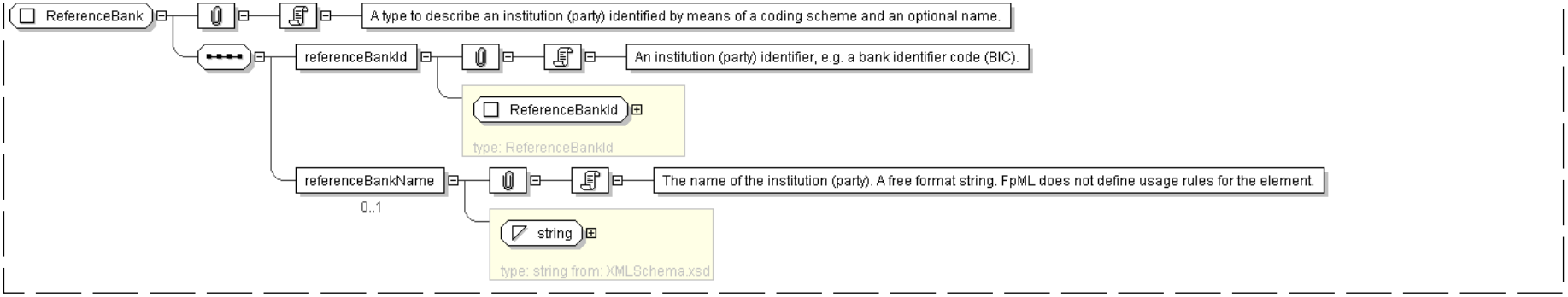
Super-types:	None
Sub-types:	None

Name	ReferenceBank
Used by (from the same schema document)	Complex Type **CashSettlementReferenceBanks**
Abstract	no
Documentation	A type to describe an institution (party) identified by means of a coding scheme and an optional name.

XML Instance Representation

```
<...>
   <referenceBankId> ReferenceBankId </referenceBankId> [1]
   'An institution (party) identifier, e.g. a bank identifier code (BIC).'

   <referenceBankName> xsd:string </referenceBankName> [0..1]
   'The name of the institution (party). A free format string. FpML does not define usage
   rules for the element.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="ReferenceBank">
   <xsd:sequence>
      <xsd:element name="referenceBankId" type=" ReferenceBankId "/>
      <xsd:element name="referenceBankName" type=" xsd:string " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: ReferenceBankId

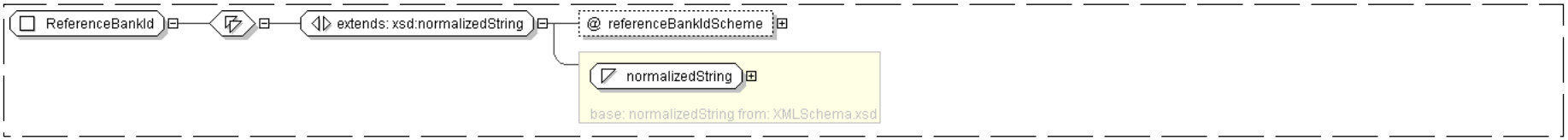
Super-types:	xsd:normalizedString < **ReferenceBankId** (by extension)
Sub-types:	None

Name	ReferenceBankId
Used by (from the same schema document)	Complex Type **ReferenceBank**
Abstract	no

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: RelativeDateOffset

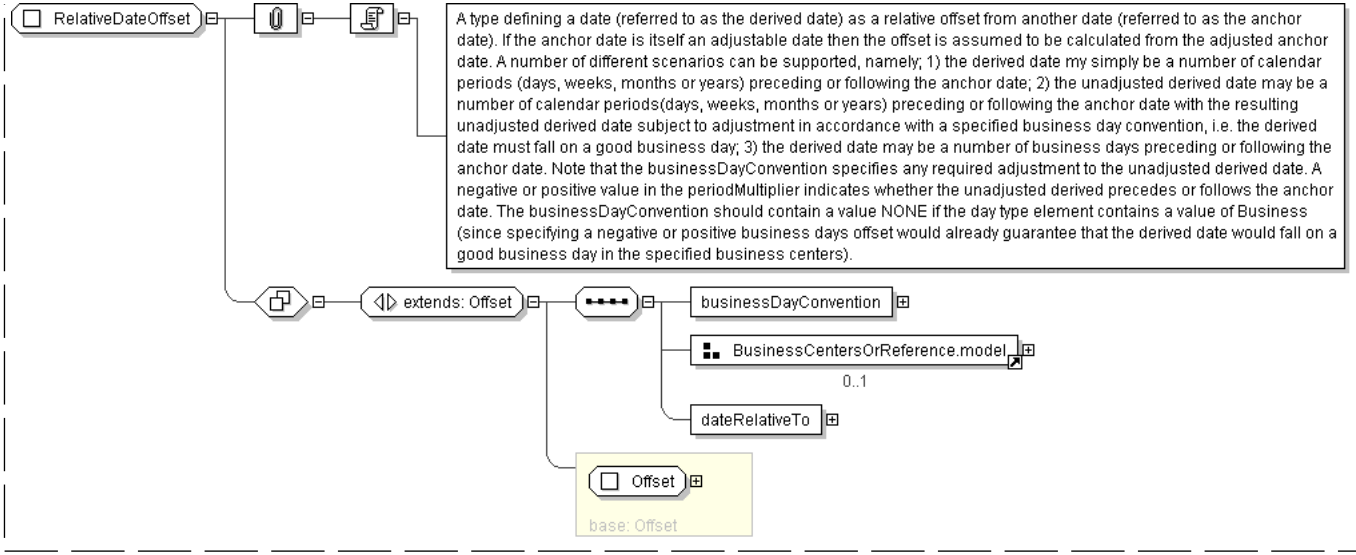
Super-types:	Interval < Offset (by extension) < **RelativeDateOffset** (by extension)
Sub-types:	• AdjustedRelativeDateOffset (by extension) • RelativeDates (by extension)

Name	RelativeDateOffset
Used by (from the same schema document)	Complex Type **AdjustableDatesOrRelativeDateOffset** , Complex Type **AdjustableOrRelativeDate** , Complex Type **ExerciseFee** , Complex Type **ExerciseFeeSchedule**
Abstract	no
Documentation	A type defining a date (referred to as the derived date) as a relative offset from another date (referred to as the anchor date). If the anchor date is itself an adjustable date then the offset is assumed to be calculated from the adjusted anchor date. A number of different scenarios can be supported, namely; 1) the derived date my simply be a number of calendar periods (days, weeks, months or years) preceding or following the anchor date; 2) the unadjusted derived date may be a number of calendar periods(days, weeks, months or years) preceding or following the anchor date with the resulting unadjusted derived date subject to adjustment in accordance with a specified business day convention, i.e. the derived date must fall on a good business day; 3) the derived date may be a number of business days preceding or following the anchor date. Note that the businessDayConvention specifies any required adjustment to the unadjusted derived date. A negative or positive value in the periodMultiplier indicates whether the unadjusted derived precedes or follows the anchor date. The businessDayConvention should contain a value NONE if the day type element contains a value of Business (since specifying a negative or positive business days offset would already guarantee that the derived date would fall on a good business day in the specified business centers).

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
   <periodMultiplier> xsd:integer </periodMultiplier> [1]
   'A time period multiplier, e.g. 1, 2 or 3 etc. A negative value can be used when specifying
   an offset relative to another date, e.g. -2 days. If the period value is T (Term)
   then periodMultiplier must contain the value 1.'

   <period> PeriodEnum </period> [1]
   'A time period, e.g. a day, week, month, year or term of the stream. If the
   periodMultiplier value is 0 (zero) then period must contain the value D (day).'

   <dayType> DayTypeEnum </dayType> [0..1]
   'In the case of an offset specified as a number of days, this element defines
   whether consideration is given as to whether a day is a good business day or not. If a day
   type of business days is specified then non-business days are ignored when calculating
   the offset. The financial business centers to use for determination of business days
   are implied by the context in which this element is used. This element must only be
   included when the offset is specified as a number of days. If the offset is zero days then
   the dayType element should not be included.'

   <businessDayConvention> BusinessDayConventionEnum </businessDayConvention> [1]
   'The convention for adjusting a date if it would otherwise fall on a day that is not a
   business day.'

Start Group: BusinessCentersOrReference.model [0..1]
Start Choice [1]
         <businessCentersReference> BusinessCentersReference </businessCentersReference> [1]
         'A pointer style reference to a set of financial business centers defined elsewhere in
         the document. This set of business centers is used to determine whether a particular day is
         a business day or not.'

         <businessCenters> BusinessCenters </businessCenters> [1]
End Choice
End Group: BusinessCentersOrReference.model
   <dateRelativeTo> DateReference </dateRelativeTo> [1]
   'Specifies the anchor as an href attribute. The href attribute value is a pointer
   style reference to the element or component elsewhere in the document where the anchor date
   is defined.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="RelativeDateOffset">
   <xsd:complexContent>
      <xsd:extension base=" Offset ">
         <xsd:sequence>
            <xsd:element name="businessDayConvention" type=" BusinessDayConventionEnum "/>
            <xsd:group ref=" BusinessCentersOrReference.model " minOccurs="0"/>
            <xsd:element name="dateRelativeTo" type=" DateReference "/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: RelativeDateSequence

Super-types:	None
Sub-types:	None

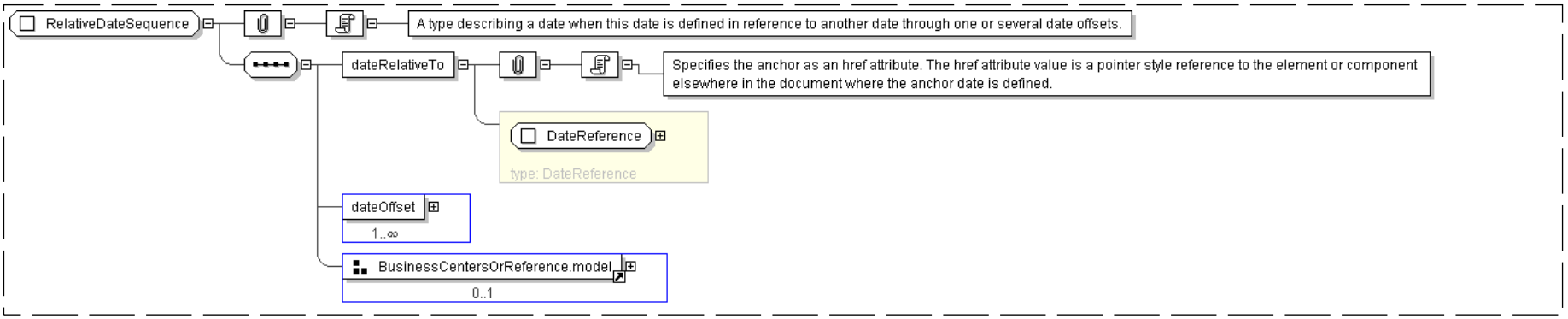
Name	RelativeDateSequence
Used by (from the same schema document)	Complex Type **AdjustableRelativeOrPeriodicDates**
Abstract	no
Documentation	A type describing a date when this date is defined in reference to another date through one or several date offsets.

XML Instance Representation

```
<...>
   <dateRelativeTo> DateReference </dateRelativeTo> [1]
   'Specifies the anchor as an href attribute. The href attribute value is a pointer
   style reference to the element or component elsewhere in the document where the anchor date
   is defined.'

   <dateOffset> DateOffset </dateOffset> [1..*]
Start Group: BusinessCentersOrReference.model [0..1]
Start Choice [1]
         <businessCentersReference> BusinessCentersReference </businessCentersReference> [1]
         'A pointer style reference to a set of financial business centers defined elsewhere in
         the document. This set of business centers is used to determine whether a particular day is
         a business day or not.'
```

```
        <businessCenters> BusinessCenters </businessCenters> [1]
End Choice
End Group: BusinessCentersOrReference.model
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="RelativeDateSequence">
   <xsd:sequence>
      <xsd:element name="dateRelativeTo" type=" DateReference "/>
      <xsd:element name="dateOffset" type=" DateOffset " maxOccurs="unbounded"/>
      <xsd:group ref=" BusinessCentersOrReference.model " minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: RelativeDates

Super-types:	Interval < Offset (by extension) < RelativeDateOffset (by extension) < **RelativeDates** (by extension)
Sub-types:	None

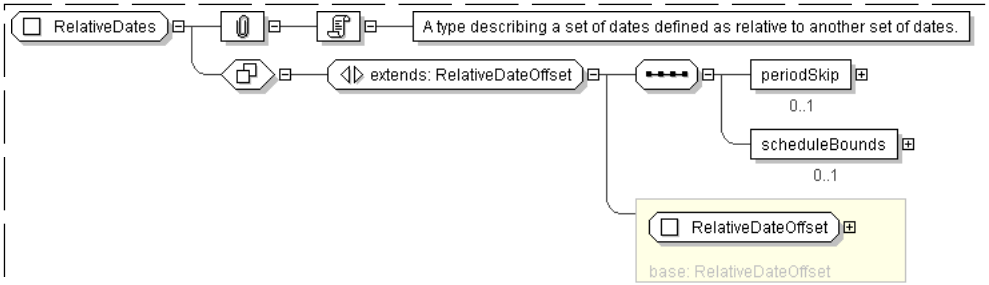
Name	RelativeDates
Used by (from the same schema document)	Complex Type **AdjustableOrRelativeDates**
Abstract	no
Documentation	A type describing a set of dates defined as relative to another set of dates.

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
   <periodMultiplier> xsd:integer </periodMultiplier> [1]
   'A time period multiplier, e.g. 1, 2 or 3 etc. A negative value can be used when specifying
   an offset relative to another date, e.g. -2 days. If the period value is T (Term)
   then periodMultiplier must contain the value 1.'

   <period> PeriodEnum </period> [1]
   'A time period, e.g. a day, week, month, year or term of the stream. If the
   periodMultiplier value is 0 (zero) then period must contain the value D (day).'

   <dayType> DayTypeEnum </dayType> [0..1]
   'In the case of an offset specified as a number of days, this element defines
   whether consideration is given as to whether a day is a good business day or not. If a day
   type of business days is specified then non-business days are ignored when calculating
   the offset. The financial business centers to use for determination of business days
   are implied by the context in which this element is used. This element must only be
   included when the offset is specified as a number of days. If the offset is zero days then
   the dayType element should not be included.'
```

```
   <businessDayConvention> BusinessDayConventionEnum </businessDayConvention> [1]
   'The convention for adjusting a date if it would otherwise fall on a day that is not a
   business day.'

Start Group: BusinessCentersOrReference.model [0..1]
Start Choice [1]
         <businessCentersReference> BusinessCentersReference </businessCentersReference> [1]
         'A pointer style reference to a set of financial business centers defined elsewhere in
         the document. This set of business centers is used to determine whether a particular day is
         a business day or not.'

