FpML Financial product Markup Language

Working Draft 17 April 2002

Version: 3.0

This Version:

http://www.fpml.org/spec/2002/wd-fpml-3-0-2002-04-17

Latest Version:

http://www.fpml.org/spec/fpml-3-0

Previous Version:

http://www.fpml.org/spec/2002/wd-fpml-3-0-2002-01-30

Table Of Contents

1 STATUS OF THIS DOCUMENT:

This is the FpML Version 3.0 Working Draft for review by the public and by FpML members and working groups. It is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to use FpML Working Drafts as reference material or to cite them as other than "work in progress". There will be a subsequent release of this working draft to include Equity Derivative Products. This is work in progress and does not imply endorsement by the FpML.

The Interest Rate Derivatives (IRD) and FX Products Working Groups encourage reviewing organizations to provide feedback as early as possible. Comments on this document should be sent via e-mail to fpml-issues@yahoogroups.com. An archive of the comments is available at

Public discussion of FpML takes place on the FpML Discussion List at:

http://groups.yahoo.com/group/fpml-discuss

A list of current FpML Recommendations and other technical documents can be found at

http://www.fpml.org/spec

While implementation experience reports are welcomed, the IRD and FX Products Working Groups will not allow early implementation to constrain its ability to make changes to this specification prior to final release.

This document has been produced as part of the FpML Version 3.0 activity and is part of the Standards Approval Process.

2 OTHER DOCUMENTATION

Data Dictionary
Scheme Definitions

2.1 Component Definitions

Main Components
Shared Components
Equity Derivative Components
FX Components
Interest Rate Components

2.2 Example XML

Equity Derivatives
FX
Interest Rate Derivatives

3 WORKING GROUP MEMBERS AND ACKNOWLEDGEMENTS:

This document was produced by the following working groups:

3.1 Equity Derivatives Working Group

- Andrew Parry (Deutsche Bank), chair
- Jonathan Smart (Deutsche Bank), deputy chair
- Jean Bouez (HSBC)
- Simon Brooke (Deutsche)
- Gordon McDermid (CSFB)
- Simon Dewberry (SSMB)
- Tracy Ruparell (MSDW)
- Kathy Wallis (Goldman Sachs)
- Ben Wydell (Barclays Capital)

3.2 FX Working Group

- Rick Schumacher (Wall Street Systems), chair
- Lee Buck (Morgan Stanley)
- Fred Burge (JP Morgan Chase)
- Marcello Davanzo (KPMG)
- Alexander Ernst (RiskTrak Financial)
- Gary Goldberg (FXPress)
- Rahul Gupta (FXAll)
- Lee Haines (Reuters)
- Richard Hall (Evolution)
- Antoine Hurstel (BNP Paribas)
- Bruce Kenney (Mizuho Capital Markets)
- Dan Kolner (UBS Warburg)
- Anna Lukasiak (Goldman Sachs)
- Tim Madeley (Cognotec)
- Francoise Massin (SWIFT)
- Donna McCollum (SunGard Trading Systems)
- Ned Micelli (Reuters)
- Justin Oglethorpe (Citigroup)
- Hugues Planzol (BNP Paribas)
- Frank Smith (Atriax)
- Gavin Smith (RCP Consultants)
- Bill Specht (Currenex)
- Tony Spraggs (Reuters)
- Viral Tolat (Integral)
- Jayesh Vira (SunGard Trading Systems)

3.3 IRD Products Working Group

- Steven Lord (UBS Warburg), chair
- Andrew Addison (Merrill Lynch)

- Ariane Athalie (BNP Paribas)
- Owen Bugge (Blackbird)
- Marisol Collazo (Mizuho Capital Markets)
- Marie-Paule Dumont (S.W.I.F.T.)
- Alexander Ernst (RiskTrak Financial)
- Tom Fahy (Goldman Sachs)
- Doug Gallager (Reval.com)
- Mark Golding (JPMorgan)
- David Gorans (BNP Paribas)
- Guy Gurden (SwapsWire)
- Paul Hoskins (Barclays Capital)
- Saleem Huda (Algorithmics)
- Vlad Iordanov (FinTrack Systems)
- Keri Jackson (independent)
- Sathy Kovvali (Citigroup)
- Philippe Negri (SunGard Trading And Risk Systems)
- Michael North (Reuters)
- Henry Teng (UBS Warburg)
- Ian Thomas(Credit Suisse First Boston)
- Patrick Treanor (Wall Street Systems)
- Olga Urrutia (S.W.I.F.T.)
- James Williams (Deutsche Bank)
- Barry Witkow (Treasury Connect)
- Chuck Witter (Bank of America)

4 INTRODUCTION

The Financial Products Markup Language (FpML) is a protocol enabling e-commerce activities in the field of financial derivatives. The development of the standard, controlled by FpML, will ultimately allow the electronic integration of a range of services, from electronic trading and confirmations to portfolio specification for risk analysis. All types of over-the-counter (OTC) derivatives will, over time, be incorporated into the standard. FpML 3.0 covers FX and Interest Rate Derivatives.

FpML is an application of XML, an internet standard language for describing data shared between computer applications.

5 SCOPE

The scope of FpML 3.0 includes the FpML 2.0 work done by the IRD Products Working Group together with work done on FX and Equity Derivatives products.

5.1 Equity Derivatives Scope

The EQD Products Working Group will extend the current FpML standard and will deliver a specification for the following Equity Derivative Products:

- Vanilla Put and Call Options
- Underlying Products are Single Stock or Indices
- European or American Exercise Style
- Cash or Physical Settlement

The Working Group will provide a technical expression of the ISDA definitions of the products which are in scope. This requires us to support operational features such as contact information and governing documentation which are present in current documents defined by ISDA standards

Outside the scope of the working group:

- Equity Derivative Products not explicitly in scope above
- Portfolio support. This is defined by FpML Taskforce
- Messaging support. The definition of such support is expected to be covered by the work of other FpML Working Groups

The Working Group will focus on providing deliverables of high value to member firms. We will prepare and conduct surveys amongst member firms in order to determine what they consider to be of high value. This information will guide our approach to future phases

Deliverables will be in the form of DTDs, sample XML formed from ISDA documentation, and associated supporting documentation

5.2 FX Scope

The FX Products Working Group will extend the current FpML standard and will deliver a specification for the following FX products:

- FX Spot
- FX Forward (including non-deliverable forwards, or NDFs)
- FX Swap
- FX OTC Options (at a minimum, European and American options)
- Option Strategies (multiple simple options)

The Working Group will evaluate the Citigroup / UBS Warburg proposals and other proposals that are deemed appropriate in order to formulate the deliverable specification. In addition to the specification itself, tasks of the Working Group will include:

- In conjunction with the IRD and Equity Derivatives Product Working Groups, define a packaging of the DTDs to allow multiple asset classes to coexist with the specification;
- Ensure that the resulting DTDs and coding schemes are published to the FpML web site once approved.

The Working Group will assess the current climate within the Foreign Exchange market to determine whether the definition of trade content is acceptable. There are various opinions being expressed that may lead to working closely with the Business Messaging and Architecture (BMA) Working Group to determine if more of the "life cycle" of a trade must be addressed (i.e., trade negotiation, trade settlement).

The Working Group will prepare and conduct a survey amongst member firms (plus potentially other institutions) to determine the types and volumes of various styles of "exotic" FX OTC options as well as other

types of FX instruments (e.g., fixing order, time options). The results of the survey will be presented to the FpML Board to enable them to determine the priority for further FpML FX product coverage. An external firm may be engaged to help with the logistics of conducting the survey and to address confidentiality concerns surrounding the data.

The group deliverables will be in the form of DTDs, sample XML and associated documentation.

5.3 IRD Scope

The scope of the IRD Products Working Group, with respect to extending the FpML 1.0 product definitions, is to complete definitions for the following new products and features:

- Interest Rate Cap
- Interest Rate Floor
- Interest Rate Swaption (European, Bermudan and American Styles; Cash and Physical Settlement)
- Extendible and Cancelable Interest Rate Swap Provisions
- Mandatory and Optional Early Termination Provisions for Interest Rate Swaps
- FX Resetable Cross-Currency Swap

Current capabilities of the FpML Version 1.0 specification to support Basis Swaps will also be reviewed.

Outside the scope of the Working Group are the following:

- Definition of business processes that might result in the trade content defined here being transmitted between parties. The definition of these processes and resulting messages is expected to be covered by the work of other FpML Working Groups.
- Definition of reference data related to the counterparty such as settlement instructions, location and contact details. It was agreed that this static data did not belong in each instance of an FpML document and would most likely be stored in central or distributed repositories and referenced from within the document. Specification or design of such repositories is also beyond the scope of the Working Group. Since identification of parties is an essential requirement of a trade content definition, the FpML Consortium has decided, to continue in this release, to identify parties using the ISO standard bank identifier code (BIC). S.W.I.F.T. is the designated registration authority for the assignment of BIC codes. Although this is the recommended identification scheme for parties wishing to use FpML for inter-firm communication , the FpML architecture supports the use and identification of alternative coding schemes through the Schemes mechanism.

5.4 Taskforce Scope

An FpML taskforce was formed from members of each of the above working groups together with members from the Architecture working group to address issues that cut across all asset classes. The main areas this group looked at were:

- Portfolios
- Product Strategy
- Messaging Framework

Their recommendations on the first two are included here. The work done on Messaging Framework has been published as a Technical Note and a seperate working group will be formed to carry this work forward to a Working Draft.

5.5 Architecture Framework

The Products Working Groups have developed FpML 3.0 within the FpML Architecture Version 1.0 framework defined by the Architecture Working Group. Their recommendations covered:

- XML tools for editing and parsing
- XML namespace usage within FpML
- FpML versioning methodology
- FpML content model - a new style for representing the FpML Document Type Definition (DTD)
- FpML referencing methodology, including guidelines for referencing coding schemes.

6 INCOMPATIBLE CHANGES TO FpML 2.0

The following changes have been introduced in FpML 3.0 Working Draft which are imcompatible with FpML 2.0

6.1 Party Element Move

The party element has been moved from beneath the trade element to directly within the FpML root element. This move was made so that if a document has mutliple trades the party element is not repeated. This is not only for practical reasons of reducing document size but also because the party element has an id attribute, which has to be unique, which means it cannot be repeated within a document.

7 REMOVAL OF SCHEME DEFAULTS

The FpML Working Groups are considering removing the scheme default attributes from the FpML root element and changing the scheme attributes from being #IMPLIED to #REQUIRED. Comments and feedback on this idea are encouraged.

This change is proposed because where scheme default attributes are used the meaning of a given node is dependent on the document it is in. This means that if part of an FpML document it copied from one document to another then it's meaning may change if the scheme default attribute is different

8 PRODUCT ARCHITECTURE OVERVIEW

8.1 Introduction

FpML incorporates a significant level of structure, rather than being a 'flat' representation of data. This structuring is achieved through the grouping of related elements describing particular features of a trade into components. Components can both contain, and be contained by, other components.

An alternative approach would have been to collect all the required elements in a single large component representing a product or trade. A flat structure of this kind would capture all the relevant information concisely but would also constrain the model in two important respects, namely, ease of implementation and extensibility.

Grouping related elements into components makes it easier to validate that the model is correct, that it is complete and that it doesn't contain redundancy. This is true, both from the perspective of readability to the human eye, and also from the perspective of processing services. Processing services that do not need all the information in a trade definition can isolate components and be sure that the complete set of elements required, and only the elements required, is available for the particular process in hand.

Components additionally serve as the building blocks for a flexible and extensible model. Generally speaking, the complexity of financial products is a result of combining a few simple ideas in a variety of different ways. The component structure supports a trade content definition that is flexible enough to represent the wide variation of features found in traded financial instruments.

It should be noted that the application of the guiding principles of extensibility and ease of use has resulted in a different approach with regard to the forward rate agreement. Because this product is straightforward, commoditized and unlikely to develop further, the advantage to be gained from the extensive use of components is outweighed by the concision of a single component.

8.2 Component Framework

The optimum level of granularity is important to FpML. FpML separates the elements which collectively describe a feature of a product or trade into a separate component with each component serving a particular semantic purpose. Every grouping of elements in FpML is regarded as a component and each component is regarded as a container for the elements that describe that component. In the majority of cases each component will contain a mixture of other components and primitive elements, e.g. a date or string, that collectively describe the features of the component. Components are typically represented in the FpML Document Type Definition (DTD) as XML entities.

Generally speaking, the lower level a component is, the more re-usable it will be. FpML makes use of a number of primitive entity components that describe the basic building blocks of financial products, for example, FpML_Money, FpML_AdjustableDate, FpML_BusinessCenters, FpML_Interval, FpML_BusinessDayAdjustments etc. These primitive components are re-used in different business contexts.