         <businessCenters> BusinessCenters </businessCenters> [1]
End Choice
End Group: BusinessCentersOrReference.model
   <dateRelativeTo> DateReference </dateRelativeTo> [1]
   'Specifies the anchor as an href attribute. The href attribute value is a pointer
   style reference to the element or component elsewhere in the document where the anchor date
   is defined.'

   <periodSkip> xsd:positiveInteger </periodSkip> [0..1]
   'The number of periods in the referenced date schedule that are between each date in
   the relative date schedule. Thus a skip of 2 would mean that dates are relative to every
   second date in the referenced schedule. If present this should have a value greater than 1.'

   <scheduleBounds> DateRange </scheduleBounds> [0..1]
   'The first and last dates of a schedule. This can be used to restrict the range of values in
   a reference series of dates.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="RelativeDates">
   <xsd:complexContent>
      <xsd:extension base=" RelativeDateOffset ">
         <xsd:sequence>
            <xsd:element name="periodSkip" type=" xsd:positiveInteger " minOccurs="0"/>
            <xsd:element name="scheduleBounds" type=" DateRange " minOccurs="0"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: RequiredIdentifierDate

Super-types:	xsd:date < **RequiredIdentifierDate** (by extension)
Sub-types:	None

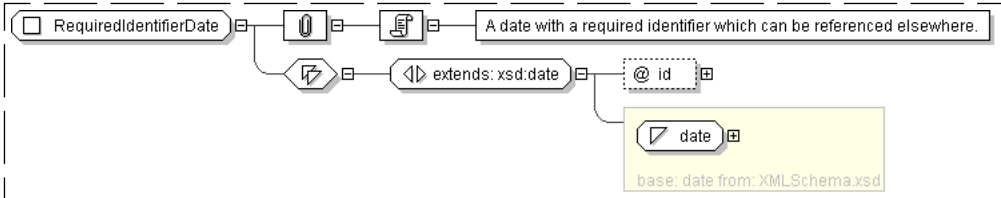
Name	RequiredIdentifierDate

Abstract	no
Documentation	A date with a required identifier which can be referenced elsewhere.

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:complexType name="RequiredIdentifierDate">
   <xsd:simpleContent>
      <xsd:extension base=" xsd:date ">
         <xsd:attribute name="id" type=" xsd:ID " use="required"/>
      </xsd:extension>
   </xsd:simpleContent>
</xsd:complexType>
```

top

Complex Type: ResetFrequency

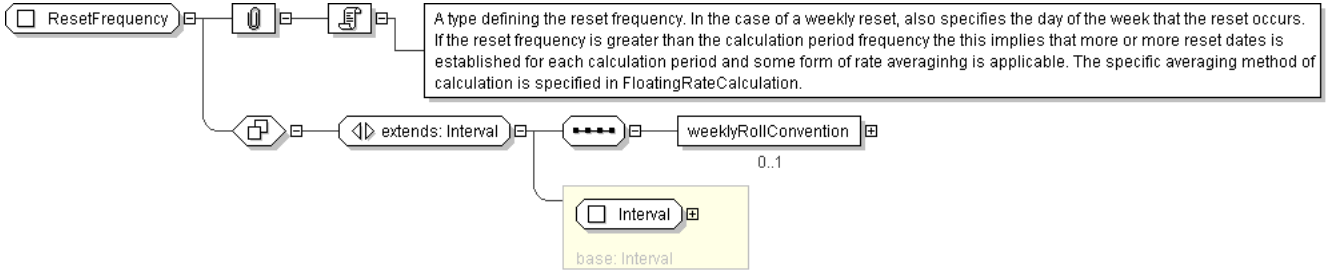
Super-types:	Interval < **ResetFrequency** (by extension)
Sub-types:	None

Name	ResetFrequency
Abstract	no
Documentation	A type defining the reset frequency. In the case of a weekly reset, also specifies the day of the week that the reset occurs. If the reset frequency is greater than the calculation period frequency the this implies that more or more reset dates is established for each calculation period and some form of rate averaginhg is applicable. The specific averaging method of calculation is specified in FloatingRateCalculation.

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
   <periodMultiplier> xsd:integer </periodMultiplier> [1]
   'A time period multiplier, e.g. 1, 2 or 3 etc. A negative value can be used when specifying
   an offset relative to another date, e.g. -2 days. If the period value is T (Term)
   then periodMultiplier must contain the value 1.'

   <period> PeriodEnum </period> [1]
   'A time period, e.g. a day, week, month, year or term of the stream. If the
   periodMultiplier value is 0 (zero) then period must contain the value D (day).'

   <weeklyRollConvention> WeeklyRollConventionEnum </weeklyRollConvention> [0..1]
   'The day of the week on which a weekly reset date occurs. This element must be included if
   the reset frequency is defined as weekly and not otherwise.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="ResetFrequency">
   <xsd:complexContent>
      <xsd:extension base=" Interval ">
         <xsd:sequence>
            <xsd:element name="weeklyRollConvention" type=" WeeklyRollConventionEnum " minOccurs="0"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: Rounding

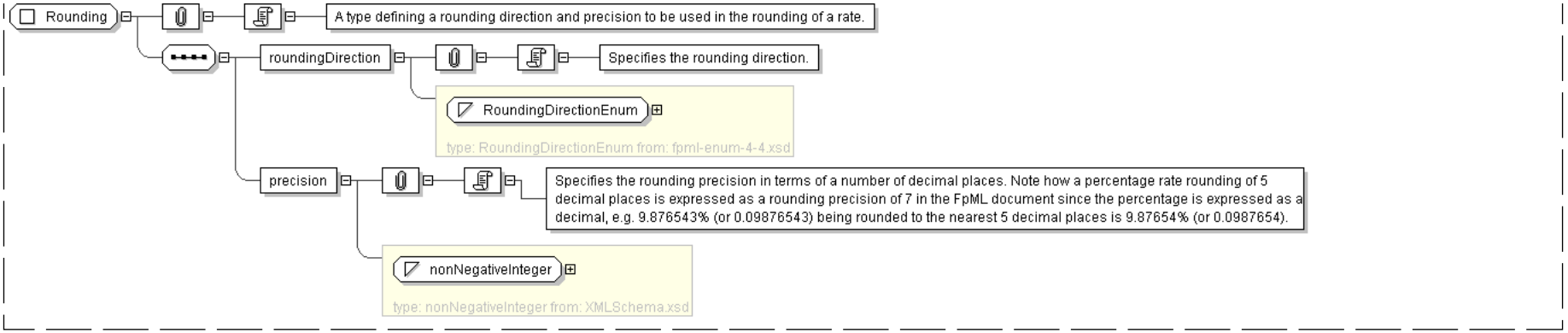
Super-types:	None
Sub-types:	None

Name	Rounding
Used by (from the same schema document)	Complex Type **FloatingRateCalculation**
Abstract	no
Documentation	A type defining a rounding direction and precision to be used in the rounding of a rate.

XML Instance Representation

```
<...>
   <roundingDirection> RoundingDirectionEnum </roundingDirection> [1]
   'Specifies the rounding direction.'

   <precision> xsd:nonNegativeInteger </precision> [1]
   'Specifies the rounding precision in terms of a number of decimal places. Note how a
   percentage rate rounding of 5 decimal places is expressed as a rounding precision of 7 in
   the FpML document since the percentage is expressed as a decimal, e.g. 9.876543%
   (or 0.09876543) being rounded to the nearest 5 decimal places is 9.87654% (or 0.0987654).'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Rounding">
   <xsd:sequence>
      <xsd:element name="roundingDirection" type=" RoundingDirectionEnum "/>
      <xsd:element name="precision" type=" xsd:nonNegativeInteger "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: Routing

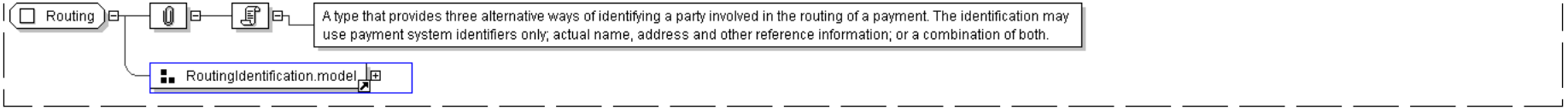
Super-types:	None
Sub-types:	None

Name	Routing
Used by (from the same schema document)	Complex Type **SplitSettlement** , Complex Type **SplitSettlement**
Abstract	no
Documentation	A type that provides three alternative ways of identifying a party involved in the routing of a payment. The identification may use payment system identifiers only; actual name, address and other reference information; or a combination of both.

XML Instance Representation

```
<...>
Start Choice [1]
      <routingIds> RoutingIds </routingIds> [1]
      'A set of unique identifiers for a party, eachone identifying the party within a
      payment system. The assumption is that each party will not have more than one identifier
      within the same payment system.'

      <routingExplicitDetails> RoutingExplicitDetails </routingExplicitDetails> [1]
      'A set of details that is used to identify a party involved in the routing of a payment
      when the party does not have a code that identifies it within one of the recognized
      payment systems.'

      <routingIdsAndExplicitDetails> RoutingIdsAndExplicitDetails </routingIdsAndExplicitDetails> [1]
      'A combination of coded payment system identifiers and details for physical addressing for
      a party involved in the routing of a payment.'

End Choice
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Routing">
   <xsd:group ref=" RoutingIdentification.model "/>
</xsd:complexType>
```

top

Complex Type: RoutingExplicitDetails

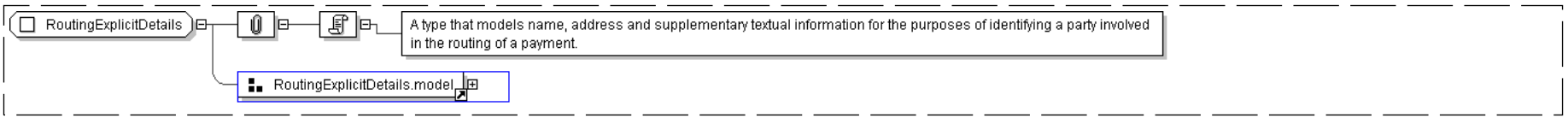
Super-types:	None
Sub-types:	None

Name	RoutingExplicitDetails
Used by (from the same schema document)	Model Group **RoutingIdentification.model**
Abstract	no
Documentation	A type that models name, address and supplementary textual information for the purposes of identifying a party involved in the routing of a payment.

XML Instance Representation

```
<...>
   <routingName> xsd:string </routingName> [1]
   'A real name that is used to identify a party involved in the routing of a payment.'

   <routingAddress> Address </routingAddress> [0..1]
   'A physical postal address via which a payment can be routed.'

   <routingAccountNumber> xsd:string </routingAccountNumber> [0..1]
   'An account number via which a payment can be routed.'

   <routingReferenceText> xsd:string </routingReferenceText> [0..*]
   'A piece of free-format text used to assist the identification of a party involved in
   the routing of a payment.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="RoutingExplicitDetails">
   <xsd:group ref=" RoutingExplicitDetails.model "/>
</xsd:complexType>
```

top

Complex Type: RoutingId

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

Name	RoutingId

Used by (from the same schema document)	Complex Type **RoutingId**s
Abstract	no

XML Instance Representation

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

Diagram

Schema Component Representation

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

top

Complex Type: RoutingIds

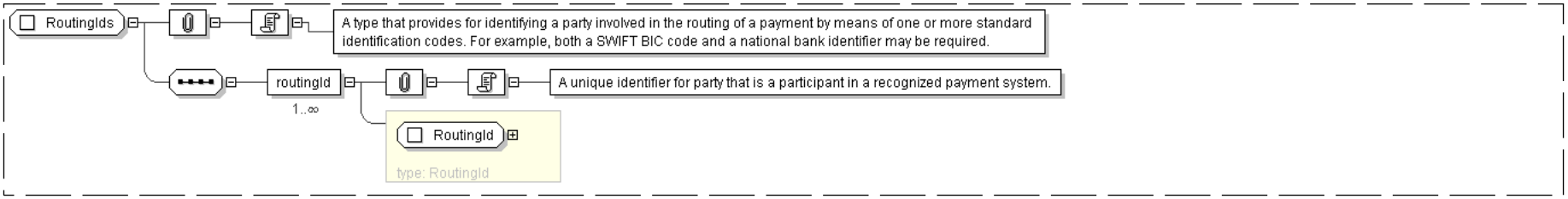
Super-types:	None
Sub-types:	None

Name	RoutingIds
Used by (from the same schema document)	Complex Type **RoutingIdsAndExplicitDetails** , Model Group **RoutingIdentification.model**
Abstract	no
Documentation	A type that provides for identifying a party involved in the routing of a payment by means of one or more standard identification codes. For example, both a SWIFT BIC code and a national bank identifier may be required.

XML Instance Representation

```
<...>
   <routingId> RoutingId </routingId> [1..*]
   'A unique identifier for party that is a participant in a recognized payment system.'

</...>
```

Diagram



Schema Component Representation

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

```
      <xsd:element name="routingId" type=" RoutingId " maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: RoutingIdsAndExplicitDetails

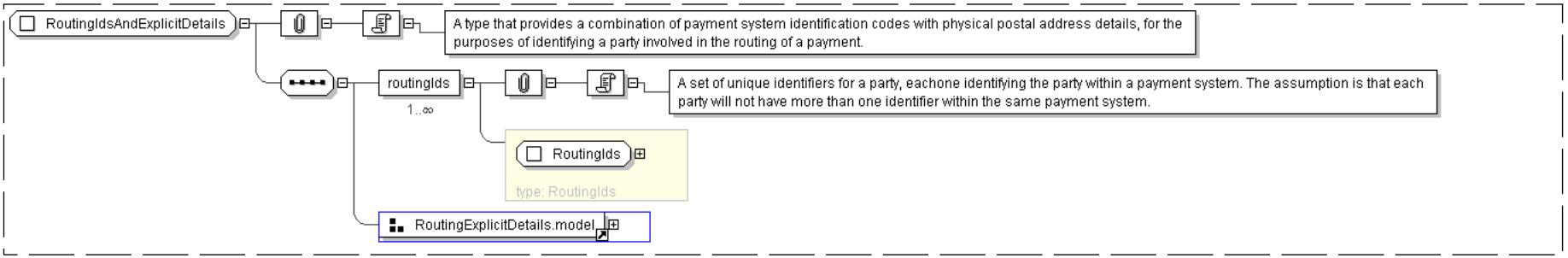
Super-types:	None
Sub-types:	None

Name	RoutingIdsAndExplicitDetails
Used by (from the same schema document)	Model Group **RoutingIdentification.model**
Abstract	no
Documentation	A type that provides a combination of payment system identification codes with physical postal address details, for the purposes of identifying a party involved in the routing of a payment.

XML Instance Representation

```
<...>
   <routingIds> RoutingIds </routingIds> [1..*]
   'A set of unique identifiers for a party, eachone identifying the party within a
   payment system. The assumption is that each party will not have more than one identifier
   within the same payment system.'

   <routingName> xsd:string </routingName> [1]
   'A real name that is used to identify a party involved in the routing of a payment.'

   <routingAddress> Address </routingAddress> [0..1]
   'A physical postal address via which a payment can be routed.'

   <routingAccountNumber> xsd:string </routingAccountNumber> [0..1]
   'An account number via which a payment can be routed.'

   <routingReferenceText> xsd:string </routingReferenceText> [0..*]
   'A piece of free-format text used to assist the identification of a party involved in
   the routing of a payment.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="RoutingIdsAndExplicitDetails">
   <xsd:sequence>
      <xsd:element name="routingIds" type=" RoutingIds " maxOccurs="unbounded"/>
      <xsd:group ref=" RoutingExplicitDetails.model "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: Schedule

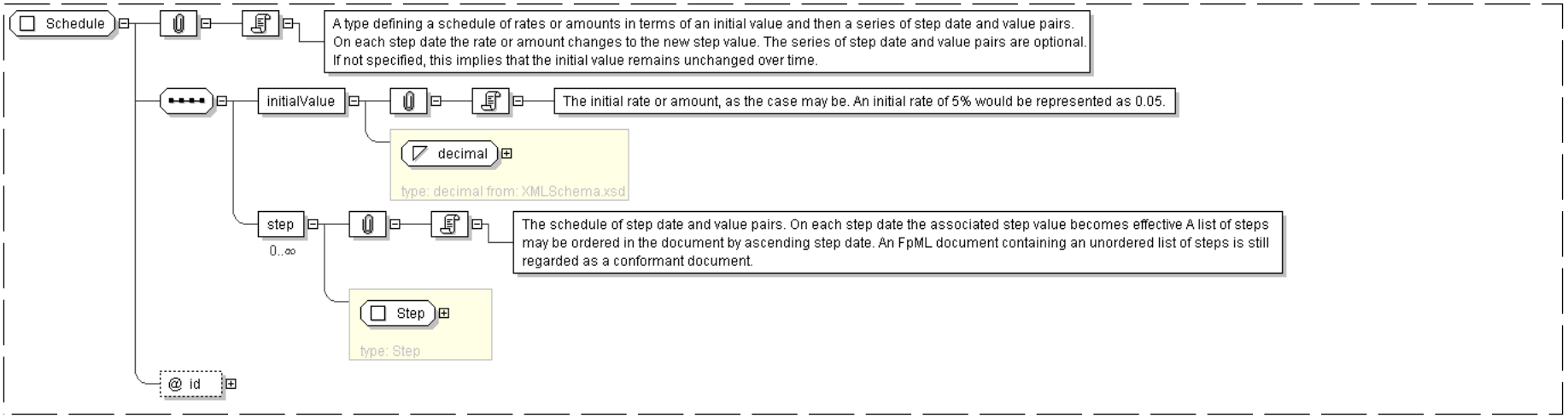
Super-types:	None
Sub-types:	• AmountSchedule (by extension) • SpreadSchedule (by extension) • StrikeSchedule (by extension)

Name	Schedule
Used by (from the same schema document)	Complex Type **ExerciseFeeSchedule** , Complex Type **FloatingRate**
Abstract	no
Documentation	A type defining a schedule of rates or amounts in terms of an initial value and then a series of step date and value pairs. On each step date the rate or amount changes to the new step value. The series of step date and value pairs are optional. If not specified, this implies that the initial value remains unchanged over time.

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
   <initialValue> xsd:decimal </initialValue> [1]
   'The initial rate or amount, as the case may be. An initial rate of 5% would be represented
   as 0.05.'

   <step> Step </step> [0..*]
   'The schedule of step date and value pairs. On each step date the associated step value
   becomes effective A list of steps may be ordered in the document by ascending step date.
   An FpML document containing an unordered list of steps is still regarded as a
   conformant document.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Schedule">
   <xsd:sequence>
      <xsd:element name="initialValue" type=" xsd:decimal "/>
      <xsd:element name="step" type=" Step " minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
   <xsd:attribute name="id" type=" xsd:ID "/>
</xsd:complexType>
```

top

Complex Type: ScheduleReference

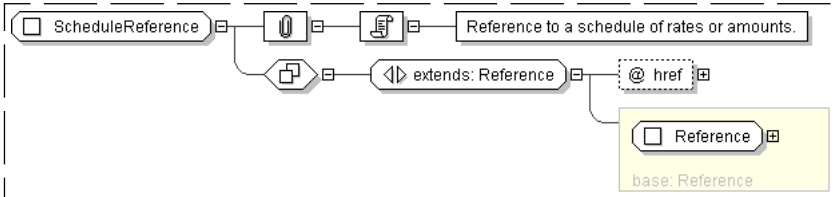
Super-types:	Reference < **ScheduleReference** (by extension)
Sub-types:	None

Name	ScheduleReference
Used by (from the same schema document)	Complex Type **ExerciseFee** , Complex Type **ExerciseFeeSchedule** , Model Group **PartialExercise.model**
Abstract	no
Documentation	Reference to a schedule of rates or amounts.

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: SettlementInformation

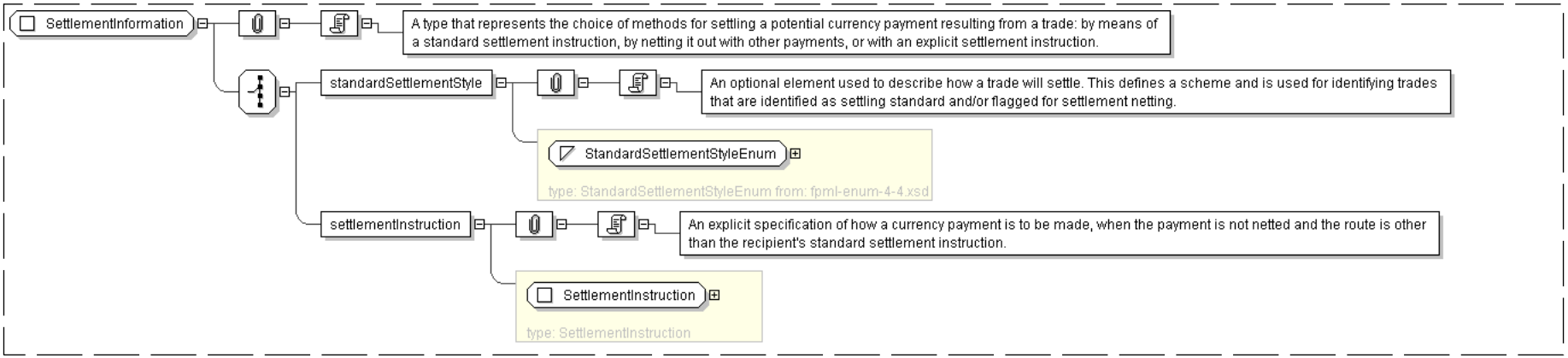
Super-types:	None
Sub-types:	None

Name	SettlementInformation
Used by (from the same schema document)	Complex Type **Payment**
Abstract	no
Documentation	A type that represents the choice of methods for settling a potential currency payment resulting from a trade: by means of a standard settlement instruction, by netting it out with other payments, or with an explicit settlement instruction.

XML Instance Representation

```
<...>
Start Choice [1]
      <standardSettlementStyle> StandardSettlementStyleEnum </standardSettlementStyle> [1]
      'An optional element used to describe how a trade will settle. This defines a scheme and
      is used for identifying trades that are identified as settling standard and/or flagged
      for settlement netting.'

      <settlementInstruction> SettlementInstruction </settlementInstruction> [1]
      'An explicit specification of how a currency payment is to be made, when the payment is
      not netted and the route is other than the recipient\'s standard settlement instruction.'

End Choice
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="SettlementInformation">
   <xsd:choice>
      <xsd:element name="standardSettlementStyle" type=" StandardSettlementStyleEnum "/>
      <xsd:element name="settlementInstruction" type=" SettlementInstruction "/>
   </xsd:choice>
</xsd:complexType>
```

top

Complex Type: SettlementInstruction

Super-types:	None
Sub-types:	None

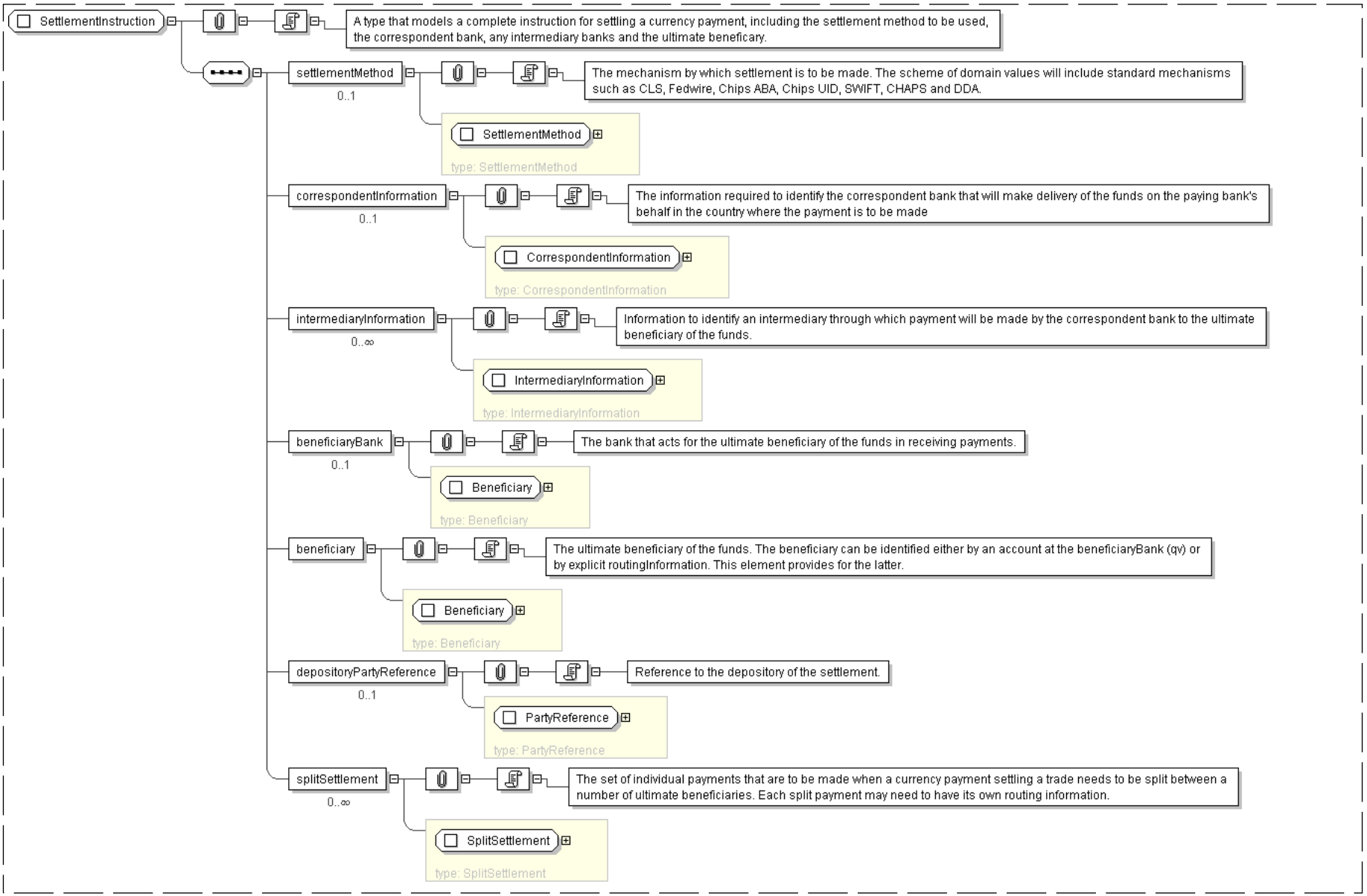
Name	SettlementInstruction
Used by (from the same schema document)	Complex Type **SettlementInformation**
Abstract	no
Documentation	A type that models a complete instruction for settling a currency payment, including the settlement method to be used, the correspondent bank, any intermediary banks and the ultimate beneficary.

XML Instance Representation

```
<...>
   <settlementMethod> SettlementMethod </settlementMethod> [0..1]
   'The mechanism by which settlement is to be made. The scheme of domain values will
   include standard mechanisms such as CLS, Fedwire, Chips ABA, Chips UID, SWIFT, CHAPS and DDA.'

   <correspondentInformation> CorrespondentInformation </correspondentInformation> [0..1]
   'The information required to identify the correspondent bank that will make delivery of
   the funds on the paying bank\'s behalf in the country where the payment is to be made'

   <intermediaryInformation> IntermediaryInformation </intermediaryInformation> [0..*]
   'Information to identify an intermediary through which payment will be made by
   the correspondent bank to the ultimate beneficiary of the funds.'

   <beneficiaryBank> Beneficiary </beneficiaryBank> [0..1]
   'The bank that acts for the ultimate beneficiary of the funds in receiving payments.'

   <beneficiary> Beneficiary </beneficiary> [1]
   'The ultimate beneficiary of the funds. The beneficiary can be identified either by an
   account at the beneficiaryBank (qv) or by explicit routingInformation. This element
   provides for the latter.'

   <depositoryPartyReference> PartyReference </depositoryPartyReference> [0..1]
   'Reference to the depository of the settlement.'
```

```
<splitSettlement> SplitSettlement </splitSettlement> [0..*]
'The set of individual payments that are to be made when a currency payment settling a
trade needs to be split between a number of ultimate beneficiaries. Each split payment may
need to have its own routing information.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="SettlementInstruction">
   <xsd:sequence>
      <xsd:element name="settlementMethod" type=" SettlementMethod " minOccurs="0"/>
```

```
      <xsd:element name="correspondentInformation" type=" CorrespondentInformation " minOccurs="0"/>
      <xsd:element name="intermediaryInformation" type=" IntermediaryInformation "
      minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="beneficiaryBank" type=" Beneficiary " minOccurs="0"/>
      <xsd:element name="beneficiary" type=" Beneficiary "/>
      <xsd:element name="depositoryPartyReference" type=" PartyReference " minOccurs="0"/>
      <xsd:element name="splitSettlement" type=" SplitSettlement "
      minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: SettlementMethod

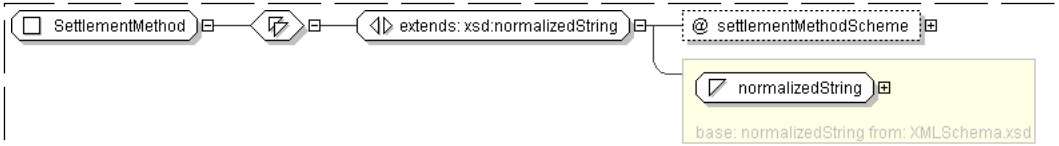
Super-types:	xsd:normalizedString < **SettlementMethod** (by extension)
Sub-types:	None

Name	SettlementMethod
Used by (from the same schema document)	Complex Type **SettlementInstruction**
Abstract	no

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: SettlementPriceSource

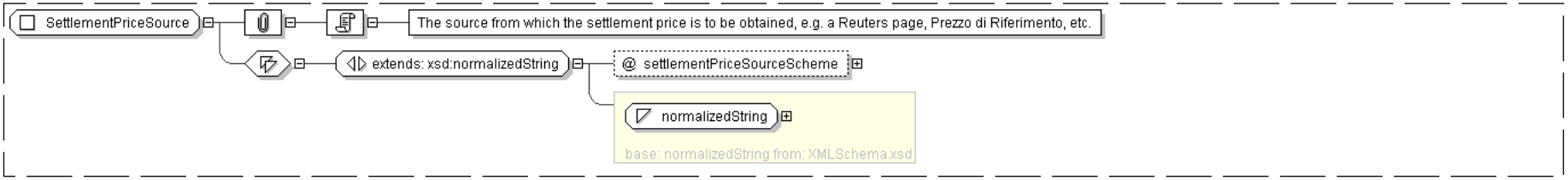
Super-types:	xsd:normalizedString < **SettlementPriceSource** (by extension)
Sub-types:	None

Name	SettlementPriceSource
Abstract	no
Documentation	The source from which the settlement price is to be obtained, e.g. a Reuters page, Prezzo di Riferimento, etc.