Primitive components are contained in higher level components that describe the features of particular products. For this reason these higher level components will tend not to be re-usable to the same extent. Examples within the definition of swapStream are the components required to construct schedules of dates such as calculationPeriodDates, resetDates and paymentDates. However, it should not be inferred from this that any fundamental distinction is drawn between components in usage or structure.
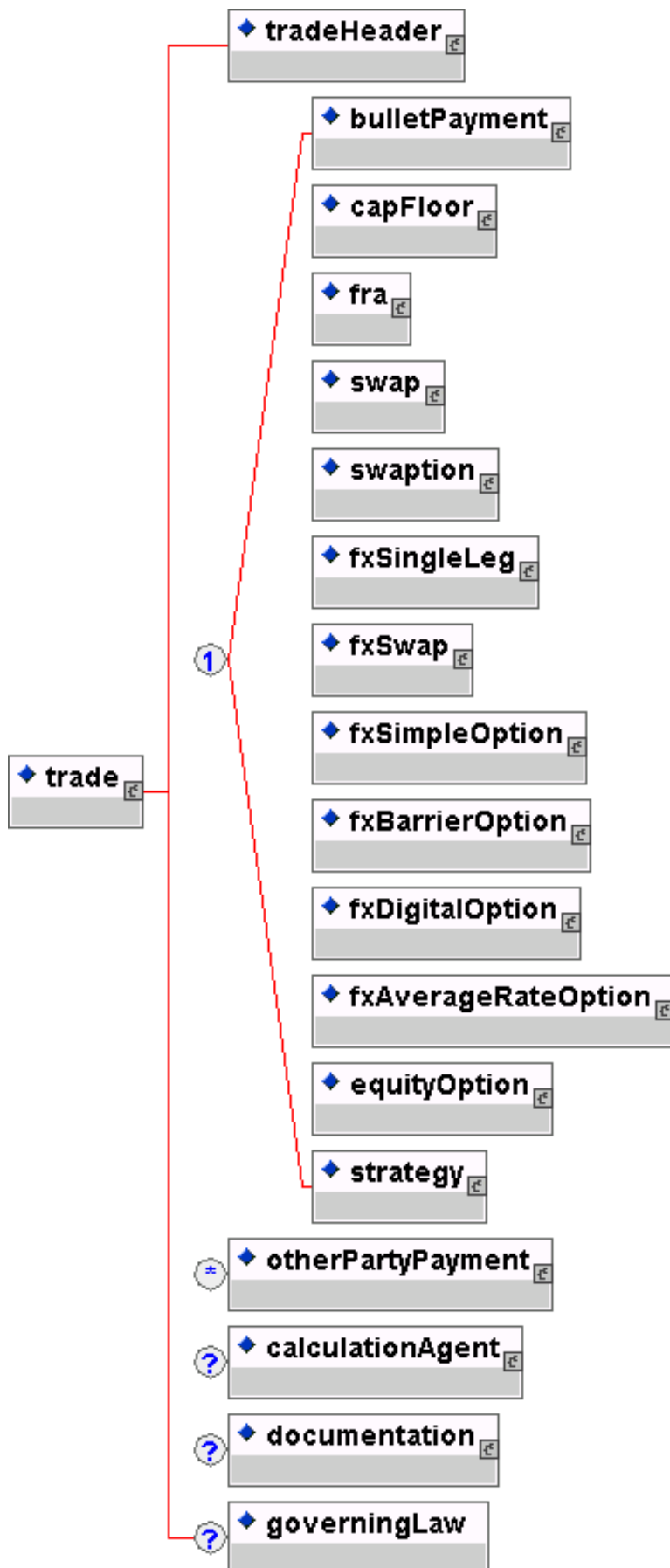
8.3 Overview of Core Components

The root element contains three elements, trades, portfolios and parties. Portfolios contain only trade references, if the trade itself needs to be included in the document then the trades can be included within the root element.
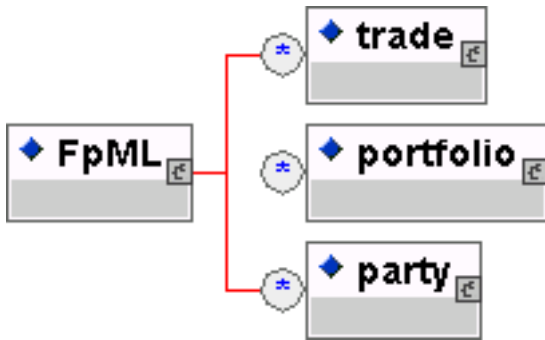
8.3.1 The Trade Component

The trade is the top-level component within the root element FpML. A trade is an agreement between two parties to enter into a financial contract and the trade component in FpML contains the economic information necessary to execute and confirm that trade. A trade contains four components: tradeHeader, product (an abstract concept), party (two or more instances) and otherPartyPayment (zero or more instances).

```
trade ─┬─ ◆ tradeHeader
       │
       ├─(1)─┬─ ◆ bulletPayment
       │     ├─ ◆ capFloor
       │     ├─ ◆ fra
       │     ├─ ◆ swap
       │     ├─ ◆ swaption
       │     ├─ ◆ fxSingleLeg
       │     ├─ ◆ fxSwap
       │     ├─ ◆ fxSimpleOption
       │     ├─ ◆ fxBarrierOption
       │     ├─ ◆ fxDigitalOption
       │     ├─ ◆ fxAverageRateOption
       │     ├─ ◆ equityOption
       │     └─ ◆ strategy
       │
       ├─(*)─ ◆ otherPartyPayment
       ├─(?)─ ◆ calculationAgent
       ├─(?)─ ◆ documentation
       └─(?)─ ◆ governingLaw
```

8.3.1.1 tradeHeader

The information within tradeHeader will be common across all types of trade regardless of product. In FpML 2.0 this contains the trade date, party trade identifiers and any calculation agent references.
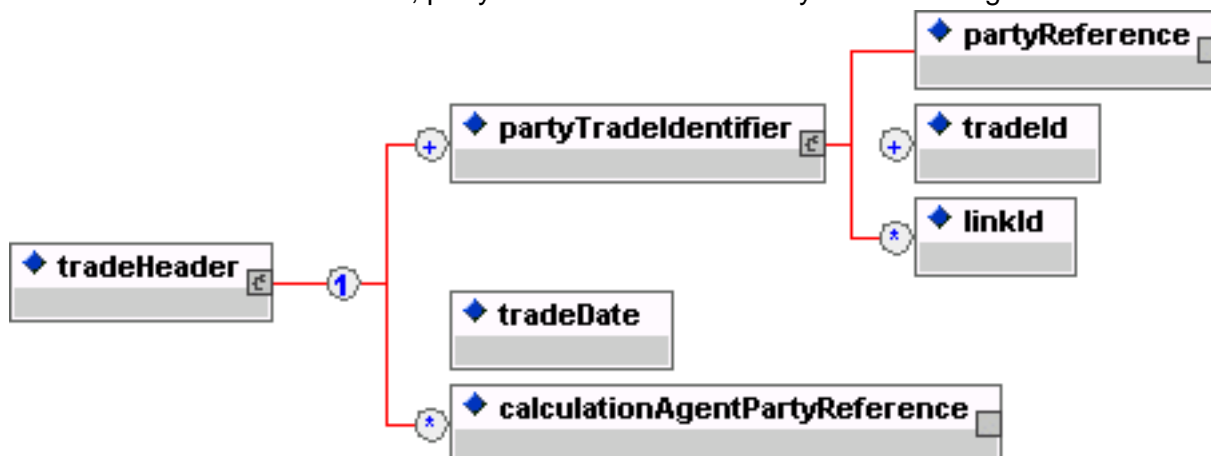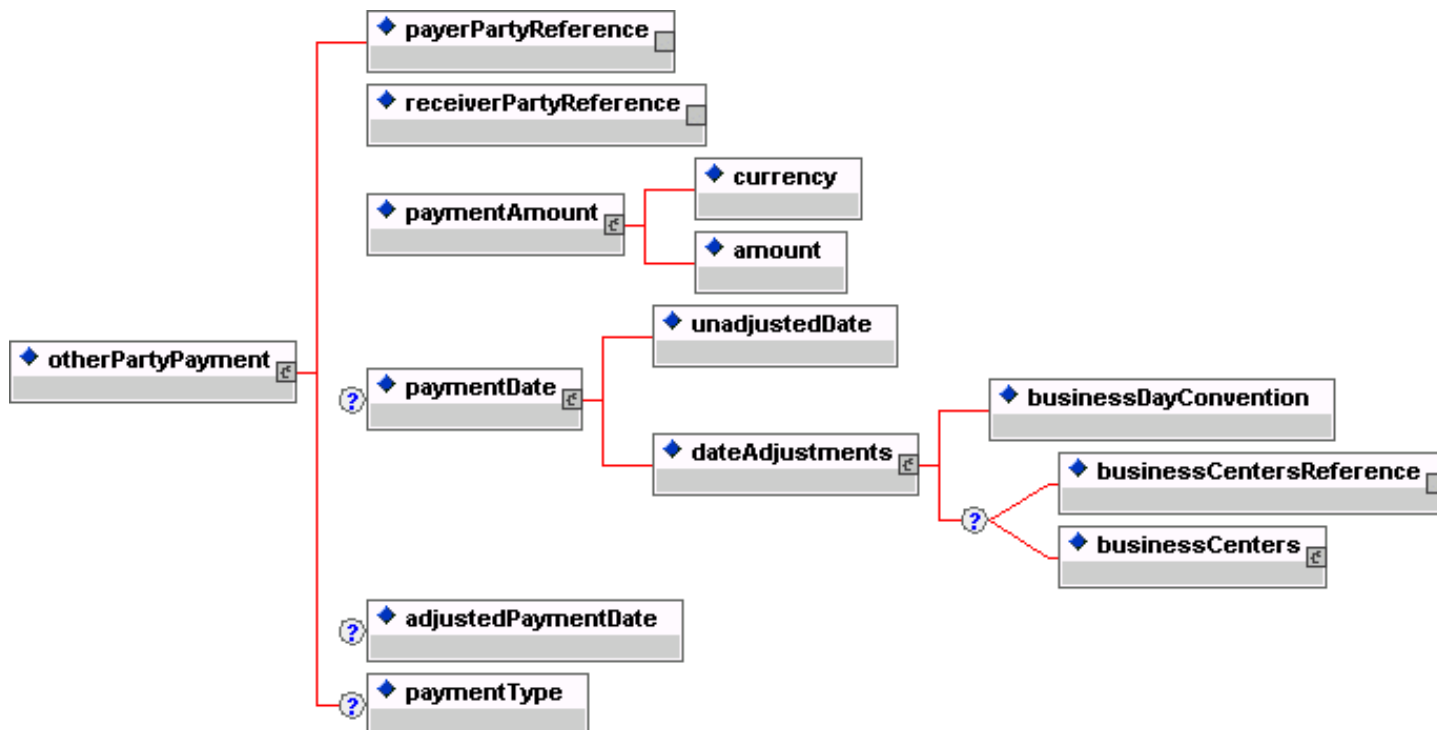

8.3.1.2 product

Product is an abstract concept in FpML and an actual product element is not used. Instead, one of the FpML products will appear directly under trade.

8.3.1.3 otherPartyPayment

This component contains additional payments such as brokerage paid to third parties which are not part of the economics of a trade itself.


8.3.2 The Portfolio Component

The portfolio component specifies a set of trades as a list of tradeIds and a list of sub portfolios. Portfolios can be composed of other portfolios using a composition pattern. By using the tradeId to identify the trade the standard allows for portfolios to be sent around without the full trade record.
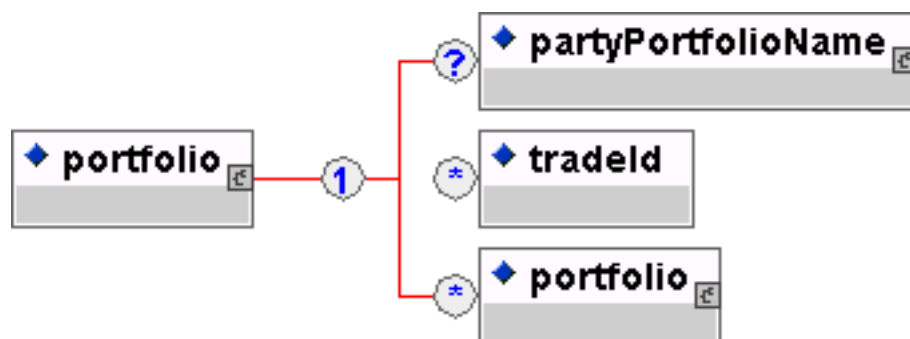
8.3.3 The Party Component

The party component holds information about a party involved any of the trades or portfolios included in the document. The parties involved will be the principals to a trade and potentially additional third parties such as a broker. For this release, this component is restricted to party identification.

It should be noted that an FpML document is not 'written' from the perspective of one particular party, i.e. it is symmetrical with respect to the principal parties. The particular role that a party plays in the trade, e.g. buyer, seller, stream payer/receiver, fee payer/receiver, is modeled via the use of references from the component where the role is identified to the party component.
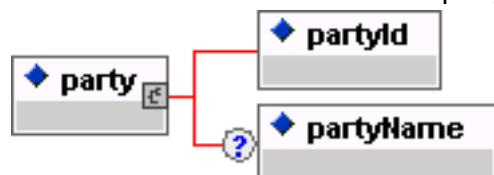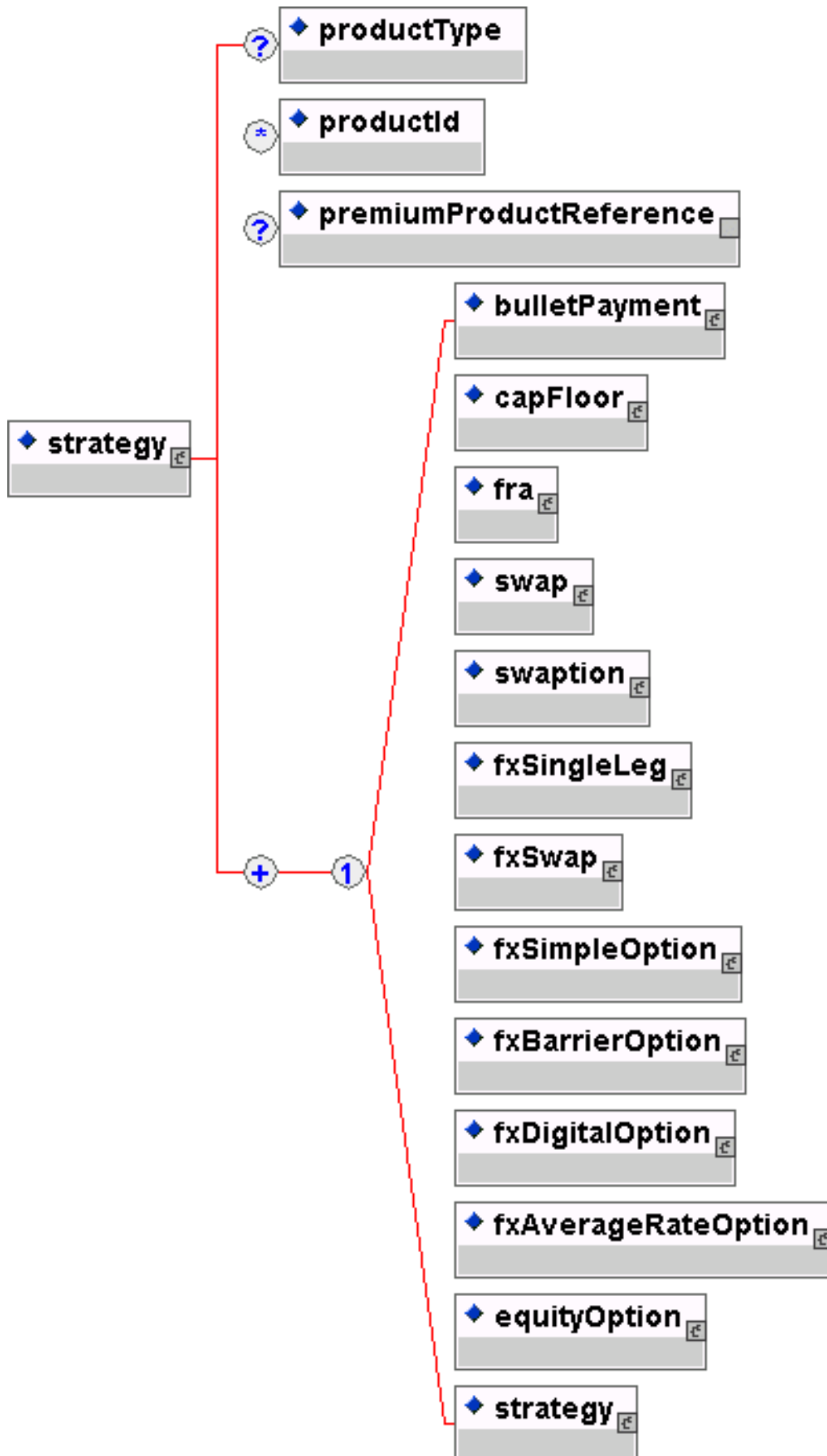

8.3.4 The Product Component

The product component specifies the financial instrument being traded. This component captures the economic details of the trade. Because of the complexity of the OTC Interest Rate Derivatives domain that FpML 2.0 covers, composing these products from various building blocks is a key aspect of the design approach.

FpML 1.0 focused on the instrument definitions for interest rate swaps (including cross currency swaps) and forward rate agreements. For that initial release, a trade was restricted to containing only a single product definition. In FpML 2.0 product strategy has been introduced which allows more than one product within a trade. In FpML 2.0 the instrument definition has been extended to include options.

8.3.5 The Strategy Component

This component allows the structuring of trade by combining any number of products within a strategy. This component makes use of a composition pattern since strategy itself is a product. This means that strategies can themselves contain strategies.

strategy

- ? **productType**
- ≠ **productId**
- ? **premiumProductReference**
- + 1
  - **bulletPayment**
  - **capFloor**
  - **fra**
  - **swap**
  - **swaption**
  - **fxSingleLeg**
  - **fxSwap**
  - **fxSimpleOption**
  - **fxBarrierOption**
  - **fxDigitalOption**
  - **fxAverageRateOption**
  - **equityOption**
  - **strategy**

8.4 Coding Schemes

A necessary feature of a portable data standard is both an agreed set of elements and an agreed set of permissible values (the value domain) for those elements. An FpML document exchanged between two parties would not be mutually understandable if either or both of the parties used internal or proprietary coding schemes to populate elements.

Reference data can originate from various sources and the range of permitted values may be more or less extensive. The dayCountFraction is an example of an element with a limited set of permissible values with well-defined meanings. The range of permitted values comes from several sources including ISDA and AFB definitions. However, the currency element is an example of where the list of permitted values is more extensive and the coding scheme reference is to a well-known standard, in this case ISO 4217.

In FpML 1.0 the recommended domain for party identification is a valid bank identifier code (BIC). The BIC is an ISO standard, ISO 9362. S.W.I.F.T. is the designated registration authority for the assignment of BIC codes.

One possible means of identifying value domains would have been to include the domain of permitted values within the DTD. This solution has been rejected for two reasons. Firstly, in many cases the scope of permitted values is extensive, most obviously with party identifiers, and this would make the standard unnecessarily bulky. Secondly, although there are varying degrees of stability, all value domains are subject to change and including them in the DTD would have necessitated a new version of FpML each time a value domain changed.

For these reasons, FpML uses Schemes to identify the permitted values for an element. In each case, the reference Scheme will be identified by a URI. The URI will either identify a well-known external standard such as ISO 4217, or where no well-established standard exists, an FpML standard. FpML 1.0 includes provision for a default Scheme and the facility to override the default Scheme at an element level. In both cases, no values are included for the URI in the DTD in order to avoid coding either particular Schemes, or particular versions of Schemes, into FpML. For the same reason, the URI quoted in an FpML document for a Scheme that is FpML controlled will include a date and version in order to identify the particular version referenced.

It should be noted that the Scheme approach adopted by FpML does not allow validation of the values within the DTD. It will be the responsibility of the applications that implement FpML to validate that the contents of an element conform to the specified Scheme.

For further details on the architectural framework behind Schemes, refer to the FpML Architecture Version 1.0 document.

9 CHARACTER ENCODING AND CHARACTER REPERTOIRE

9.1 Character Encoding

Producers of FpML documents intended for interchange with other parties must encode such documents using either UTF-8 or UTF-16. Consumers of FpML documents must be able to process documents encoded using UTF-8, as well as documents encoded using UTF-16. For more information, see

9.2 Character Repertoire

FpML element content, as well as values of the FpML id and href attributes, may use any valid XML characters. For more information, see

10 INTEREST RATE DERIVATIVE PRODUCT ARCHITECTURE

10.1 Interest Rate Swap

A swap component contains one or more instances of the swapStream component, zero or more instances of the additionalPayment component together with an optional cancelableProvision component,an optional extendibleProvision component and an optional earlyTerminationProvision component. A swapStream contains the elements required to define an individual swap leg.

Within an FpML swap there is no concept of a swap header. Details of payment flows are defined within swapStreams which each contain their own independent parameters. There can also be additionalPayment elements that contain fees. The additionalPayment component is identical to the otherPartyPayment component shown above.

FpML 2.0 adds option related features. These include cancelable, extendible swaps and early termination provisions. Combining these together with swaptions into a single component was considered but rejected in favour of identifying the different option types with their own components. This provided more clarity and allowed for easier combination of the different options into a single trade. As such a swap can contain a cancelableProvision, extendibleProvision and an earlyTerminationProvision. All these components are very similar (and similar to the swaption component), re-use is achieved by using shared entities within each of the components.
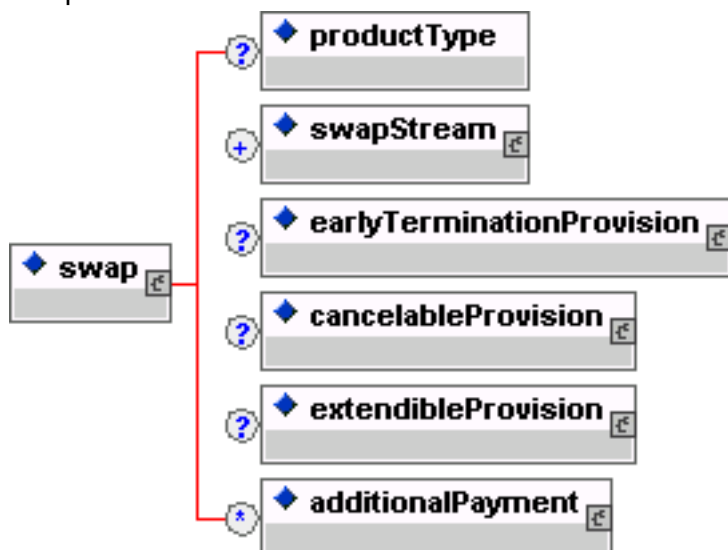
FpML supports two representations of a swap stream; a parametric representation, and a cashflows representation. The parametric representation is designed to capture all the economic information required regarding dates, amounts and rates to allow trade execution and confirmation. The parametric representation is mandatory. The cashflows representation specifies an optional additional description of the same stream. The main purpose of this is to allow the inclusion of adjusted dates within an FpML representation of a trade. It can also be used to represent adhoc trades not achievable by easy manipulation of the parameters of a stream (i.e. by changing the adjusted dates). This would lead to the cashflows not matching those generated by the parameters (see more discussion later) and would also render the representation of the trade unsuitable for a confirmation. The spirit of FpML is that such manipulation of cashflows would be achieved by splitting a single stream into a number of streams though it is recognized that this may be impractical in some systems.

The cashflows representation is not self contained as it relies on certain information contained within the stream's parametric definition. The elements required from the parametric definition to complete the cashflows representation are:

The following elements and their sub-elements within the calculationPeriodAmount element:

- floatingRateIndex
- indexTenor
- rateTreatment
- finalRateRounding

- averagingMethod
- negativeInterestRateTreatment
- dayCountFraction
- discounting
- compoundingMethod.

The following elements and their sub-elements within the stubCalculationPeriodAmount element:

- floatingRateIndex
- indexTenor.

The inclusion of the cashflows representation is intended to support Application integration. For example, a financial institution may have one application that captures trade parameters and constructs the trade schedules and then publishes the result for use by other applications. In this case it may be either undesirable, or impossible, for each of the subscribing applications to store and calculate schedules.

The flexibility of the cashflows representation also allows payment and calculation schedules which can not be fully represented by the parametric description. If this situation arises, the mandatory parametric data should still be included in the document and the flag cashflowsMatchParameters should contain the value false to indicate that it is not possible to generate the cashflows from this parametric data. The setting of this flag to true means that the cashflows can be regenerated at any time without loss of information.

Parties wishing to take advantage of the facility for specifying cashflows which are inconsistent with the parametric representation will need to specify additional rules for how the parametric representation should be processed. This applies to both the creation of the parametric data as well as its interpretation.

The cashflows representation specifies adjusted dates, that is, dates that have already been adjusted for the relevant business day convention using the relevant set of business day calendars (lists of valid business days for each business center). The FpML standard does not specify the source of these business day calendars. This may lead applications to generate differing cashflow representations from the same parametric representation if they use different business day calendars. The use of adjusted dates also produces schedules that are only valid at a particular instance of time. Additional holidays for a business center may subsequently be introduced that would result in changes to the adjusted dates, which would not be reflected in the cashflows representation.

Analogous to cashflows being used to represent adjusted dates, with the addition of options it was important to be able to represented the adjusted dates associated with an option. Thus, where appropriate, a component includes an optional element to represent a schedule of adjusted dates for the option. Such a schedule would include details of adjusted dates such as adjusted exercise dates and cash settlement dates.

In general, an interest rate swap will be a swap with a fixed leg and a floating leg, two floating legs, or two fixed legs. However, certain types of trades may contain more than two legs. FpML does not restrict the number of legs that may be defined. From a modeling perspective, FpML does not distinguish between a swap leg referencing a fixed rate and a swap leg referencing a floating rate, the difference being indicated by the existence, for example, of the resetDates component in a floating rate leg.

The structure of a swapStream is shown diagrammatically below:
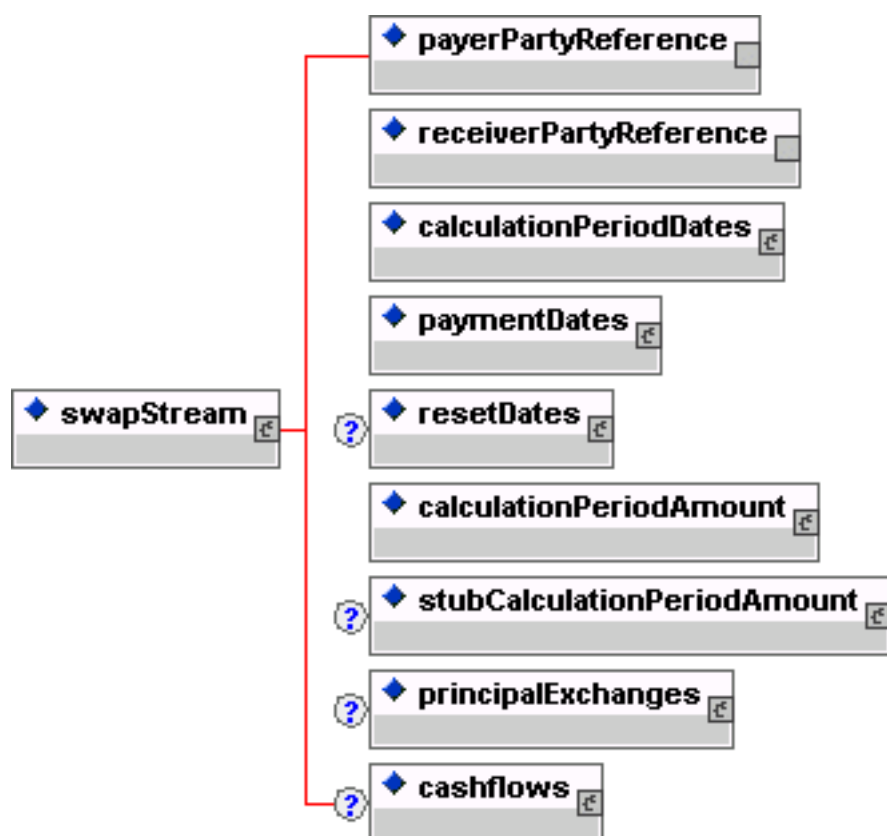
The components within a swapStream cannot be randomly combined and cannot be thought of as existing in their own right; they only make sense in the given context and in relationship to other components within the swapStream container.