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: SettlementRateSource

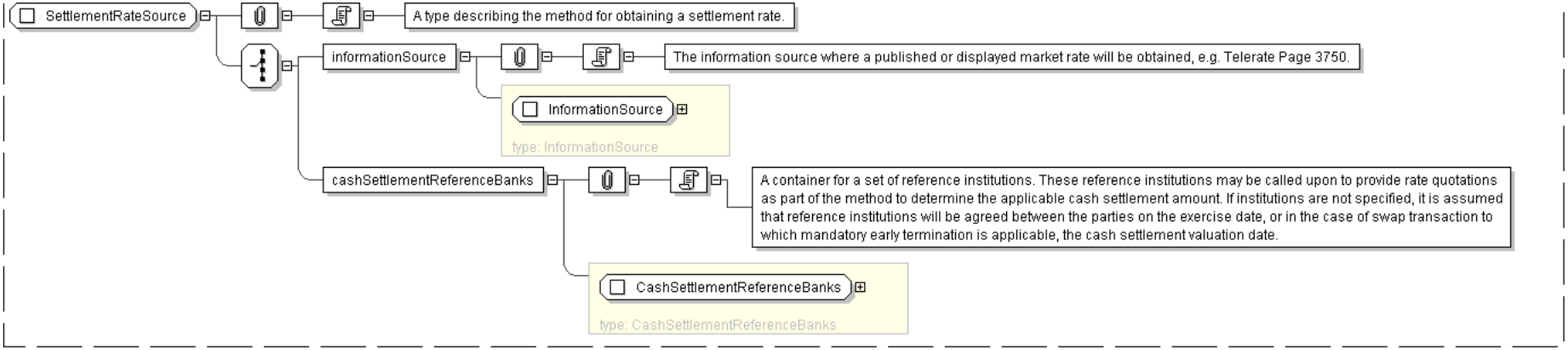
Super-types:	None
Sub-types:	None

Name	SettlementRateSource
Abstract	no
Documentation	A type describing the method for obtaining a settlement rate.

XML Instance Representation

```
<...>
Start Choice [1]
      <informationSource> InformationSource </informationSource> [1]
      'The information source where a published or displayed market rate will be obtained, e.
      g. Telerate Page 3750.'

      <cashSettlementReferenceBanks> CashSettlementReferenceBanks </cashSettlementReferenceBanks> [1]
      'A container for a set of reference institutions. These reference institutions may be
      called upon to provide rate quotations as part of the method to determine the applicable
      cash settlement amount. If institutions are not specified, it is assumed that
      reference institutions will be agreed between the parties on the exercise date, or in the
      case of swap transaction to which mandatory early termination is applicable, the
      cash settlement valuation date.'

End Choice
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="SettlementRateSource">
   <xsd:choice>
      <xsd:element name="informationSource" type=" InformationSource "/>
      <xsd:element name="cashSettlementReferenceBanks" type=" CashSettlementReferenceBanks "/>
   </xsd:choice>
</xsd:complexType>
```

top

Complex Type: SharedAmericanExercise

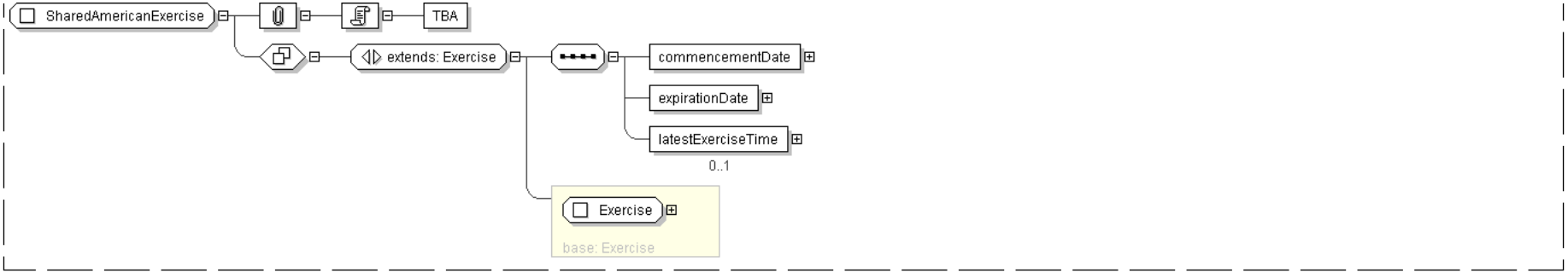
Super-types:	Exercise < **SharedAmericanExercise** (by extension)
Sub-types:	None

Name	SharedAmericanExercise
Abstract	no
Documentation	TBA

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
   <commencementDate> AdjustableOrRelativeDate </commencementDate> [1]
   'The first day of the exercise period for an American style option.'

   <expirationDate> AdjustableOrRelativeDate </expirationDate> [1]
   'The last day within an exercise period for an American style option. For a European
   style option it is the only day within the exercise period.'

   <latestExerciseTime> BusinessCenterTime </latestExerciseTime> [0..1]
   'For a Bermuda or American style option, the latest time on an exercise business day
   (excluding the expiration date) within the exercise period that notice can be given by
   the buyer to the seller or seller\'s agent. Notice of exercise given after this time will
   be deemed to have been given on the next exercise business day.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="SharedAmericanExercise">
   <xsd:complexContent>
      <xsd:extension base=" Exercise ">
         <xsd:sequence>
            <xsd:element name="commencementDate" type=" AdjustableOrRelativeDate "/>
            <xsd:element name="expirationDate" type=" AdjustableOrRelativeDate "/>
            <xsd:element name="latestExerciseTime" type=" BusinessCenterTime " minOccurs="0"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: SimplePayment

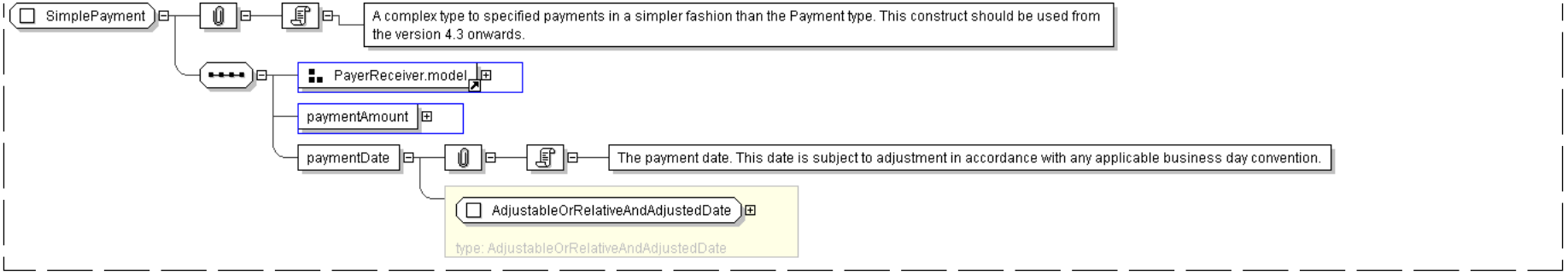
Super-types:	None
Sub-types:	None

Name	SimplePayment
Abstract	no
Documentation	A complex type to specified payments in a simpler fashion than the Payment type. This construct should be used from the version 4.3 onwards.

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.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="SimplePayment">
   <xsd:sequence>
      <xsd:group ref=" PayerReceiver.model "/>
      <xsd:element name="paymentAmount" type=" Money "/>
      <xsd:element name="paymentDate" type=" AdjustableOrRelativeAndAdjustedDate "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: SplitSettlement

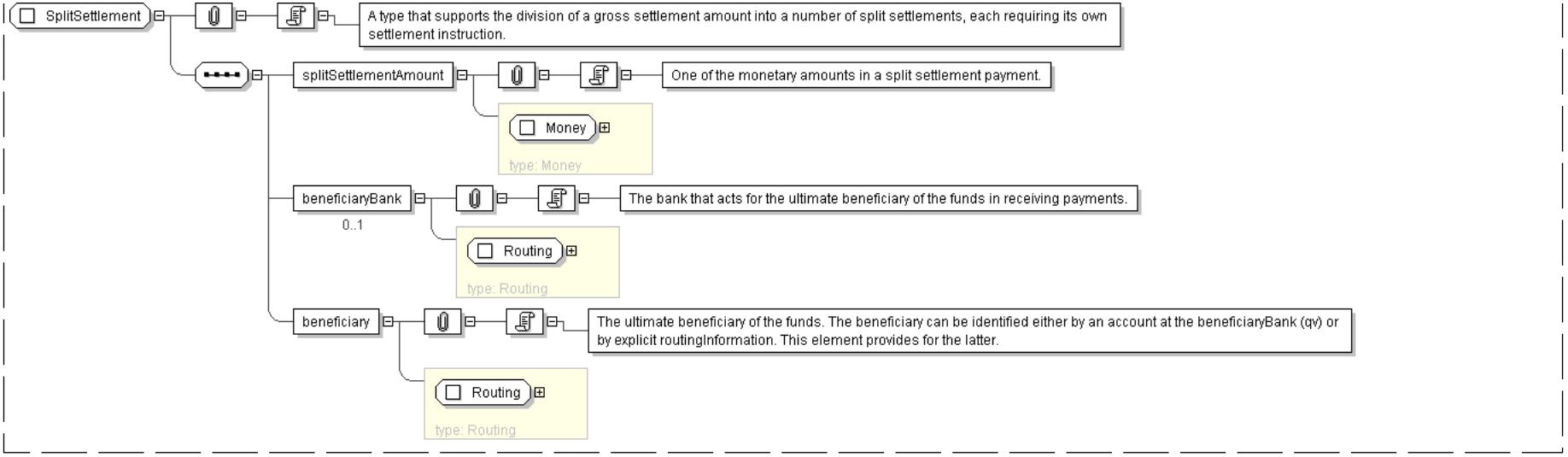
Super-types:	None
Sub-types:	None

Name	SplitSettlement
Used by (from the same schema document)	Complex Type **SettlementInstruction**
Abstract	no
Documentation	A type that supports the division of a gross settlement amount into a number of split settlements, each requiring its own settlement instruction.

XML Instance Representation

```
<...>
   <splitSettlementAmount> Money </splitSettlementAmount> [1]
   'One of the monetary amounts in a split settlement payment.'

   <beneficiaryBank> Routing </beneficiaryBank> [0..1]
   'The bank that acts for the ultimate beneficiary of the funds in receiving payments.'

   <beneficiary> Routing </beneficiary> [1]
   'The ultimate beneficiary of the funds. The beneficiary can be identified either by an
   account at the beneficiaryBank (qv) or by explicit routingInformation. This element
   provides for the latter.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="SplitSettlement">
   <xsd:sequence>
      <xsd:element name="splitSettlementAmount" type=" Money "/>
      <xsd:element name="beneficiaryBank" type=" Routing " minOccurs="0"/>
      <xsd:element name="beneficiary" type=" Routing "/>
   </xsd:sequence>
</xsd:complexType>
```

top

Complex Type: SpreadSchedule

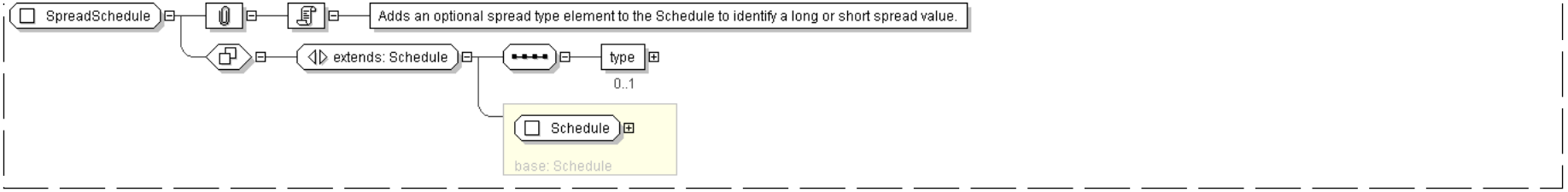
Super-types:	Schedule < **SpreadSchedule** (by extension)
Sub-types:	None

Name	SpreadSchedule
Used by (from the same schema document)	Complex Type **FloatingRate**
Abstract	no
Documentation	Adds an optional spread type element to the Schedule to identify a long or short spread value.

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
   <initialValue> xsd:decimal </initialValue> [1]
   'The initial rate or amount, as the case may be. An initial rate of 5% would be represented
   as 0.05.'

   <step> Step </step> [0..*]
   'The schedule of step date and value pairs. On each step date the associated step value
   becomes effective A list of steps may be ordered in the document by ascending step date.
   An FpML document containing an unordered list of steps is still regarded as a
   conformant document.'

   <type> SpreadScheduleType </type> [0..1]
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="SpreadSchedule">
   <xsd:complexContent>
      <xsd:extension base=" Schedule ">
         <xsd:sequence>
            <xsd:element name="type" type=" SpreadScheduleType " minOccurs="0"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: SpreadScheduleReference

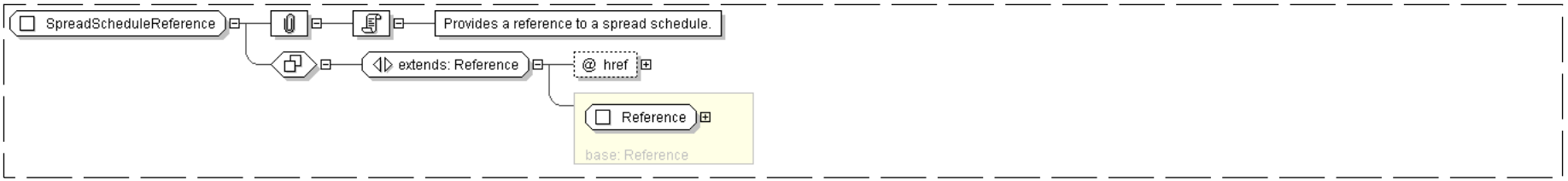
Super-types:	Reference < **SpreadScheduleReference** (by extension)
Sub-types:	None

Name	SpreadScheduleReference
Abstract	no
Documentation	Provides a reference to a spread schedule.

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: SpreadScheduleType

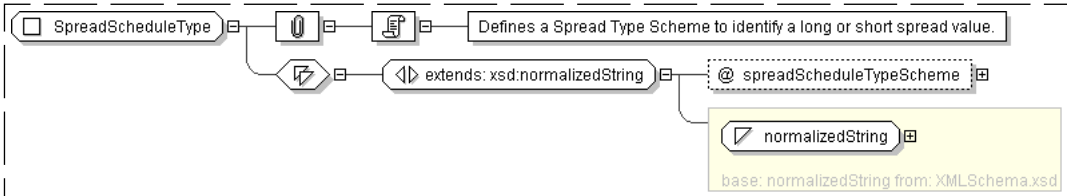
Super-types:	xsd:normalizedString < **SpreadScheduleType** (by extension)
Sub-types:	None

Name	SpreadScheduleType
Used by (from the same schema document)	Complex Type **SpreadSchedule**
Abstract	no
Documentation	Defines a Spread Type Scheme to identify a long or short spread value.

XML Instance Representation

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

Diagram



Schema Component Representation

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

top

Complex Type: Step

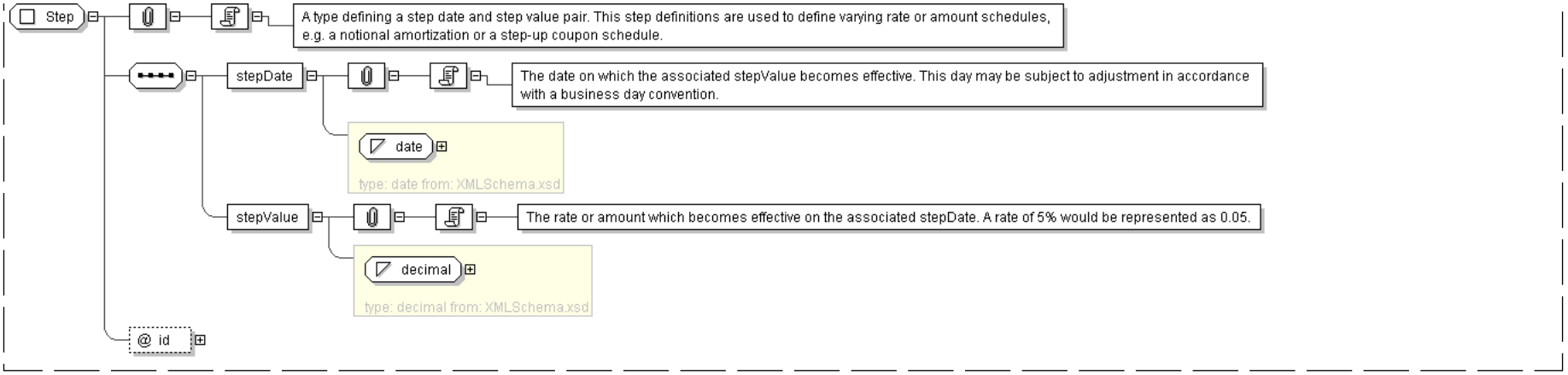
Super-types:	None
Sub-types:	None

Name	Step
Used by (from the same schema document)	Complex Type **Schedule**
Abstract	no
Documentation	A type defining a step date and step value pair. This step definitions are used to define varying rate or amount schedules, e.g. a notional amortization or a step-up coupon schedule.

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
   <stepDate> xsd:date </stepDate> [1]
   'The date on which the associated stepValue becomes effective. This day may be subject
   to adjustment in accordance with a business day convention.'

   <stepValue> xsd:decimal </stepValue> [1]
   'The rate or amount which becomes effective on the associated stepDate. A rate of 5% would
   be represented as 0.05.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Step">
   <xsd:sequence>
      <xsd:element name="stepDate" type=" xsd:date "/>
      <xsd:element name="stepValue" type=" xsd:decimal "/>
   </xsd:sequence>
   <xsd:attribute name="id" type=" xsd:ID "/>
</xsd:complexType>
```

top

Complex Type: StreetAddress

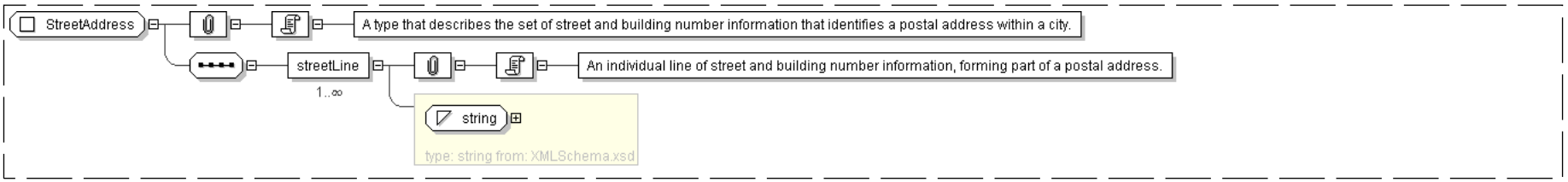
Super-types:	None
Sub-types:	None

Name	StreetAddress
Used by (from the same schema document)	Complex Type **Address**
Abstract	no
Documentation	A type that describes the set of street and building number information that identifies a postal address within a city.

XML Instance Representation

```
<...>
   <streetLine> xsd:string </streetLine> [1..*]
   'An individual line of street and building number information, forming part of a
   postal address.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="StreetAddress">
   <xsd:sequence>
      <xsd:element name="streetLine" type=" xsd:string " maxOccurs="unbounded"/>
```

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

top

Complex Type: Strike

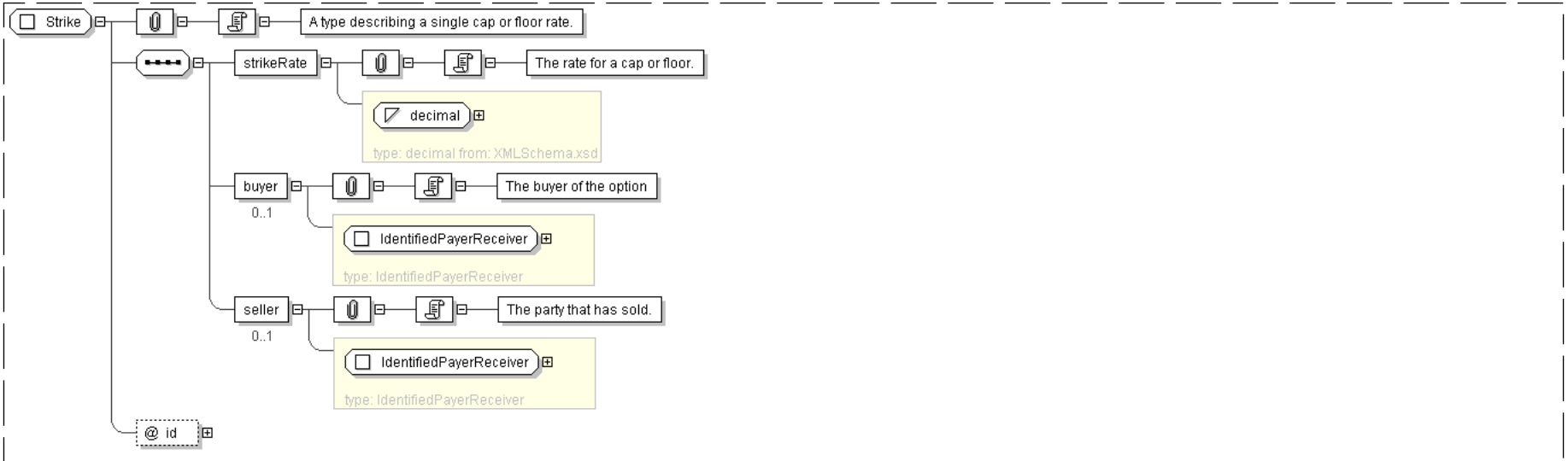
Super-types:	None
Sub-types:	None

Name	Strike
Abstract	no
Documentation	A type describing a single cap or floor rate.

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
   <strikeRate> xsd:decimal </strikeRate> [1]
   'The rate for a cap or floor.'

   <buyer> IdentifiedPayerReceiver </buyer> [0..1]
   'The buyer of the option'

   <seller> IdentifiedPayerReceiver </seller> [0..1]
   'The party that has sold.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Strike">
   <xsd:sequence>
      <xsd:element name="strikeRate" type=" xsd:decimal "/>
      <xsd:element name="buyer" type=" IdentifiedPayerReceiver " minOccurs="0"/>
      <xsd:element name="seller" type=" IdentifiedPayerReceiver " minOccurs="0"/>
   </xsd:sequence>
   <xsd:attribute name="id" type=" xsd:ID "/>
</xsd:complexType>
```

top

Complex Type: StrikeSchedule

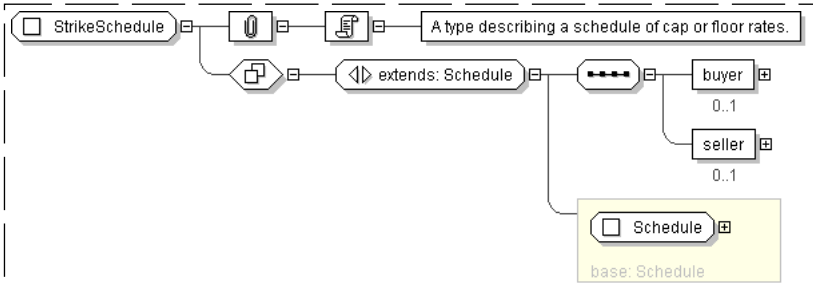
Super-types:	Schedule < **StrikeSchedule** (by extension)
Sub-types:	None

Name	StrikeSchedule
Used by (from the same schema document)	Complex Type **FloatingRate** , Complex Type **FloatingRate**
Abstract	no
Documentation	A type describing a schedule of cap or floor rates.

XML Instance Representation

```
<...
id=" xsd:ID [0..1]">
   <initialValue> xsd:decimal </initialValue> [1]
   'The initial rate or amount, as the case may be. An initial rate of 5% would be represented
   as 0.05.'

   <step> Step </step> [0..*]
   'The schedule of step date and value pairs. On each step date the associated step value
   becomes effective A list of steps may be ordered in the document by ascending step date.
   An FpML document containing an unordered list of steps is still regarded as a
   conformant document.'

   <buyer> IdentifiedPayerReceiver </buyer> [0..1]
   'The buyer of the option'

   <seller> IdentifiedPayerReceiver </seller> [0..1]
   'The party that has sold.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="StrikeSchedule">
   <xsd:complexContent>
      <xsd:extension base=" Schedule ">
         <xsd:sequence>
            <xsd:element name="buyer" type=" IdentifiedPayerReceiver " minOccurs="0"/>
            <xsd:element name="seller" type=" IdentifiedPayerReceiver " minOccurs="0"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: Stub

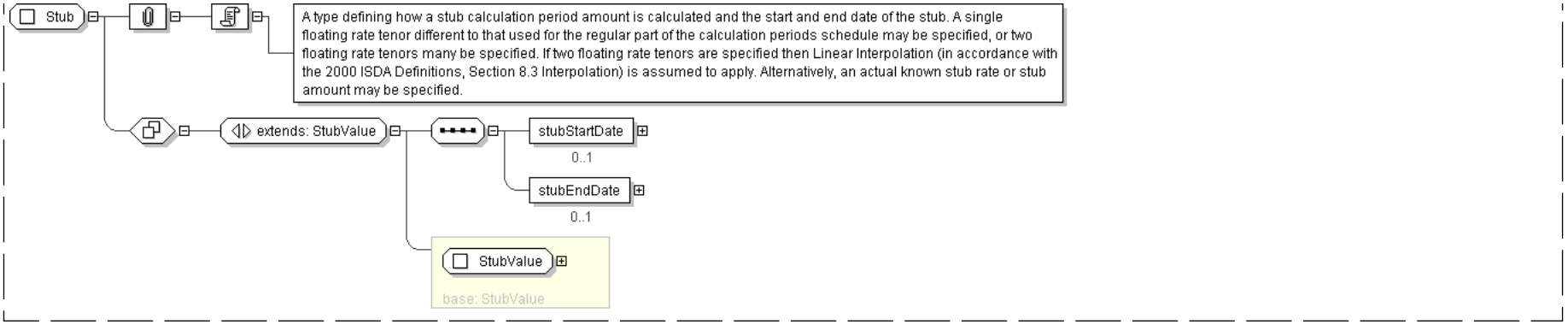
Super-types:	StubValue < **Stub** (by extension)
Sub-types:	None

Name	Stub
Abstract	no
Documentation	A type defining how a stub calculation period amount is calculated and the start and end date of the stub. A single floating rate tenor different to that used for the regular part of the calculation periods schedule may be specified, or two floating rate tenors many be specified. If two floating rate tenors are specified then Linear Interpolation (in accordance with the 2000 ISDA Definitions, Section 8.3 Interpolation) is assumed to apply. Alternatively, an actual known stub rate or stub amount may be specified.

XML Instance Representation

```
<...>
Start Choice [1]
      <floatingRate> FloatingRate </floatingRate> [1..*]
      'The rates to be applied to the initial or final stub may be the linear interpolation of
      two different rates. While the majority of the time, the rate indices will be the same as
      that specified in the stream and only the tenor itself will be different, it is possible
      to specift two different rates. For example, a 2 month stub period may use the
      linear interpolation of a 1 month and 3 month rate. The different rates would be specified
      in this component. Note that a maximum of two rates can be specified. If a stub period uses
      the same floating rate index, including tenor, as the regular calculation periods then
      this should not be specified again within this component, i.e. the stub calculation
      period amount component may not need to be specified even if there is an initial or final
      stub period. If a stub period uses a different floating rate index compared to the
      regular calculation periods then this should be specified within this component. If
      specified here, they are likely to have id attributes, allowing them to be referenced
      from within the cashflows component.'

      <stubRate> xsd:decimal </stubRate> [1]
      'An actual rate to apply for the initial or final stub period may have been agreed between
      the principal parties (in a similar way to how an initial rate may have been agreed for
      the first regular period). If an actual stub rate has been agreed then it would be included
      in this component. It will be a per annum rate, expressed as a decimal. A stub rate of 5%
      would be represented as 0.05.'

      <stubAmount> Money </stubAmount> [1]
      'An actual amount to apply for the initial or final stub period may have been agreed between
      th two parties. If an actual stub amount has been agreed then it would be included in
      this component.'

End Choice
   <stubStartDate> AdjustableOrRelativeDate </stubStartDate> [0..1]
   'Start date of stub period. This was created to support use of the InterestRateStream
   within the Equity Derivative sphere, and this element is not expected to be produced in
   the representation of Interest Rate products.'