In FpML, the schedule of dates within a swapStream is based around the calculationPeriodDates component. The definition of a calculation period in FpML differs in some respects from the International Swaps and Derivatives Association (ISDA) definition of Calculation Period. In the case of a trade involving compounding, ISDA introduces the concept of a Compounding Period, with several Compounding Periods contributing to a single Calculation Period. The FpML calculation period is equivalent to the ISDA definition of Compounding Period when compounding is applicable, i.e. the calculation period frequency will correspond to the compounding frequency. An FpML calculation period is directly comparable to the ISDA defined Calculation Period when only one calculation period contributes to a payment.

The other date components within swapStream are related to the calculationPeriodDates component. The paymentDates and resetDates components contain the information necessary to construct a schedule of payment and reset dates relative to the calculation period dates.

FpML uses the ISDA Floating Rate Option to specify the floating rate being observed. This scheme was used rather than attempting to parameterize into elements because although most floating rate indices are defined fully by a standard set of parameters (namely index, currency and fixing source) there are sometimes other details including fixing offsets and formulas. This approach allows for more flexibility in adding new floating rate indices without having to introduce new elements, although this comes at the expense of a self contained definition within the standard.
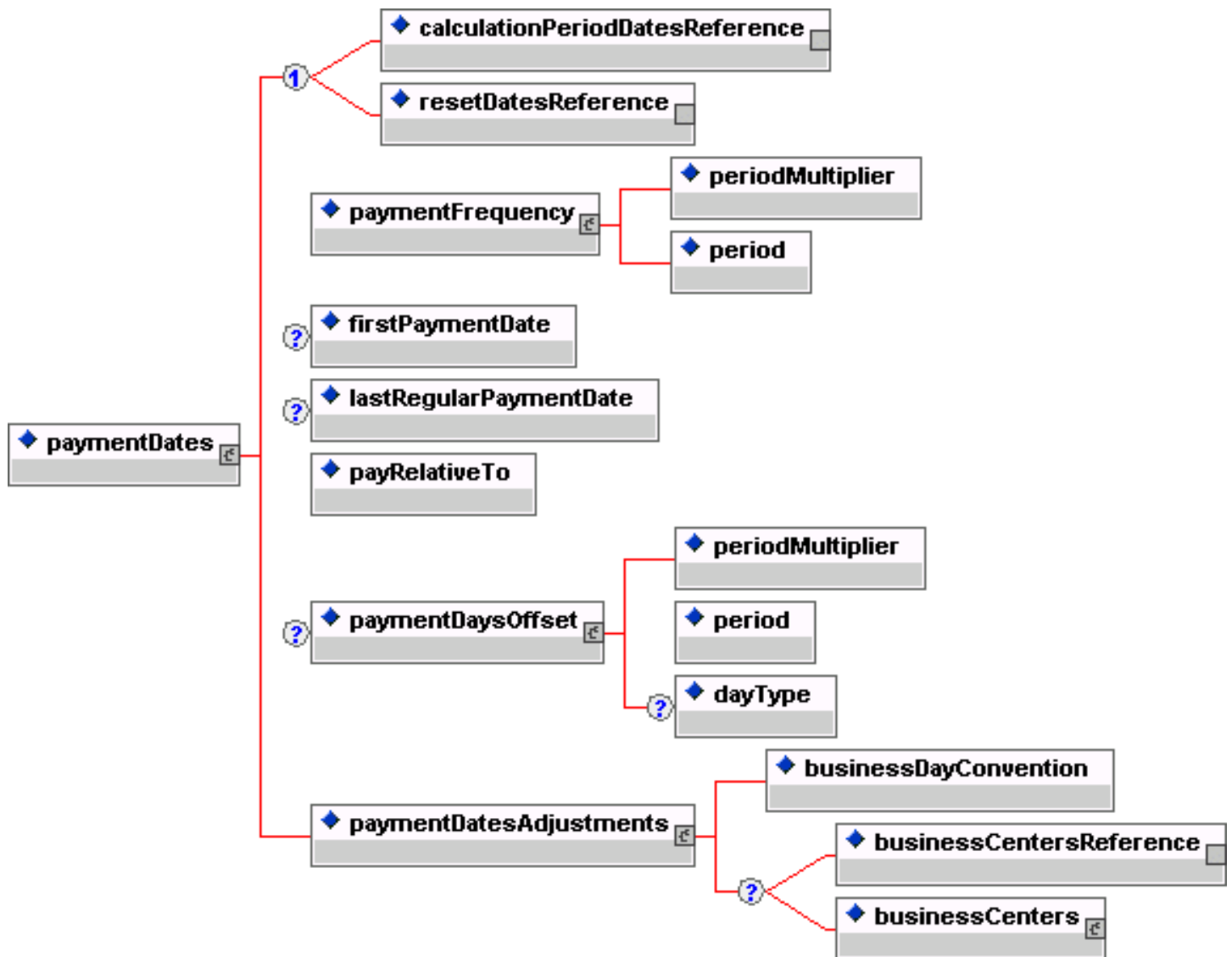
The information relating to amounts and rates is collected in the calculationPeriodAmount and stubCalculationPeriodAmount components. FpML 2.0 has introduced fxLinkedNotionalSchedule as an alternative to notionalSchedule for defining notionals. This allows for the definition of FX Resetable trades by allowing for the notional of a stream to be linked to notionals from another stream by way of the spot fx rate.
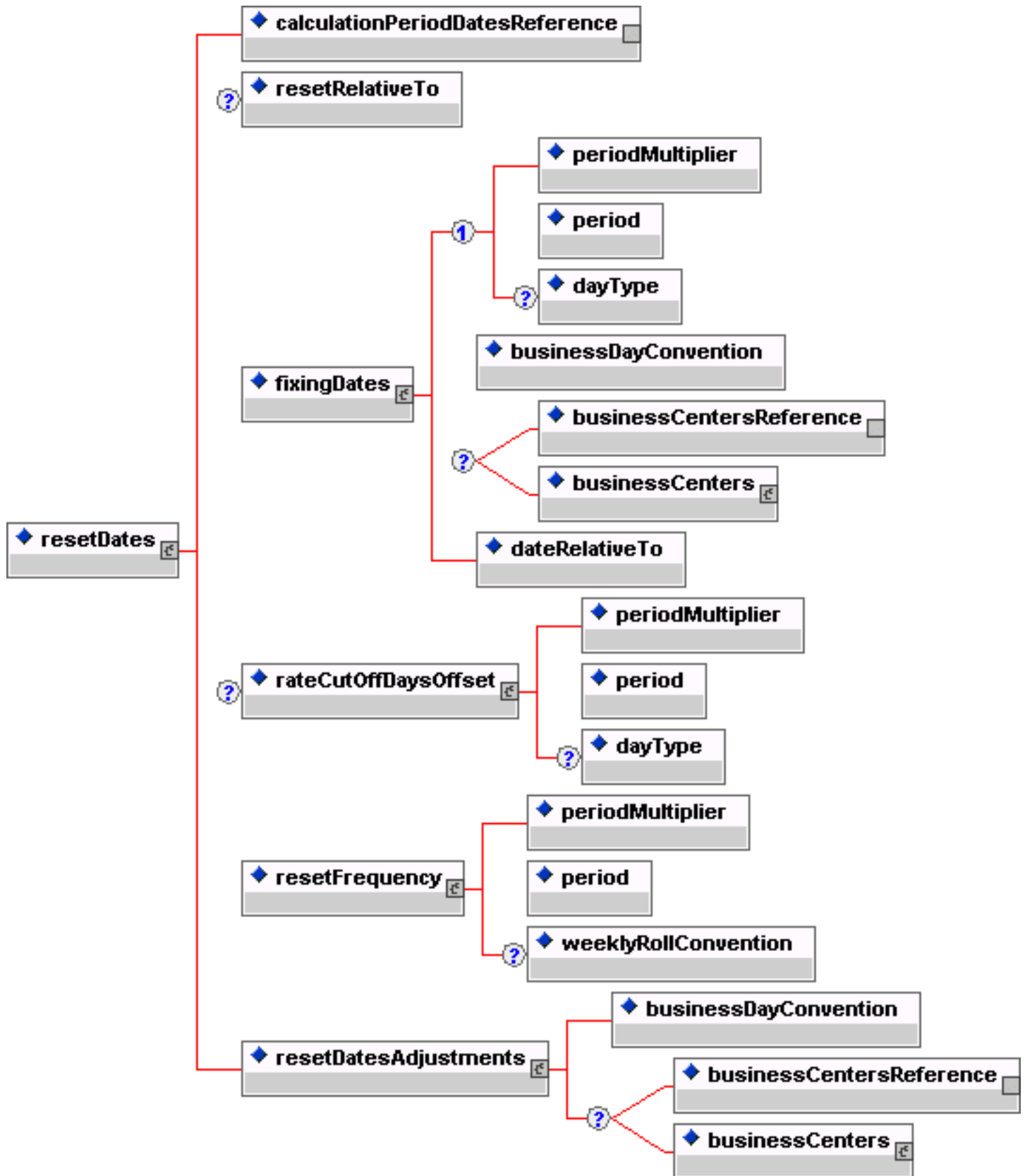
Certain swapStream components are designated as being optional (although it would be more accurate to say that they are conditional). Thus a fixed rate stream never includes a resetDates component, but this is required for a floating rate stream. Similarly, the stubCalculationPeriodAmount component will be required if the swap leg has either an initial or final stub, or indeed both, but should otherwise not be specified. The principalExchanges component is required in the case of cross currency swaps or other types of swap involving exchanges of principal amounts.
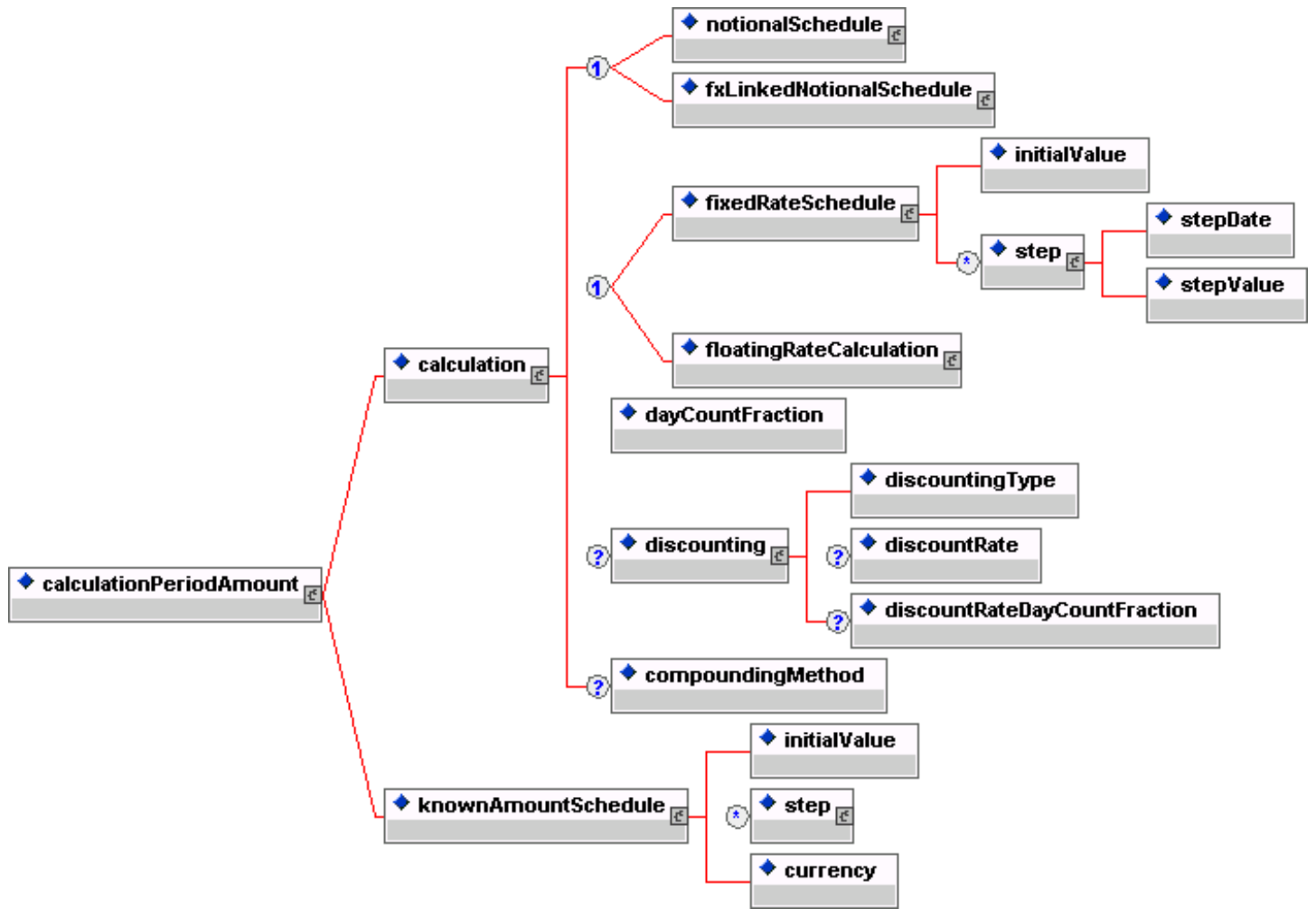
The payerPartyReference and receiverPartyReference elements indicate which party is paying and which receiving the stream payments. This is done by referencing the appropriate party within the party component.
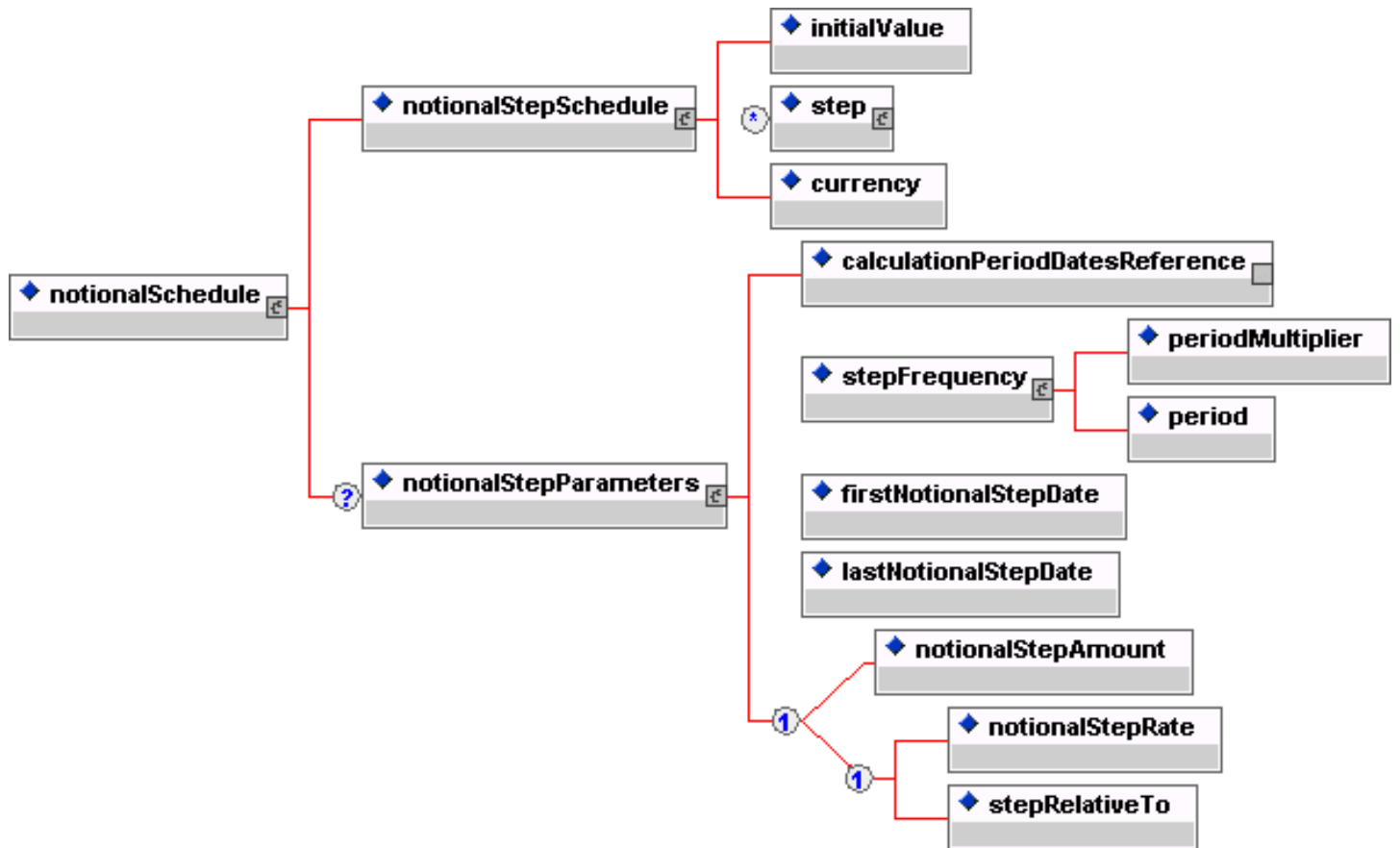
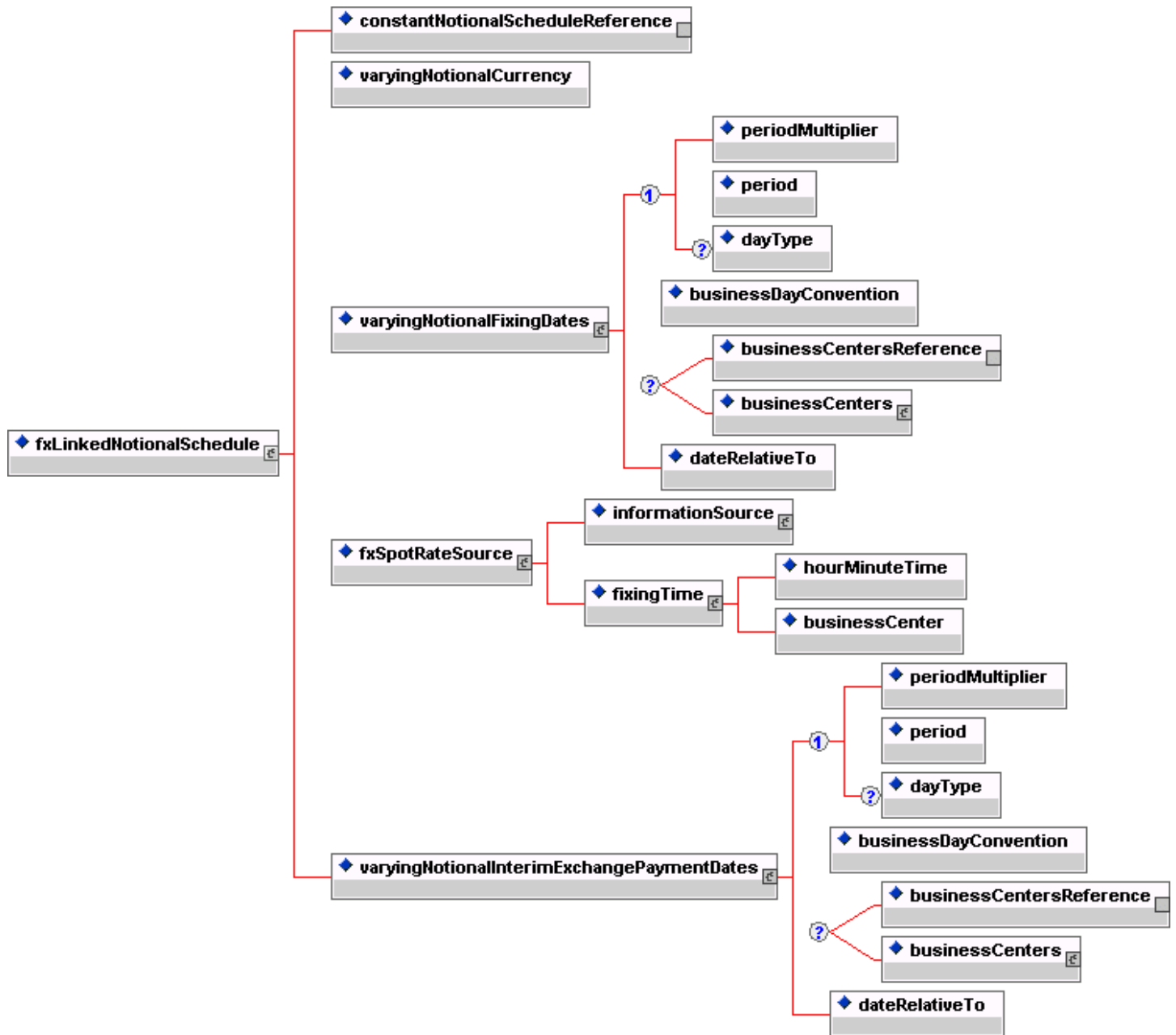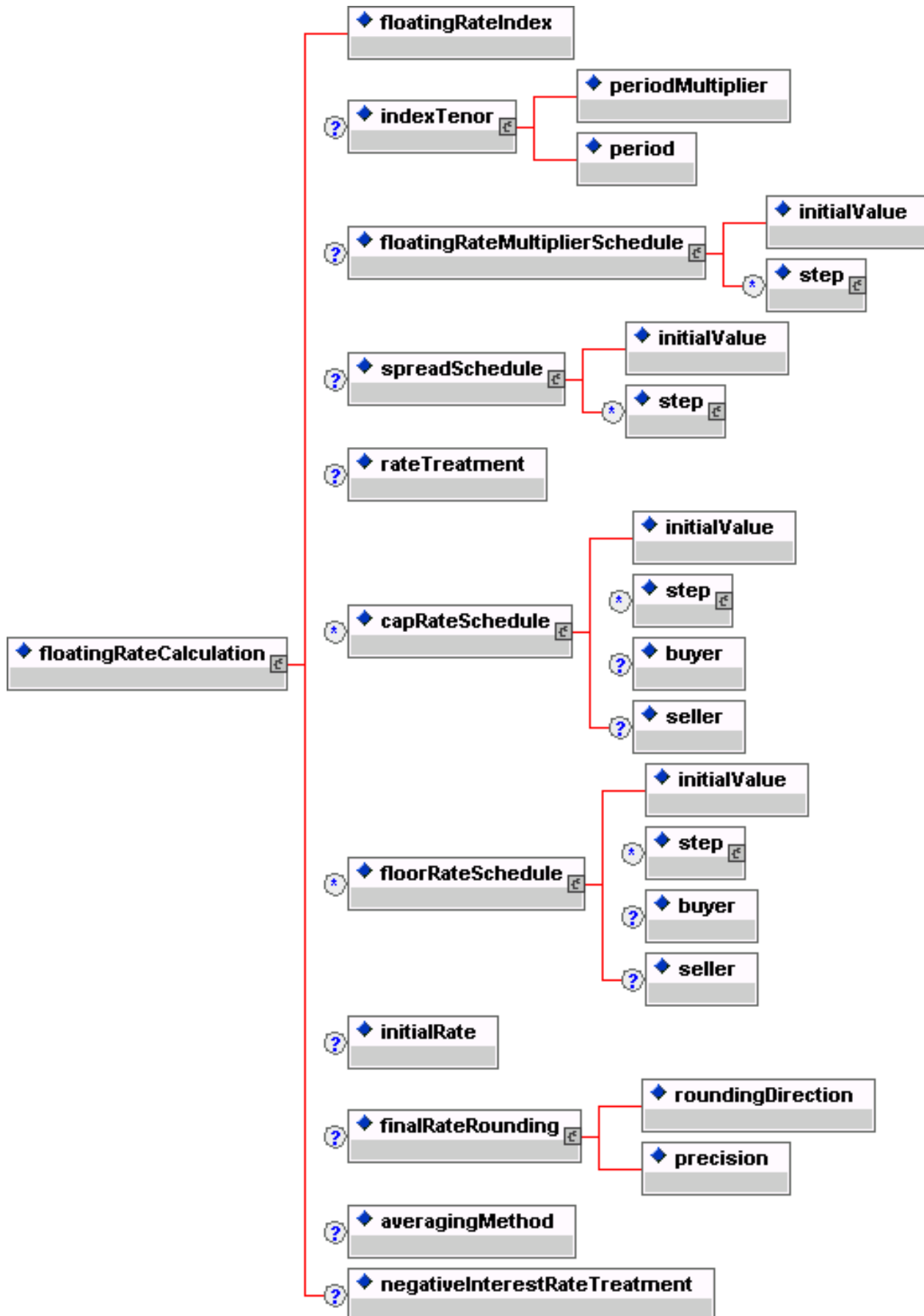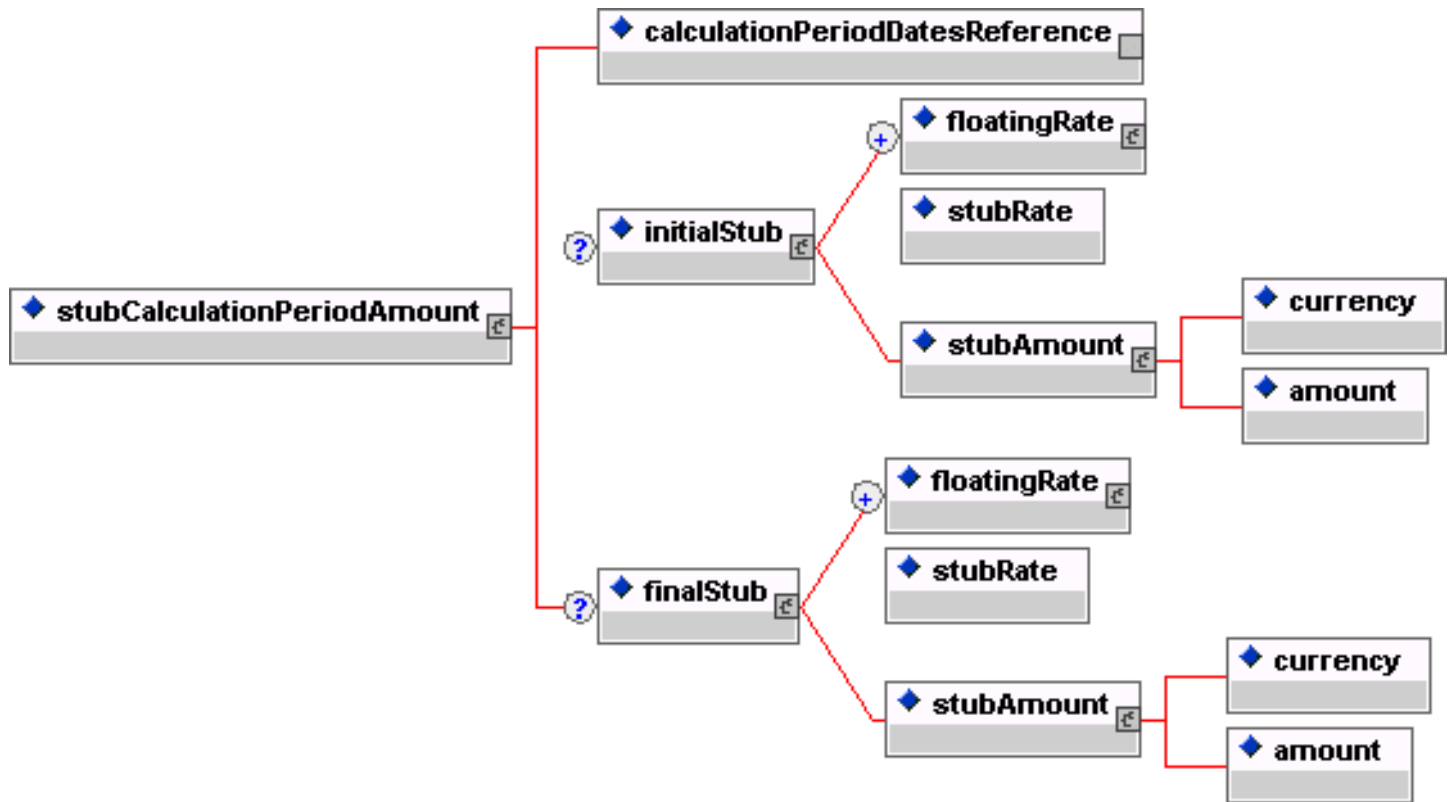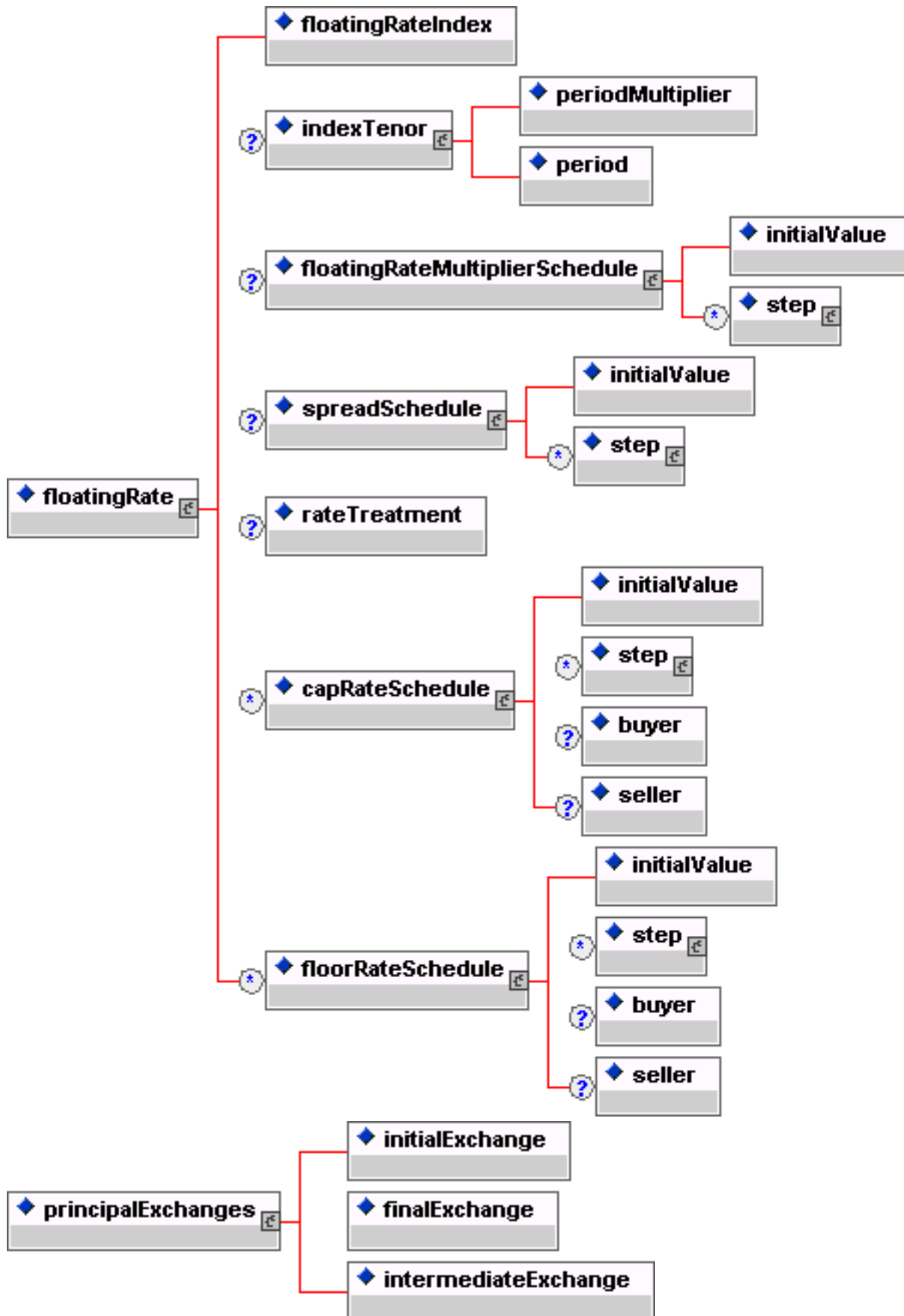The detailed structures within the swapStream are shown diagrammatically below:

floatingRateCalculation

- floatingRateIndex
- indexTenor
  - periodMultiplier
  - period
- floatingRateMultiplierSchedule
  - initialValue
  - step
- spreadSchedule
  - initialValue
  - step
- rateTreatment
- capRateSchedule
  - initialValue
  - step
  - buyer
  - seller
- floorRateSchedule
  - initialValue
  - step
  - buyer
  - seller
- initialRate
- finalRateRounding
  - roundingDirection
  - precision
- averagingMethod
- negativeInterestRateTreatment

10.2 Forward Rate Agreement

As noted above, the definition of a forward rate agreement trade is contained within a single component. A forward rate agreement is a simple and commoditized product. This means there is no variation in the product traded and it is not expected to become more complex in the future.

The structure of the fra component is shown diagrammatically below:
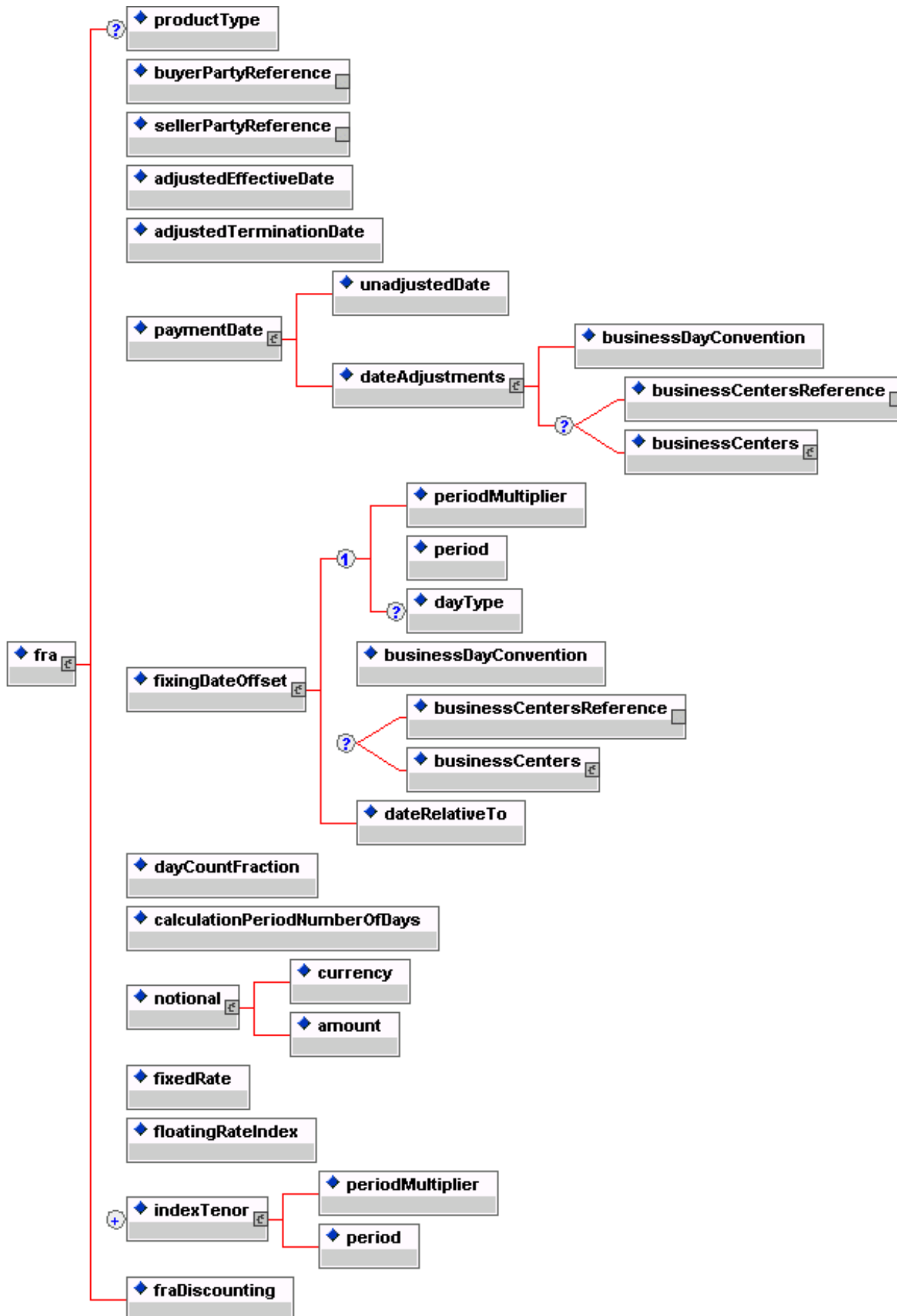
10.3 Option Components

FpML 2.0 has introduced interest rate options. The components introduced are:

- Early Termination Provision (Optional or Mandatory) for a swap
- Cancelable Provision for a swap
- Extendible Provision for a swap
- Swaption
- Cap / Floor

The ISDA 2000 Definitions have been followed closely in defining the various option dates and element names. Thus components for European, Bermudan and American exercise have been defined which are re-used in each of the first four components above. These components share an element called relevantUnderlyingDate whose meaning is dependent on the option component it is contained in:

- OptionalEarlyTermination - It represents the new terminationDate of the underlying swapStreams if the trade is terminated early.
- CancelableProvision - It represents the new terminationDate of the underlying swapStreams if the trade is cancelled.
- ExtendibleProvision - It represents the new terminationDate of the underlying swapStreams if the trade is extended.
- Swaption - It represents the effectiveDate of the underlying swapStreams if the swaption is exercised.

10.3.1 European Exercise

This is a style of option to which the right or rights granted are exercisable on a single date referred to as the expiration date. This date can be specified either as an adjustableDate or as a relativeDate though the latter is only expected to be used in the case of cash settled cancellations where the expiration date may be defined as an offset to the cash settlement payment date.

The relevantUnderlyingDate is optional, in its absence the effectiveDate of the underlying is the effectiveDate defined in the swapStreams. This can only be excluded for european swaptions.

europeanExercise

- expirationDate
 - adjustableDate
    - unadjustedDate
    - dateAdjustments
  - relativeDate
    - periodMultiplier
    - period
    - dayType
    - businessDayConvention
    - businessCentersReference
    - businessCenters
    - dateRelativeTo
- relevantUnderlyingDate
  - adjustableDates
  - relativeDates
- earliestExerciseTime
  - hourMinuteTime
  - businessCenter
- expirationTime
  - hourMinuteTime
  - businessCenter
- partialExercise
  - notionalReference
  - integralMultipleAmount
  - minimumNotionalAmount
- exerciseFee
  - notionalReference
  - feeAmount
  - feeRate
  - feePaymentDate
    - periodMultiplier
    - period
    - dayType
    - businessDayConvention
    - businessCentersReference
    - businessCenters
    - dateRelativeTo

10.3.2 American Exercise

This is a style of option to which the right or rights granted are exercisable during the exercise period which consists of a period of days. The underlying should specify its effective date based on the earliest possible exercise. When exercise implies a stub period this will be taken to be a short stub at the start, i.e. the underlying swap defines a series of flows, exercise merely brings the flows into existence from the relevantUnderlyingDate.
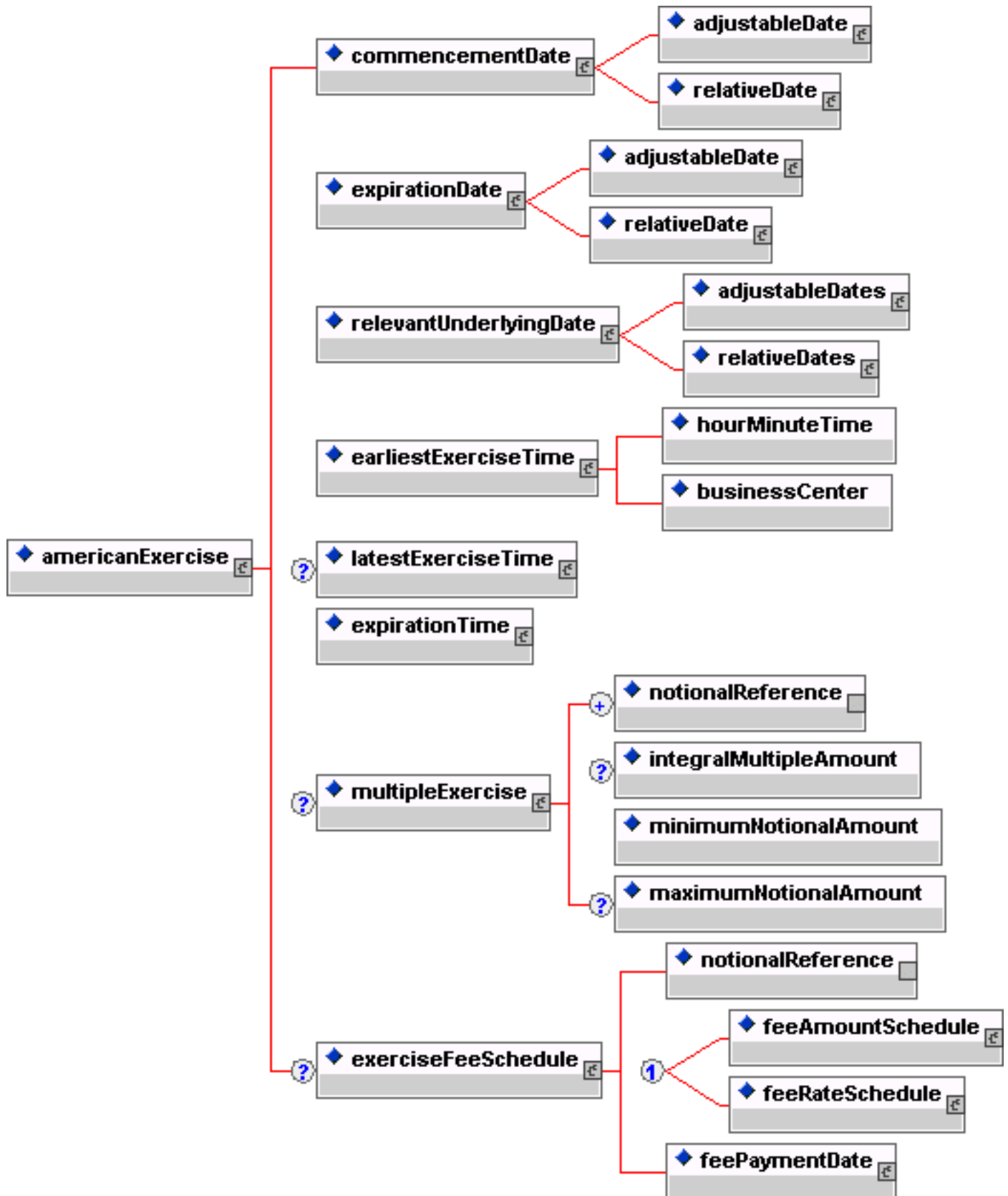
10.3.3 Bermudan Exercise

This is a style of option to which the right or rights granted are exercisable during an exercise period which

consists of a number of specified dates. These dates can be defined as a list together with adjustments or by reference to an existing schedule elsewhere in the trade (e.g. resetDates). In the latter case bounds can be placed on the referenced schedule to define a subset of the whole schedule.
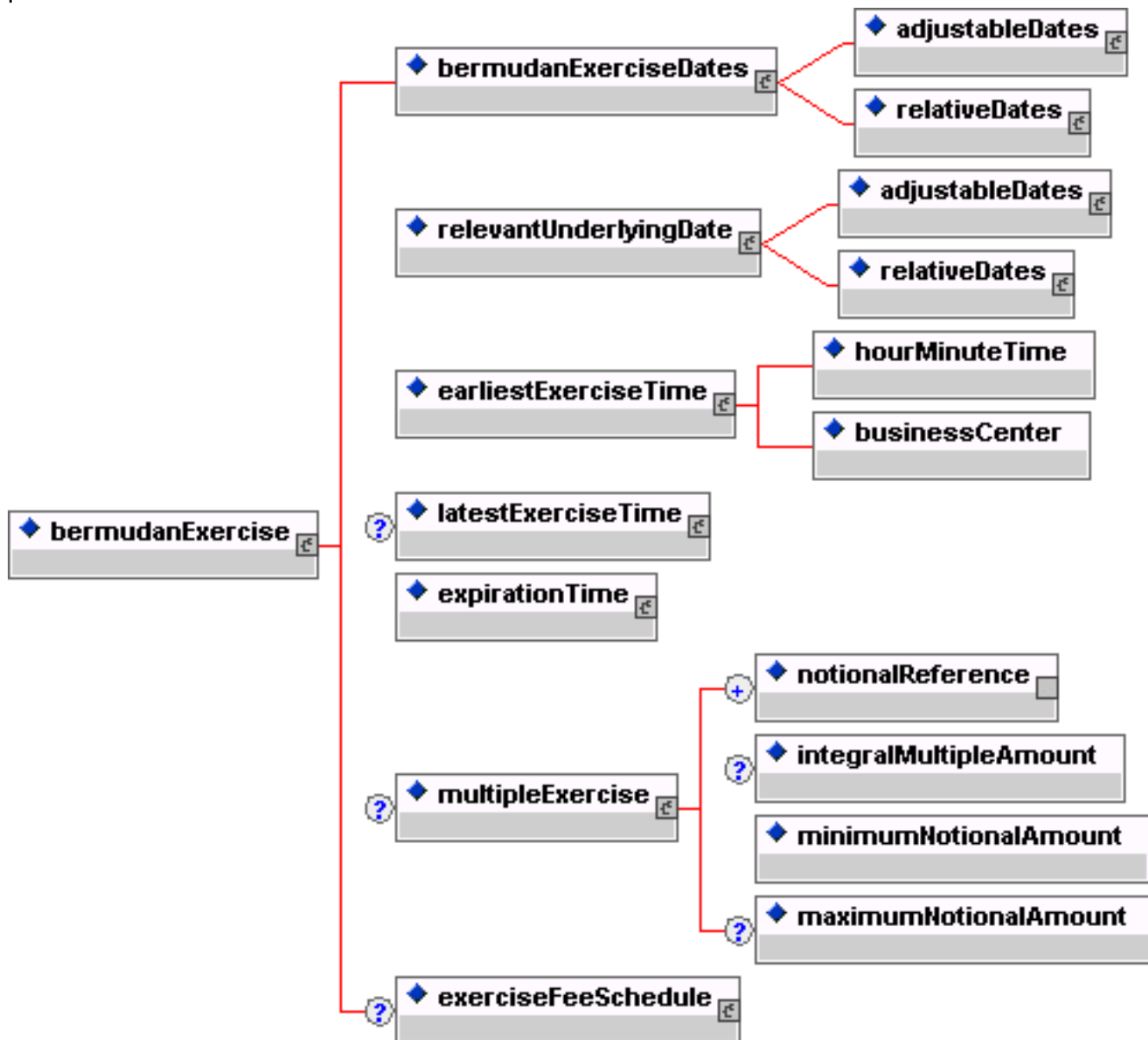
10.3.4 Early Termination Provision

The right for one or both parties to terminate the trade and settle the remaining term of the swap for fair value. In the case of a mandatory early termination the termination is mandatory.
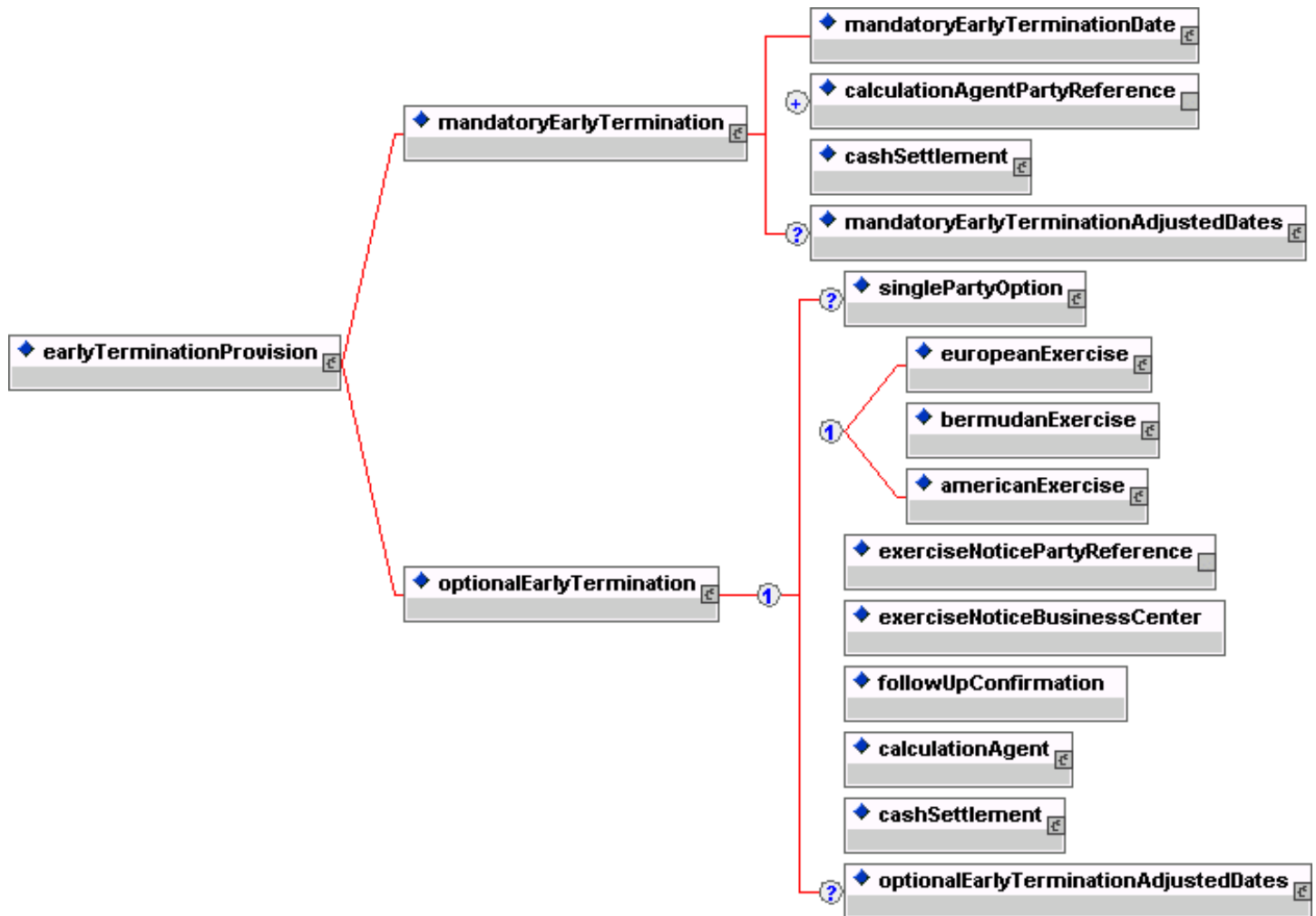
10.3.5 Cancelable Provision

With a cancelableProvision the seller grants the buyer the right to terminate all swapStreams, typically in exchange for an upfront premium. Unlike optionalEarlyTermination, the cancellation of the swap does not require the parties to exchange a cash settlement amount on exercise representing the fair value of the remaining life of the swap although an exercise fee can be specified in the exerciseFeeSchedule.
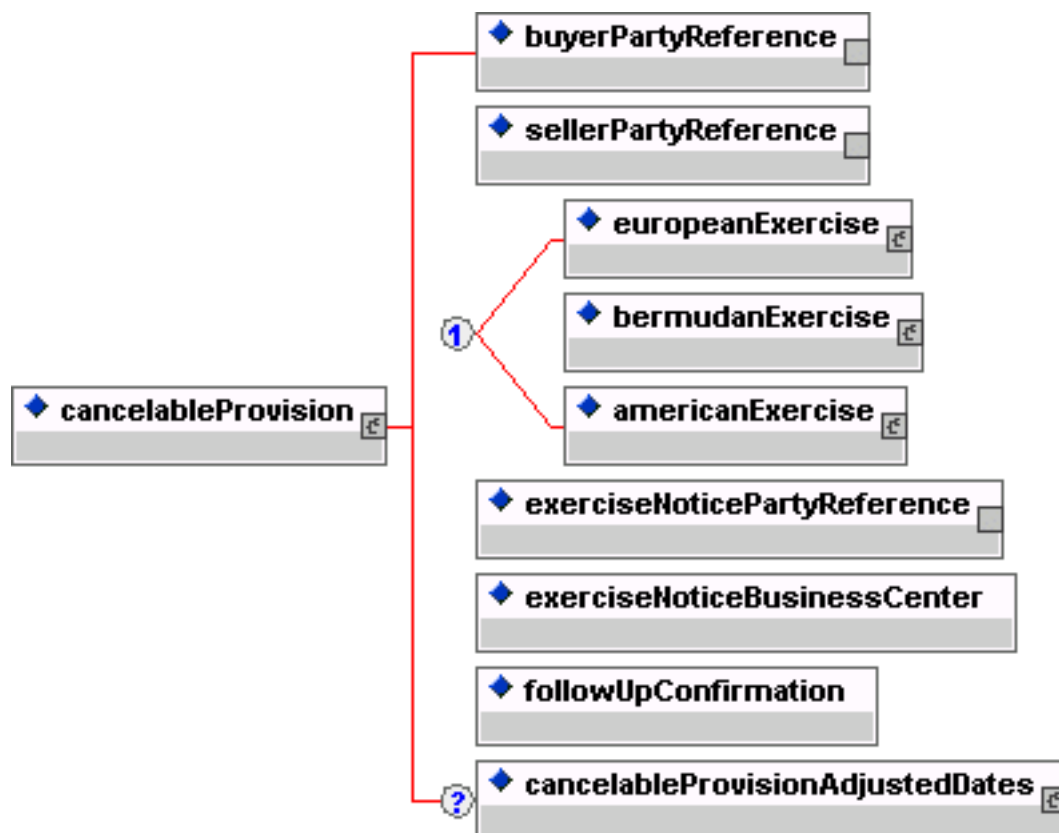
10.3.6 Extendible Provision