   <stubEndDate> AdjustableOrRelativeDate </stubEndDate> [0..1]
   'End date of stub period. This was created to support use of the InterestRateStream within
   the Equity Derivative sphere, and this element is not expected to be produced in
   the representation of Interest Rate products.'

</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="Stub">
   <xsd:complexContent>
      <xsd:extension base=" StubValue ">
         <xsd:sequence>
            <xsd:element name="stubStartDate" type=" AdjustableOrRelativeDate " minOccurs="0"/>
            <xsd:element name="stubEndDate" type=" AdjustableOrRelativeDate " minOccurs="0"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

top

Complex Type: StubValue

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

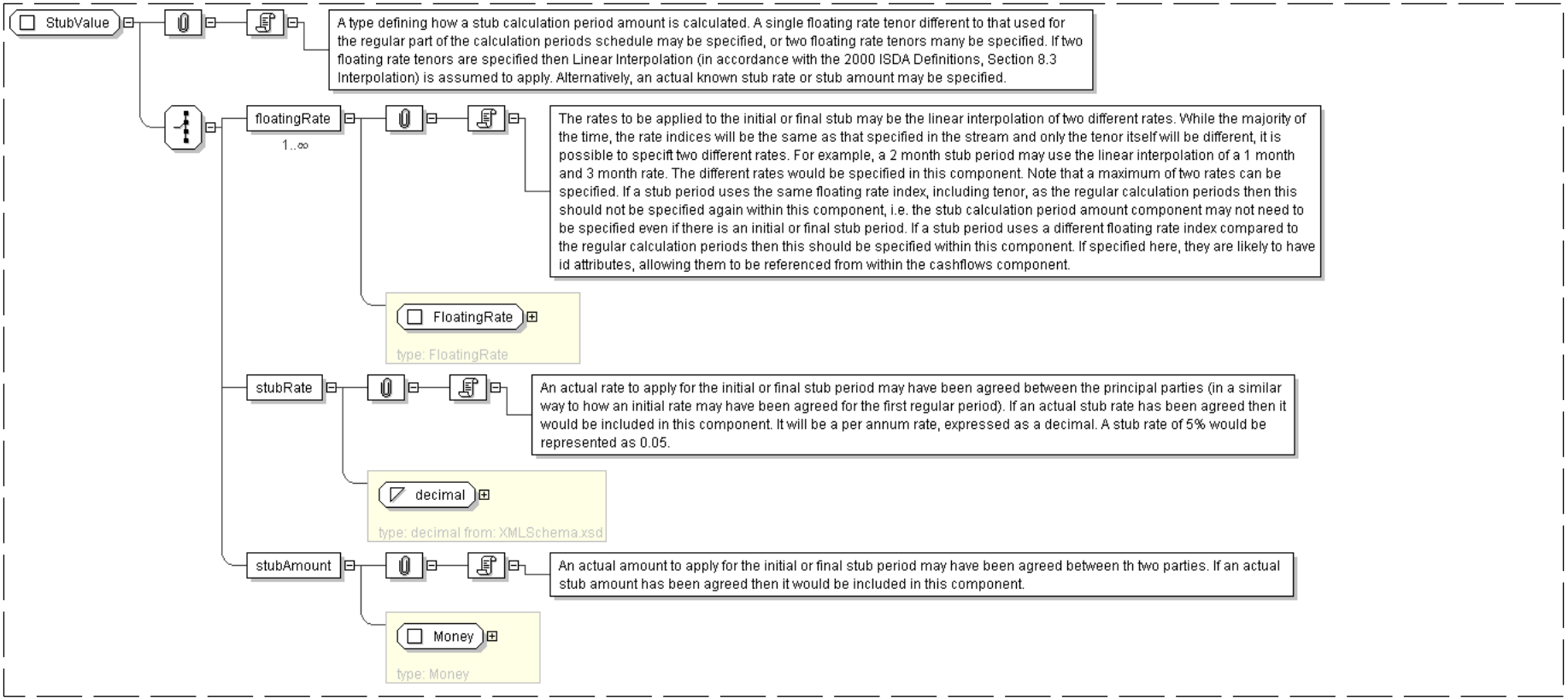
Name	StubValue
Abstract	no
Documentation	A type defining how a stub calculation period amount is calculated. A single floating rate tenor different to that used for the regular part of the calculation periods schedule may be specified, or two floating rate tenors many be specified. If two floating rate tenors are specified then Linear Interpolation (in accordance with the 2000 ISDA Definitions, Section 8.3 Interpolation) is assumed to apply. Alternatively, an actual known stub rate or stub amount may be specified.

XML Instance Representation

```
<...>
Start Choice [1]
      <floatingRate> FloatingRate </floatingRate> [1..*]
      'The rates to be applied to the initial or final stub may be the linear interpolation of
      two different rates. While the majority of the time, the rate indices will be the same as
      that specified in the stream and only the tenor itself will be different, it is possible
      to specift two different rates. For example, a 2 month stub period may use the
      linear interpolation of a 1 month and 3 month rate. The different rates would be specified
      in this component. Note that a maximum of two rates can be specified. If a stub period uses
      the same floating rate index, including tenor, as the regular calculation periods then
      this should not be specified again within this component, i.e. the stub calculation
      period amount component may not need to be specified even if there is an initial or final
      stub period. If a stub period uses a different floating rate index compared to the
      regular calculation periods then this should be specified within this component. If
      specified here, they are likely to have id attributes, allowing them to be referenced
      from within the cashflows component.'

      <stubRate> xsd:decimal </stubRate> [1]
      'An actual rate to apply for the initial or final stub period may have been agreed between
      the principal parties (in a similar way to how an initial rate may have been agreed for
      the first regular period). If an actual stub rate has been agreed then it would be included
      in this component. It will be a per annum rate, expressed as a decimal. A stub rate of 5%
      would be represented as 0.05.'
```

```
      <stubAmount> Money </stubAmount> [1]
      'An actual amount to apply for the initial or final stub period may have been agreed between
      th two parties. If an actual stub amount has been agreed then it would be included in
      this component.'

End Choice
</...>
```

Diagram



Schema Component Representation

```
<xsd:complexType name="StubValue">
   <xsd:choice>
      <xsd:element name="floatingRate" type=" FloatingRate " maxOccurs="unbounded"/>
      <xsd:element name="stubRate" type=" xsd:decimal "/>
      <xsd:element name="stubAmount" type=" Money "/>
   </xsd:choice>
</xsd:complexType>
```

top

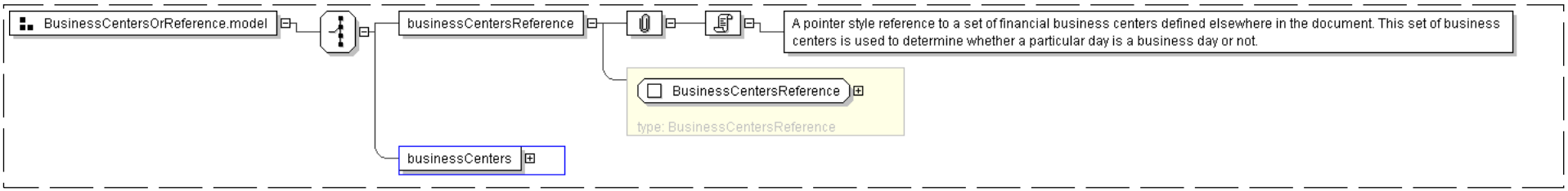
Model Group: BusinessCentersOrReference.model

Name	BusinessCentersOrReference.model
Used by (from the same schema document)	Complex Type **BusinessDateRange** , Complex Type **BusinessDayAdjustments** , Complex Type **RelativeDateOffset** , Complex Type **RelativeDateSequence**

XML Instance Representation

```
Start Choice [1]
   <businessCentersReference> BusinessCentersReference </businessCentersReference> [1]
   'A pointer style reference to a set of financial business centers defined elsewhere in
   the document. This set of business centers is used to determine whether a particular day is
   a business day or not.'

   <businessCenters> BusinessCenters </businessCenters> [1]
End Choice
```

Diagram



Schema Component Representation

```
<xsd:group name="BusinessCentersOrReference.model">
   <xsd:choice>
      <xsd:element name="businessCentersReference" type=" BusinessCentersReference "/>
      <xsd:element name="businessCenters" type=" BusinessCenters "/>
   </xsd:choice>
</xsd:group>
```

top

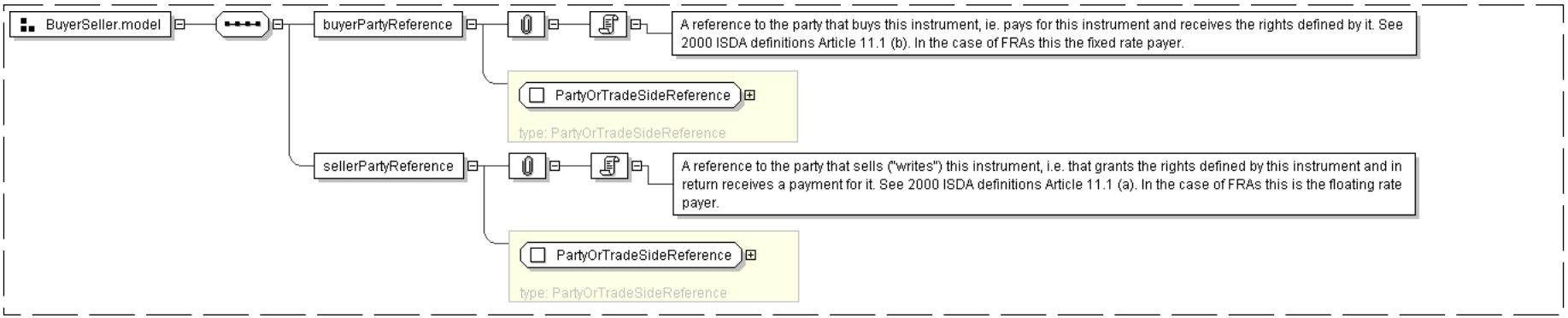
Model Group: BuyerSeller.model

Name	BuyerSeller.model

XML Instance Representation

```
<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.'
```

Diagram



Schema Component Representation

```
<xsd:group name="BuyerSeller.model">
   <xsd:sequence>
      <xsd:element name="buyerPartyReference" type=" PartyOrTradeSideReference "/>
      <xsd:element name="sellerPartyReference" type=" PartyOrTradeSideReference "/>
   </xsd:sequence>
</xsd:group>
```

top

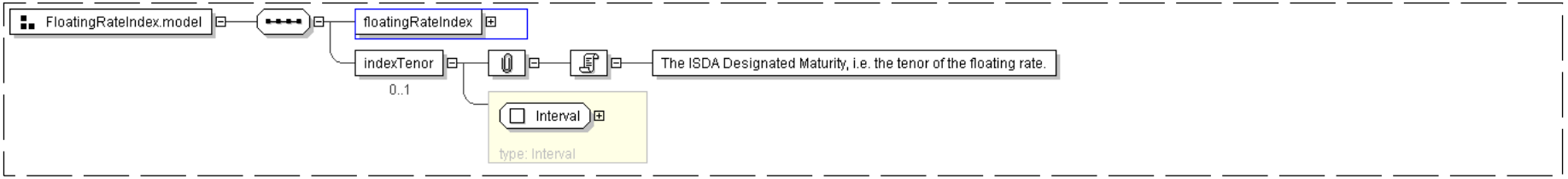
Model Group: FloatingRateIndex.model

Name	FloatingRateIndex.model
Used by (from the same schema document)	Complex Type **FloatingRate**

XML Instance Representation

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

Diagram



Schema Component Representation

```
<xsd:group name="FloatingRateIndex.model">
   <xsd:sequence>
      <xsd:element name="floatingRateIndex" type=" FloatingRateIndex "/>
      <xsd:element name="indexTenor" type=" Interval " minOccurs="0"/>
   </xsd:sequence>
</xsd:group>
```

top

Model Group: PartialExercise.model

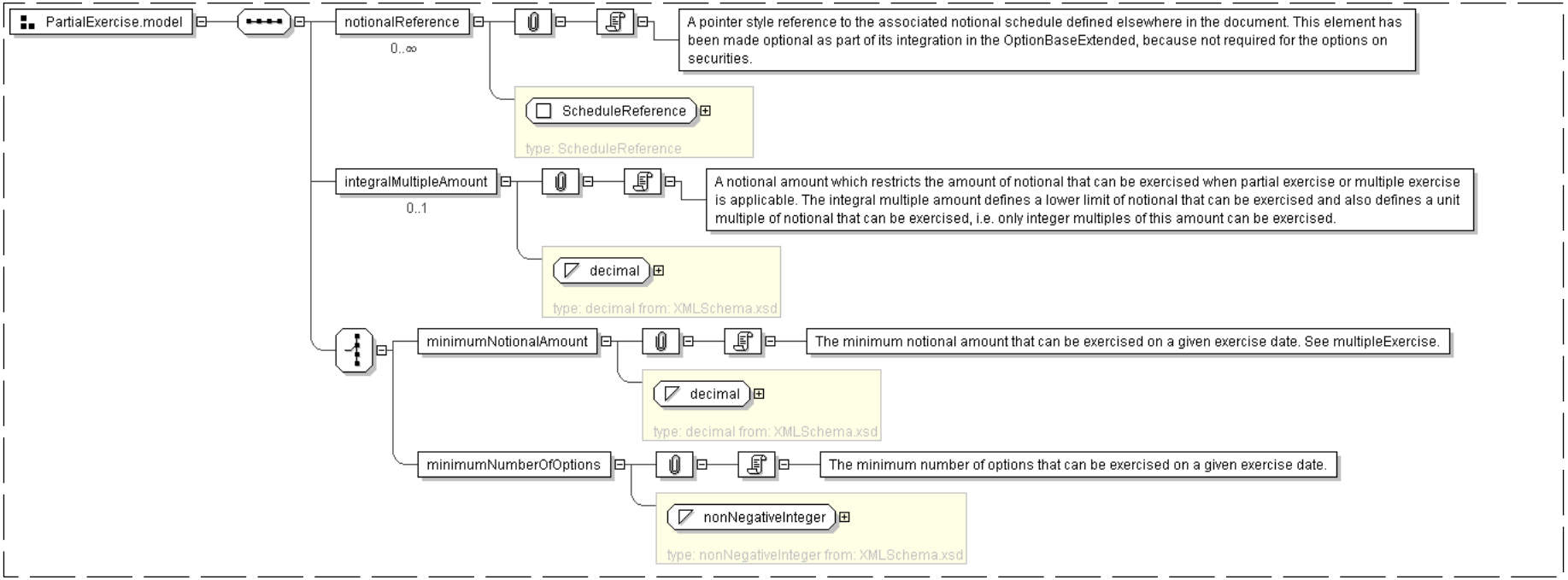
Name	PartialExercise.model
Used by (from the same schema document)	Complex Type **MultipleExercise** , Complex Type **PartialExercise**

XML Instance Representation

```
<notionalReference> ScheduleReference </notionalReference> [0..*]
'A pointer style reference to the associated notional schedule defined elsewhere in
the document. This element has been made optional as part of its integration in
the OptionBaseExtended, because not required for the options on securities.'

<integralMultipleAmount> xsd:decimal </integralMultipleAmount> [0..1]
'A notional amount which restricts the amount of notional that can be exercised when
partial exercise or multiple exercise is applicable. The integral multiple amount defines
a lower limit of notional that can be exercised and also defines a unit multiple of
notional that can be exercised, i.e. only integer multiples of this amount can be exercised.'

Start Choice [1]
   <minimumNotionalAmount> xsd:decimal </minimumNotionalAmount> [1]
   'The minimum notional amount that can be exercised on a given exercise date.
   See multipleExercise.'
```

```
<minimumNumberOfOptions> xsd:nonNegativeInteger </minimumNumberOfOptions> [1]
'The minimum number of options that can be exercised on a given exercise date.'

End Choice
```

Diagram



Schema Component Representation

```
<xsd:group name="PartialExercise.model">
   <xsd:sequence>
      <xsd:element name="notionalReference" type=" ScheduleReference "
      minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="integralMultipleAmount" type=" xsd:decimal " minOccurs="0"/>
      <xsd:choice>
         <xsd:element name="minimumNotionalAmount" type=" xsd:decimal "/>
         <xsd:element name="minimumNumberOfOptions" type=" xsd:nonNegativeInteger "/>
      </xsd:choice>
   </xsd:sequence>
</xsd:group>
```

top

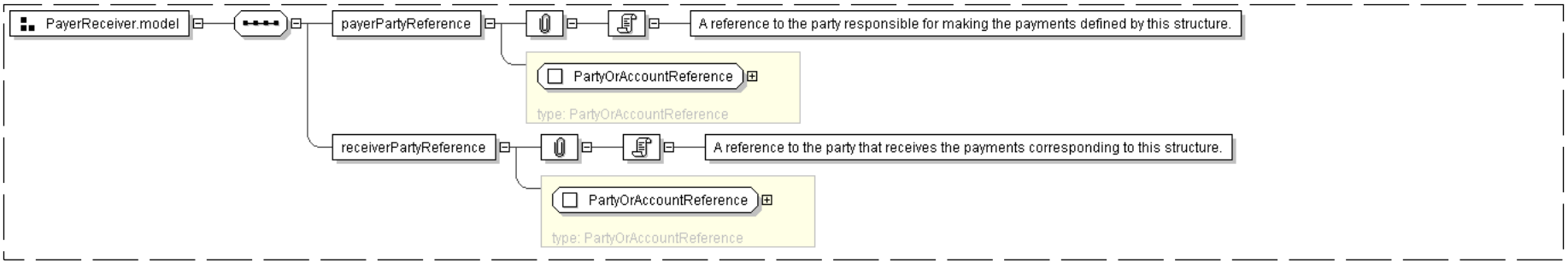
Model Group: PayerReceiver.model

Name	PayerReceiver.model
Used by (from the same schema document)	Complex Type **ExerciseFee** , Complex Type **ExerciseFeeSchedule** , Complex Type **Payment** , Complex Type **SimplePayment**

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.'
```

Diagram



Schema Component Representation

```
<xsd:group name="PayerReceiver.model">
   <xsd:sequence>
      <xsd:element name="payerPartyReference" type=" PartyOrAccountReference "/>
      <xsd:element name="receiverPartyReference" type=" PartyOrAccountReference "/>
   </xsd:sequence>
</xsd:group>
```

top

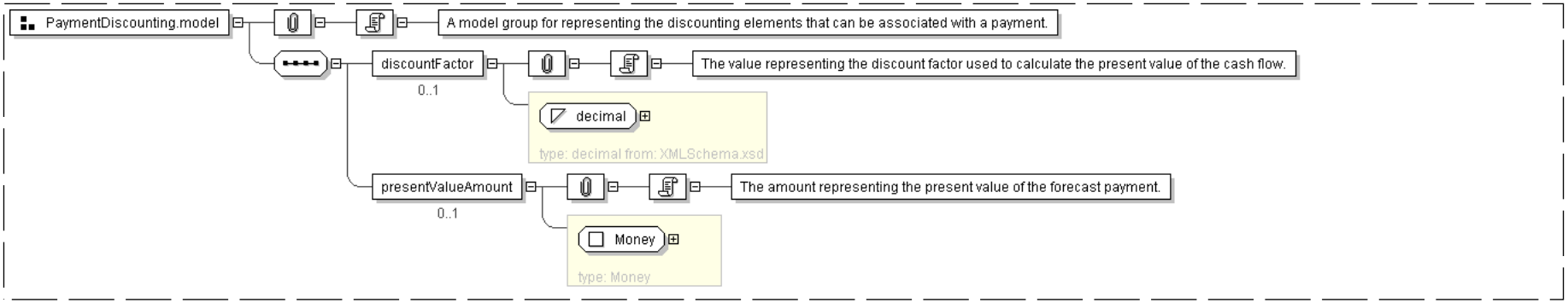
Model Group: PaymentDiscounting.model

Name	PaymentDiscounting.model
Documentation	A model group for representing the discounting elements that can be associated with a payment.

XML Instance Representation

```
<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.'
```

Diagram



Schema Component Representation

```
<xsd:group name="PaymentDiscounting.model">
   <xsd:sequence>
      <xsd:element name="discountFactor" type=" xsd:decimal " minOccurs="0"/>
      <xsd:element name="presentValueAmount" type=" Money " minOccurs="0"/>
   </xsd:sequence>
</xsd:group>
```

top

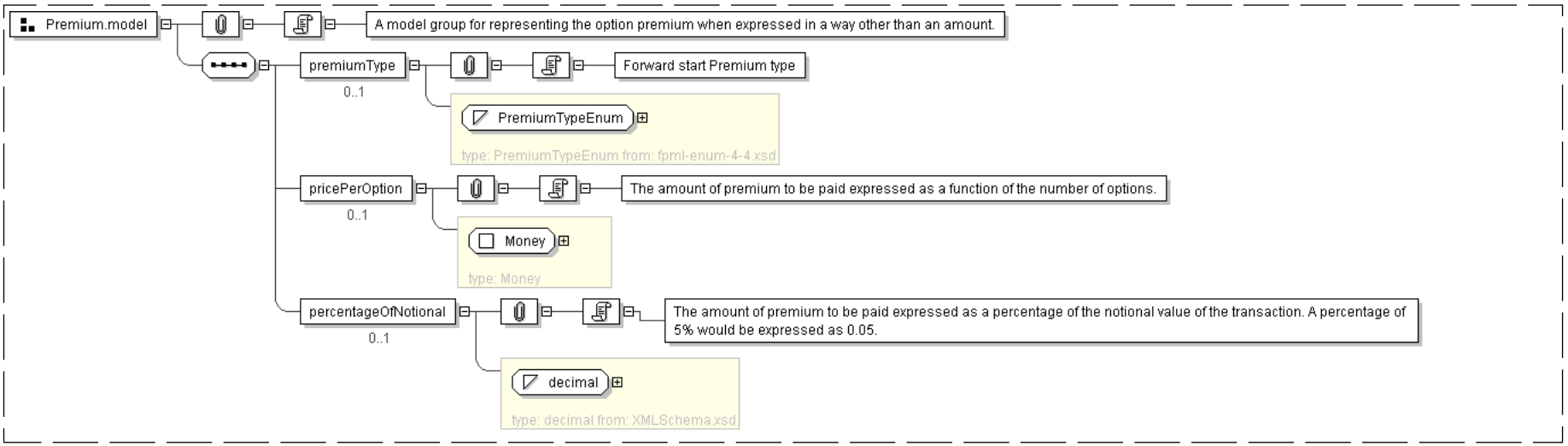
Model Group: Premium.model

Name	Premium.model
Documentation	A model group for representing the option premium when expressed in a way other than an amount.

XML Instance Representation

```
<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.'
```

Diagram



Schema Component Representation

```
<xsd:group name="Premium.model">
   <xsd:sequence>
      <xsd:element name="premiumType" type=" PremiumTypeEnum " minOccurs="0"/>
      <xsd:element name="pricePerOption" type=" Money " minOccurs="0"/>
      <xsd:element name="percentageOfNotional" type=" xsd:decimal " minOccurs="0"/>
   </xsd:sequence>
</xsd:group>
```

top

Model Group: Product.model

Name	Product.model
Used by (from the same schema document)	Complex Type **Product**

XML Instance Representation

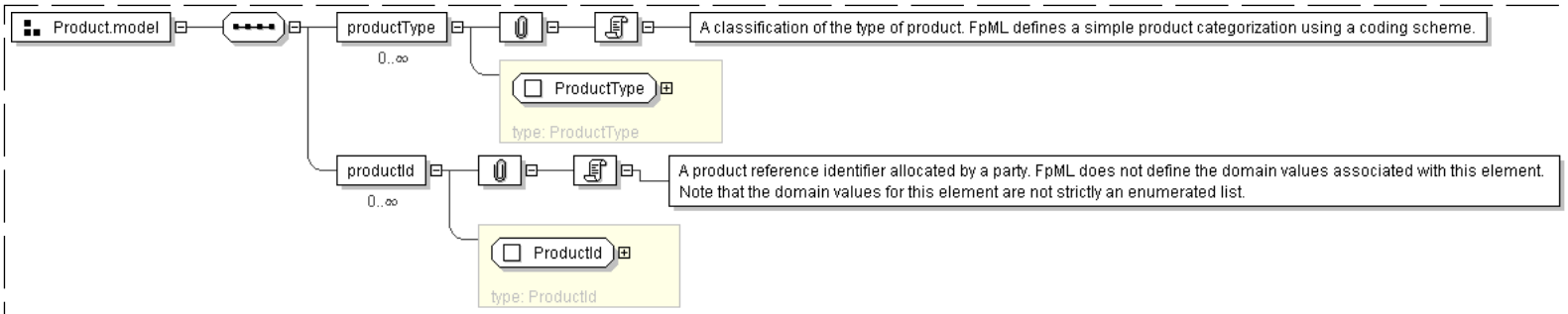
```
<productType> ProductType </productType> [0..*]
'A classification of the type of product. FpML defines a simple product categorization using
```

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.'

Diagram



Schema Component Representation

```
<xsd:group name="Product.model">
   <xsd:sequence>
      <xsd:element name="productType" type=" ProductType " minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="productId" type=" ProductId " minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:group>
```

top

Model Group: RoutingExplicitDetails.model

Name	RoutingExplicitDetails.model
Used by (from the same schema document)	Complex Type **RoutingExplicitDetails** , Complex Type **RoutingIdsAndExplicitDetails**

XML Instance Representation

<routingName> xsd:string </routingName> [1]

'A real name that is used to identify a party involved in the routing of a payment.'

<routingAddress> Address </routingAddress> [0..1]

'A physical postal address via which a payment can be routed.'

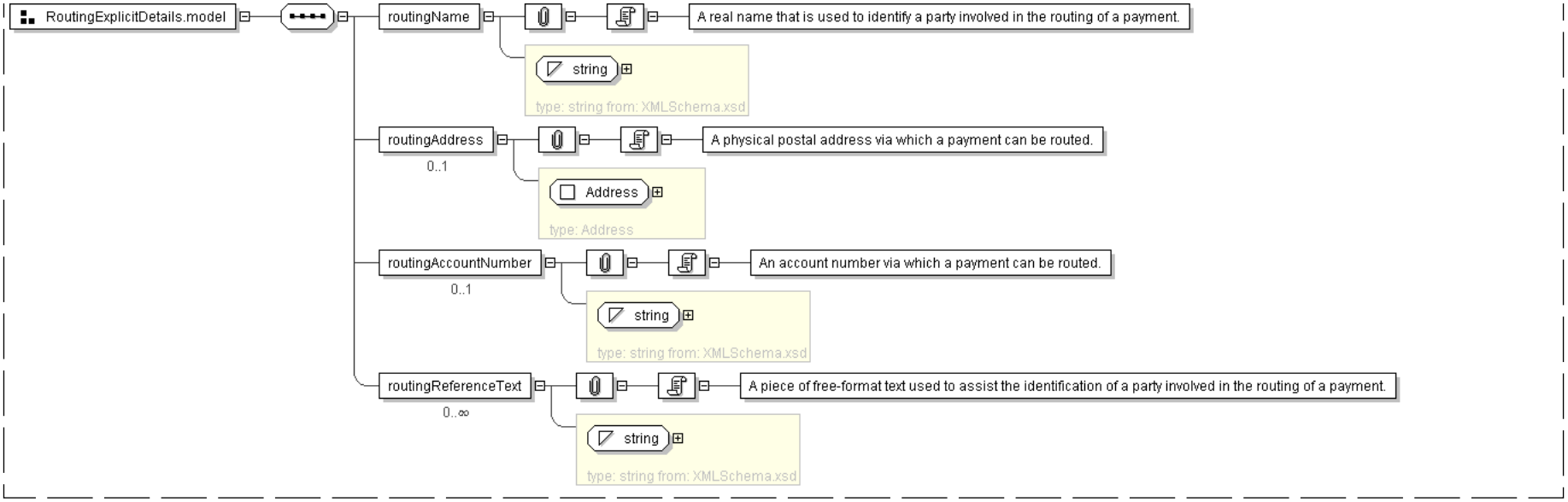
<routingAccountNumber> xsd:string </routingAccountNumber> [0..1]

'An account number via which a payment can be routed.'

<routingReferenceText> xsd:string </routingReferenceText> [0..*]

'A piece of free-format text used to assist the identification of a party involved in the routing of a payment.'

Diagram



Schema Component Representation

```
<xsd:group name="RoutingExplicitDetails.model">
   <xsd:sequence>
      <xsd:element name="routingName" type=" xsd:string "/>
      <xsd:element name="routingAddress" type=" Address " minOccurs="0"/>
      <xsd:element name="routingAccountNumber" type=" xsd:string " minOccurs="0"/>
      <xsd:element name="routingReferenceText" type=" xsd:string "
      minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:group>
```

top

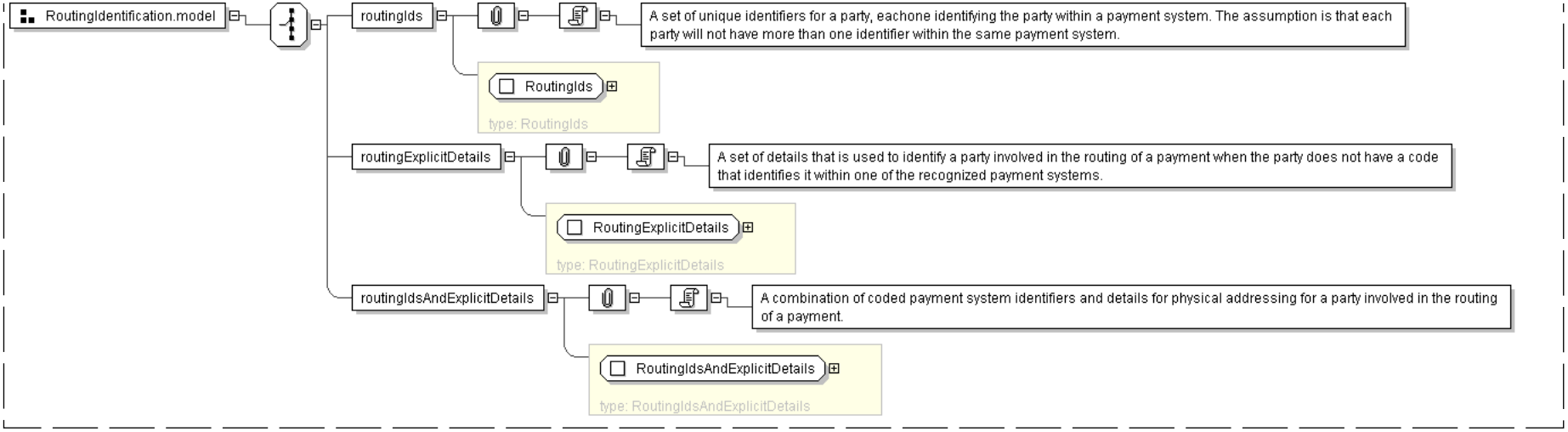
Model Group: RoutingIdentification.model

Name	RoutingIdentification.model
Used by (from the same schema document)	Complex Type **Beneficiary** , Complex Type **CorrespondentInformation** , Complex Type **IntermediaryInformation** , Complex Type **Routing**

XML Instance Representation

```
Start Choice [1]
   <routingIds> RoutingIds </routingIds> [1]
   'A set of unique identifiers for a party, eachone identifying the party within a
   payment system. The assumption is that each party will not have more than one identifier
   within the same payment system.'

   <routingExplicitDetails> RoutingExplicitDetails </routingExplicitDetails> [1]
   'A set of details that is used to identify a party involved in the routing of a payment
   when the party does not have a code that identifies it within one of the recognized
   payment systems.'

   <routingIdsAndExplicitDetails> RoutingIdsAndExplicitDetails </routingIdsAndExplicitDetails> [1]
   'A combination of coded payment system identifiers and details for physical addressing for
   a party involved in the routing of a payment.'

End Choice
```

Diagram



Schema Component Representation

```
<xsd:group name="RoutingIdentification.model">
   <xsd:choice>
      <xsd:element name="routingIds" type=" RoutingIds "/>
      <xsd:element name="routingExplicitDetails" type=" RoutingExplicitDetails "/>
      <xsd:element name="routingIdsAndExplicitDetails" type=" RoutingIdsAndExplicitDetails "/>
   </xsd:choice>
</xsd:group>
```

top

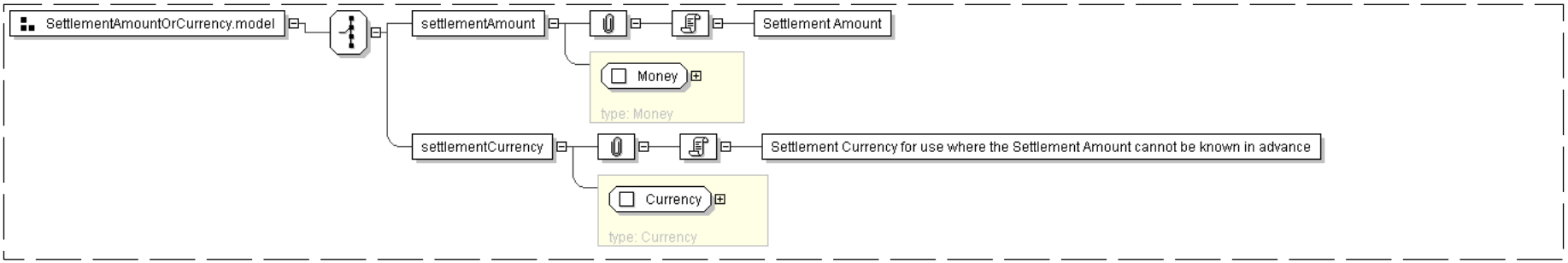
Model Group: SettlementAmountOrCurrency.model

Name	SettlementAmountOrCurrency.model

XML Instance Representation

```
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
```

Diagram



Schema Component Representation

```
<xsd:group name="SettlementAmountOrCurrency.model">
   <xsd:choice>
      <xsd:element name="settlementAmount" type=" Money "/>
      <xsd:element name="settlementCurrency" type=" Currency "/>
   </xsd:choice>
</xsd:group>
```

top

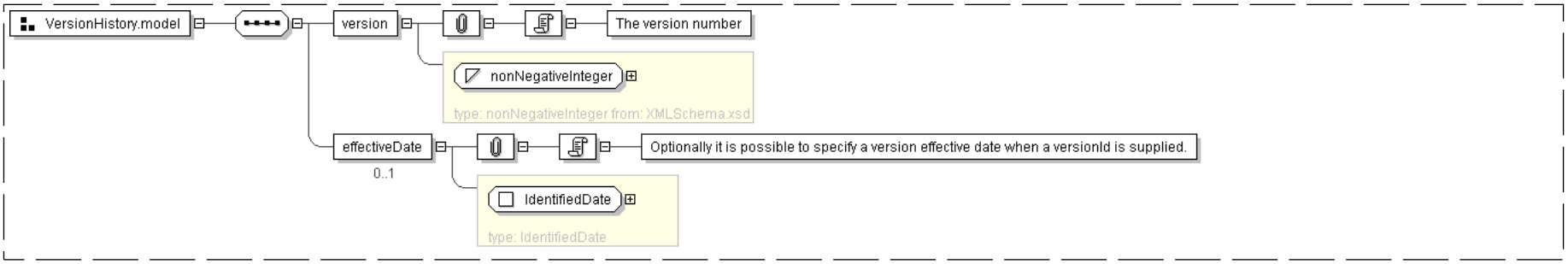
Model Group: VersionHistory.model

Name	VersionHistory.model

XML Instance Representation

```
<version> xsd:nonNegativeInteger </version> [1]
'The version number'

<effectiveDate> IdentifiedDate </effectiveDate> [0..1]
'Optionally it is possible to specify a version effective date when a versionId is supplied.'
```

Diagram



Schema Component Representation

```
<xsd:group name="VersionHistory.model">
   <xsd:sequence>
      <xsd:element name="version" type=" xsd:nonNegativeInteger "/>
      <xsd:element name="effectiveDate" type=" IdentifiedDate " minOccurs="0"/>
   </xsd:sequence>
</xsd:group>
```

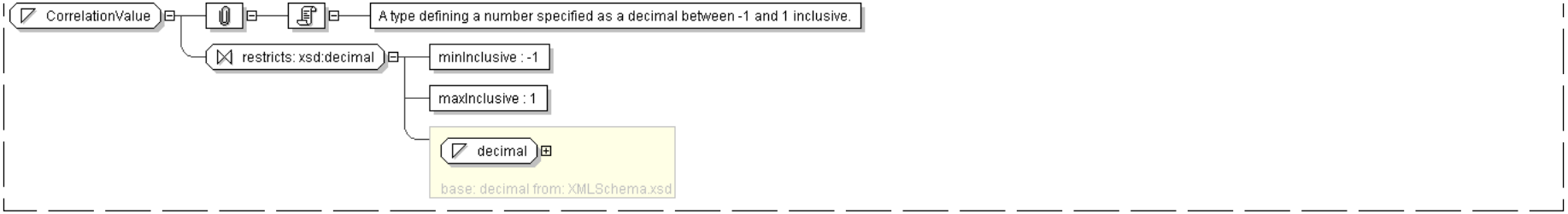
top

Simple Type: CorrelationValue

Super-types:	xsd:decimal < **CorrelationValue** (by restriction)
Sub-types:	None

Name	CorrelationValue
Content	• Base XSD Type: decimal • -1 <= *value* <= 1
Documentation	A type defining a number specified as a decimal between -1 and 1 inclusive.

Diagram



Schema Component Representation

```
<xsd:simpleType name="CorrelationValue">
   <xsd:restriction base=" xsd:decimal ">
      <xsd:minInclusive value="-1"/>
      <xsd:maxInclusive value="1"/>
   </xsd:restriction>
</xsd:simpleType>
```

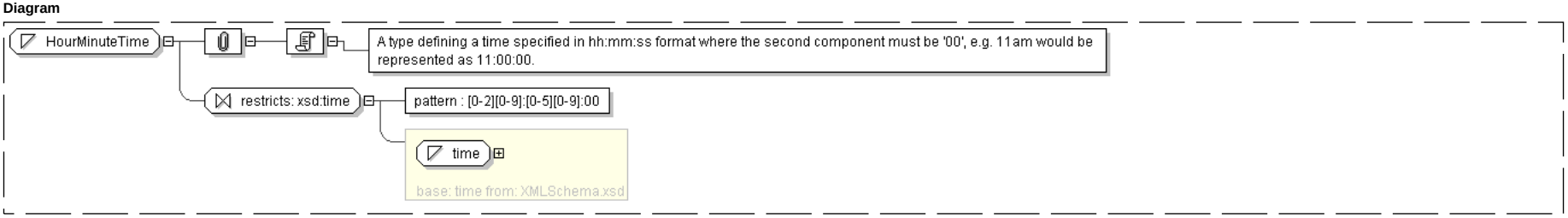
top

Simple Type: HourMinuteTime

Super-types:	xsd:time < **HourMinuteTime** (by restriction)
Sub-types:	None

Name	HourMinuteTime
Used by (from the same schema document)	Complex Type **BusinessCenterTime**
Content	• Base XSD Type: time • *pattern* = [0-2][0-9]:[0-5][0-9]:00
Documentation	A type defining a time specified in hh:mm:ss format where the second component must be '00', e.g. 11am would be represented as 11:00:00.

Diagram



Schema Component Representation

```
<xsd:simpleType name="HourMinuteTime">
   <xsd:restriction base=" xsd:time ">
      <xsd:pattern value="[0-2][0-9]:[0-5][0-9]:00"/>
   </xsd:restriction>
</xsd:simpleType>
```

top

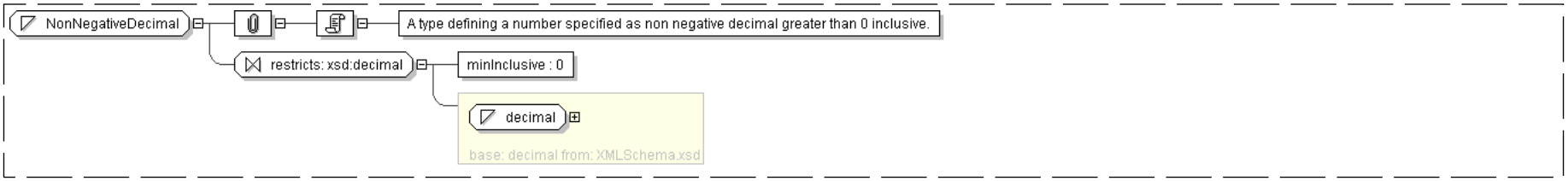
Simple Type: NonNegativeDecimal

Super-types:	xsd:decimal < **NonNegativeDecimal** (by restriction)
Sub-types:	None

Name	NonNegativeDecimal

Content	• Base XSD Type: decimal • *value* >= 0
Documentation	A type defining a number specified as non negative decimal greater than 0 inclusive.

Diagram



Schema Component Representation

```
<xsd:simpleType name="NonNegativeDecimal">
   <xsd:restriction base=" xsd:decimal ">
      <xsd:minInclusive value="0"/>
   </xsd:restriction>
</xsd:simpleType>
```

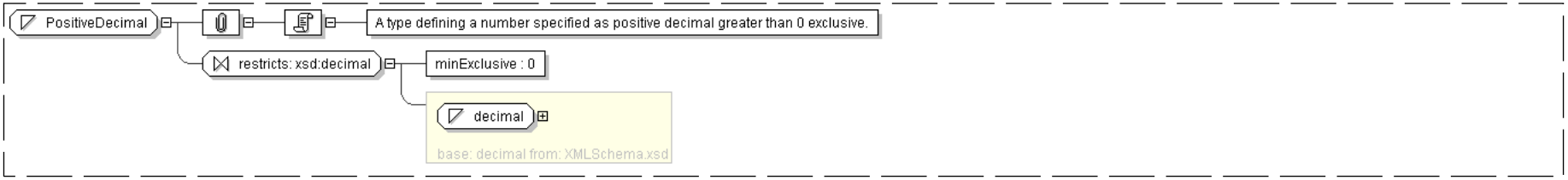
top

Simple Type: PositiveDecimal

Super-types:	xsd:decimal < **PositiveDecimal** (by restriction)
Sub-types:	None

Name	PositiveDecimal
Content	• Base XSD Type: decimal • *value* > 0
Documentation	A type defining a number specified as positive decimal greater than 0 exclusive.

Diagram



Schema Component Representation

```
<xsd:simpleType name="PositiveDecimal">
   <xsd:restriction base=" xsd:decimal ">
      <xsd:minExclusive value="0"/>
   </xsd:restriction>
</xsd:simpleType>
```

top

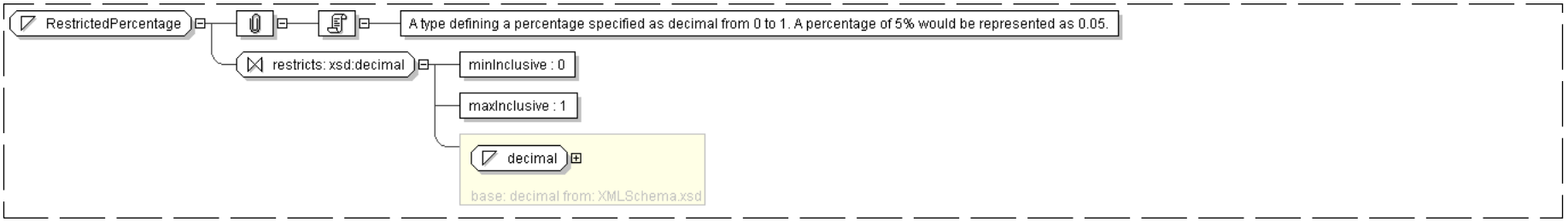
Simple Type: RestrictedPercentage

Super-types:	xsd:decimal < **RestrictedPercentage** (by restriction)
Sub-types:	None

Name	RestrictedPercentage

Content	• Base XSD Type: decimal • 0 <= *value* <= 1
Documentation	A type defining a percentage specified as decimal from 0 to 1. A percentage of 5% would be represented as 0.05.

Diagram



Schema Component Representation

```
<xsd:simpleType name="RestrictedPercentage">
   <xsd:restriction base=" xsd:decimal ">
      <xsd:minInclusive value="0"/>
      <xsd:maxInclusive value="1"/>
   </xsd:restriction>
</xsd:simpleType>
```

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.