With an extendibleProvision the seller grants the buyer the right to extend all swapStreams, typically in exchange for an upfront premium. This provision is very similar to a cancelableProvision and in fact the two share the same market risk profile. FpML makes a clear distinction between the two since the operational risk associated with mis-recording the type of applicable provision can be high. For example, a 10 year swap with the right to cancel after 5 years has exactly the same risk profile as a 5 year swap with the right to extend for 5 years after 5 years. However, failing to give notice of exercise after 5 years will in one case (extendibleProvision) result in the swap terminating after 5 years and in the other case (cancelableProvision) result in the swap terminating after 10 years, i.e. action after 5 years is required in one case to lengthen the term of the swap in the other to shorten it.
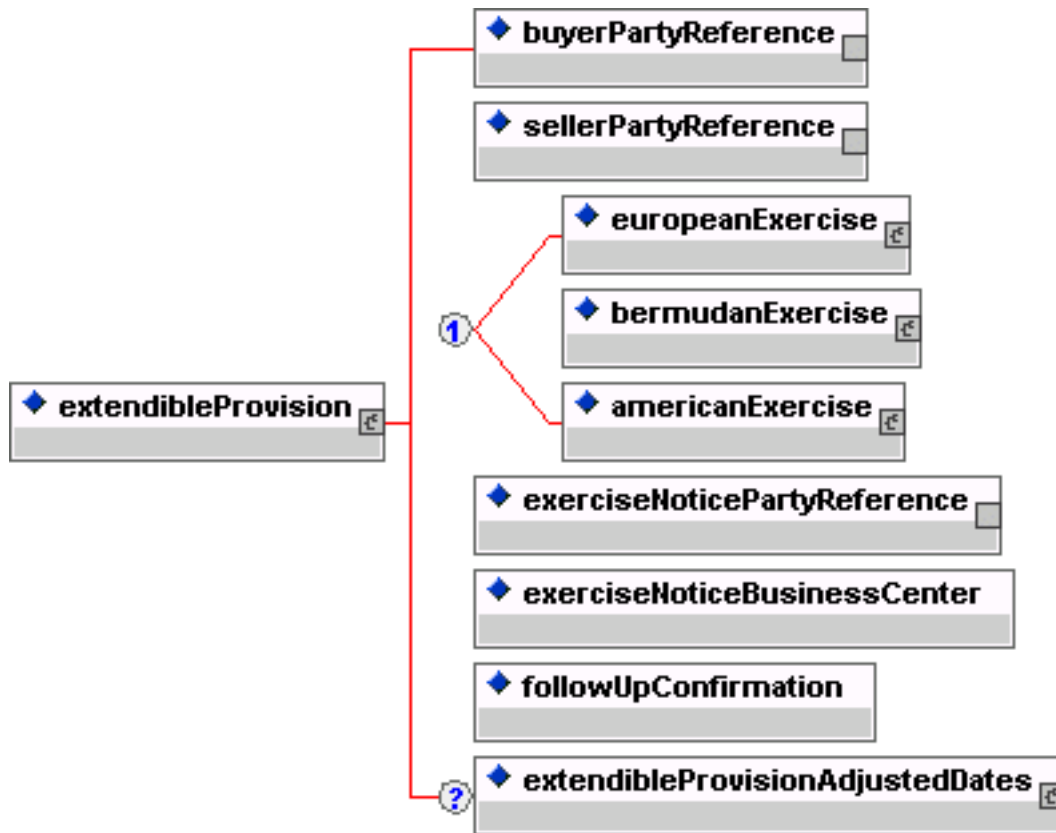
10.3.7 Swaption

The option to enter into a swap is defined as it's own product and contains the underlying swap as a swap element. A swaption straddle is defined by setting the swaptionStraddle element to true: this implies that the swaption buyer has the right, on exercise, to decide whether to pay or receive fixed on the underlying swap. If the underlying does not contain a single fixed stream and a single floating stream then the straddle is invalid and thus this flag should be set to false..
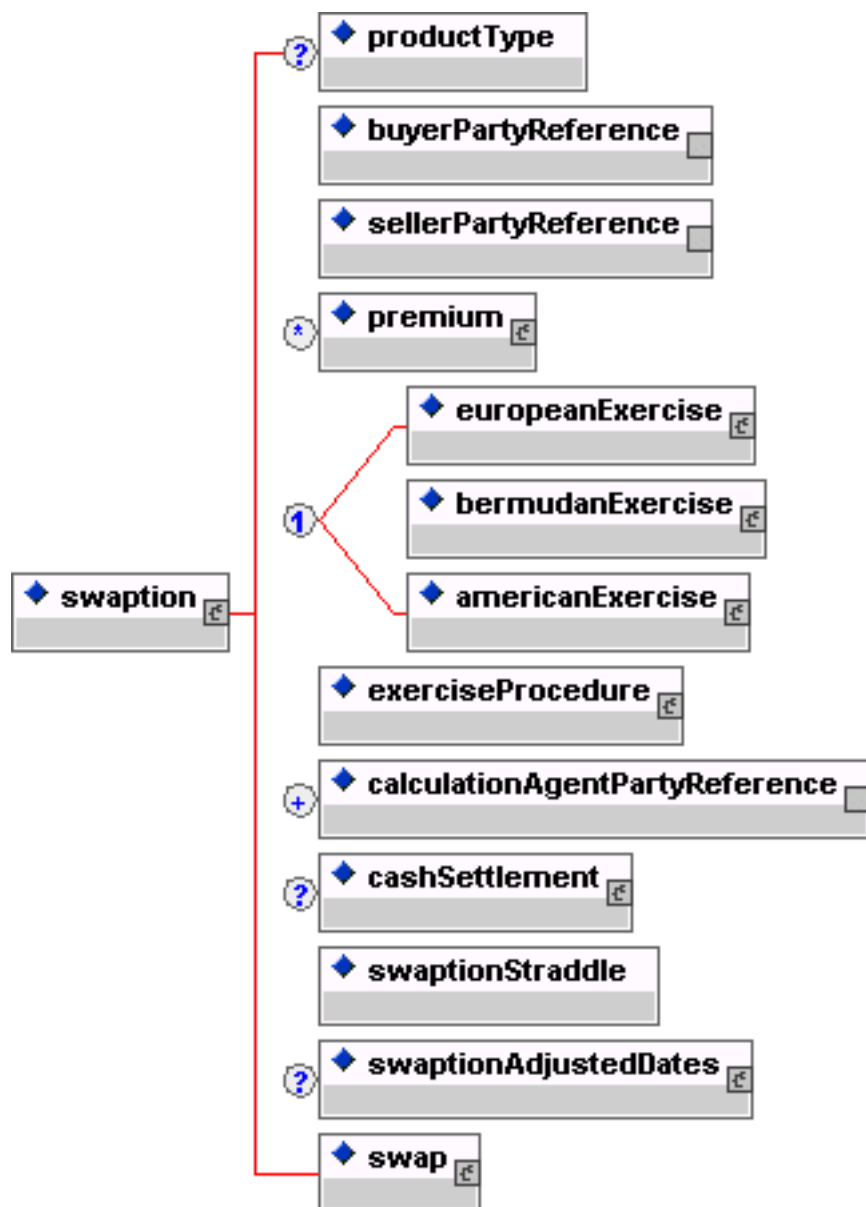
10.3.8 Cap / Floor

Caps and Floors are defined as one or more capFloorStreams and zero or more additionalPayments. The capFloorStream re-uses the FpML_InterestRateStream entity and thus its content is identical to a swapStream.
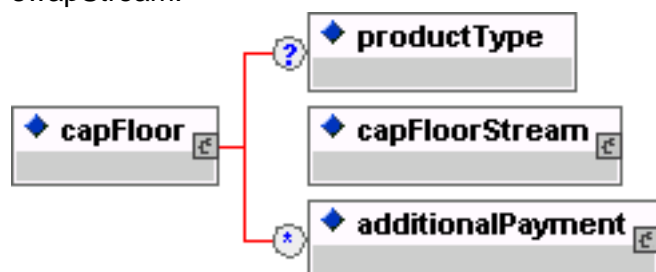
Though a capFloorStream allows the definition of fixed streams or known amount streams these are not the intended use of this component and there use would be considered an invalid FpML trade.

The floatingRateCalculation component has been amended to allow the specification of cap/floor structures within a single stream (e.g. straddles, corridors). The changes are:

- The occurrence rules for the components capRateSchedule and floorRateSchedule have been changed from 'zero or one' to 'zero or more'.
- An optional buyer and seller reference have been added to these schedules

These additions allow for multiple cap and floor rates to be added to the stream and to define precisely which party bought and sold them. To maintain backward compatibility with FpML1.0 the buyer and seller are optional. When absent the following rules apply:

- CapRateSchedule: BUYER = Stream payer SELLER = Stream receiver
- floorRateSchedule: BUYER = Stream receiver SELLER = Stream payer

10.4 Cash Settlement

The cash settlement component is used by mandatoryEarlyTermination, optionEarlyTermination and swaption. The language used within the component corresponds to the ISDA language for the various cash settlement methods. Of the five methods included, three share one underlying component and the other two share another component. Thus there is re-use whilst maintaining ease of identification of the type. Also, this approach allows for easy integration of other methods should they arise.
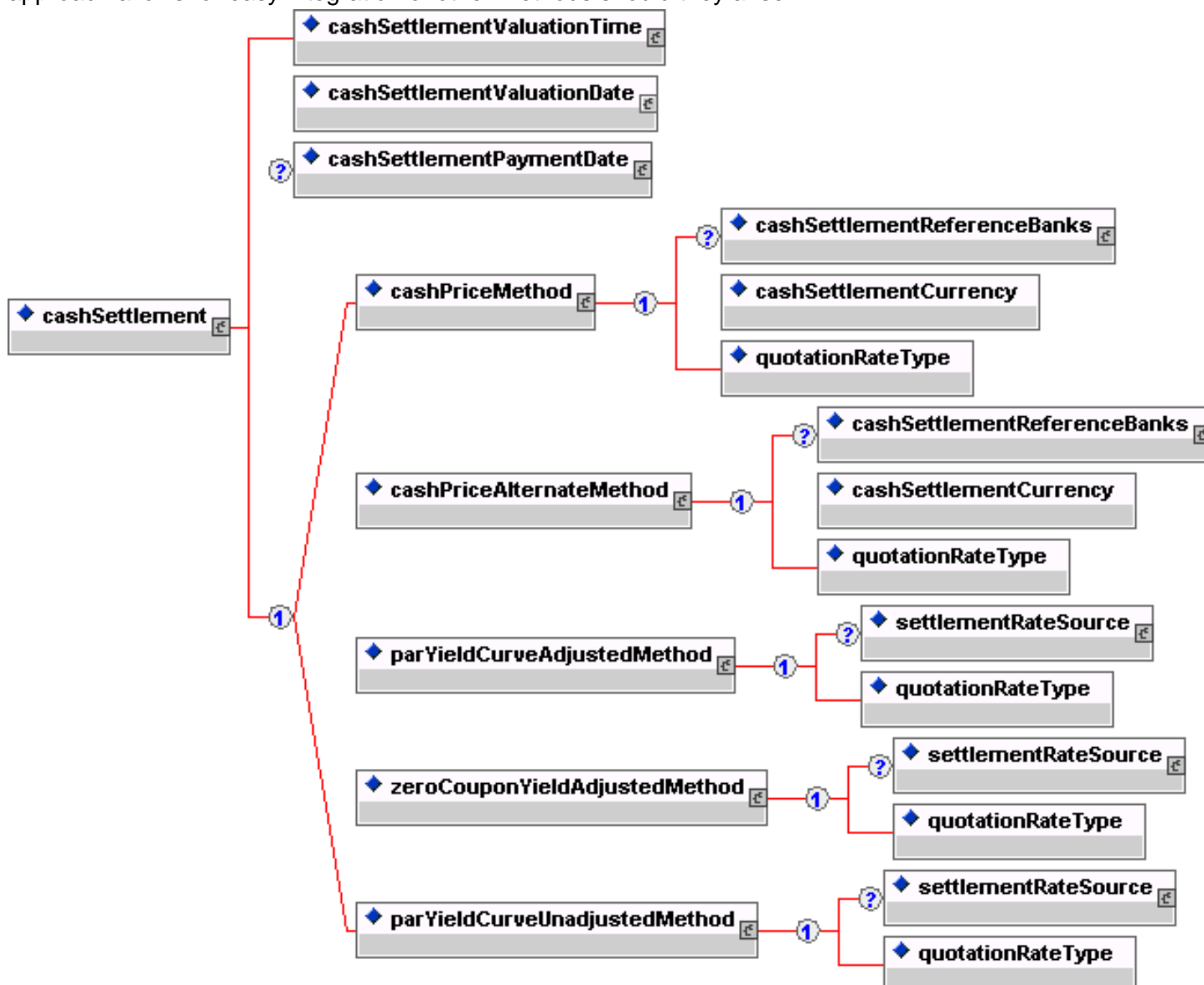
# 11 FX PRODUCT ARCHITECTURE

11.1 Foreign Exchange Spot and Forward

Foreign exchange single-legged instruments include spot and forwards. fxSingleLeg contains two instances of the exchangedCurrency component (the first currency and the second currency), either a single value date component for the trade or an optional value date per exchanged currency, a single instance of the exchangedRate component, and an optional nonDeliverableForward component.
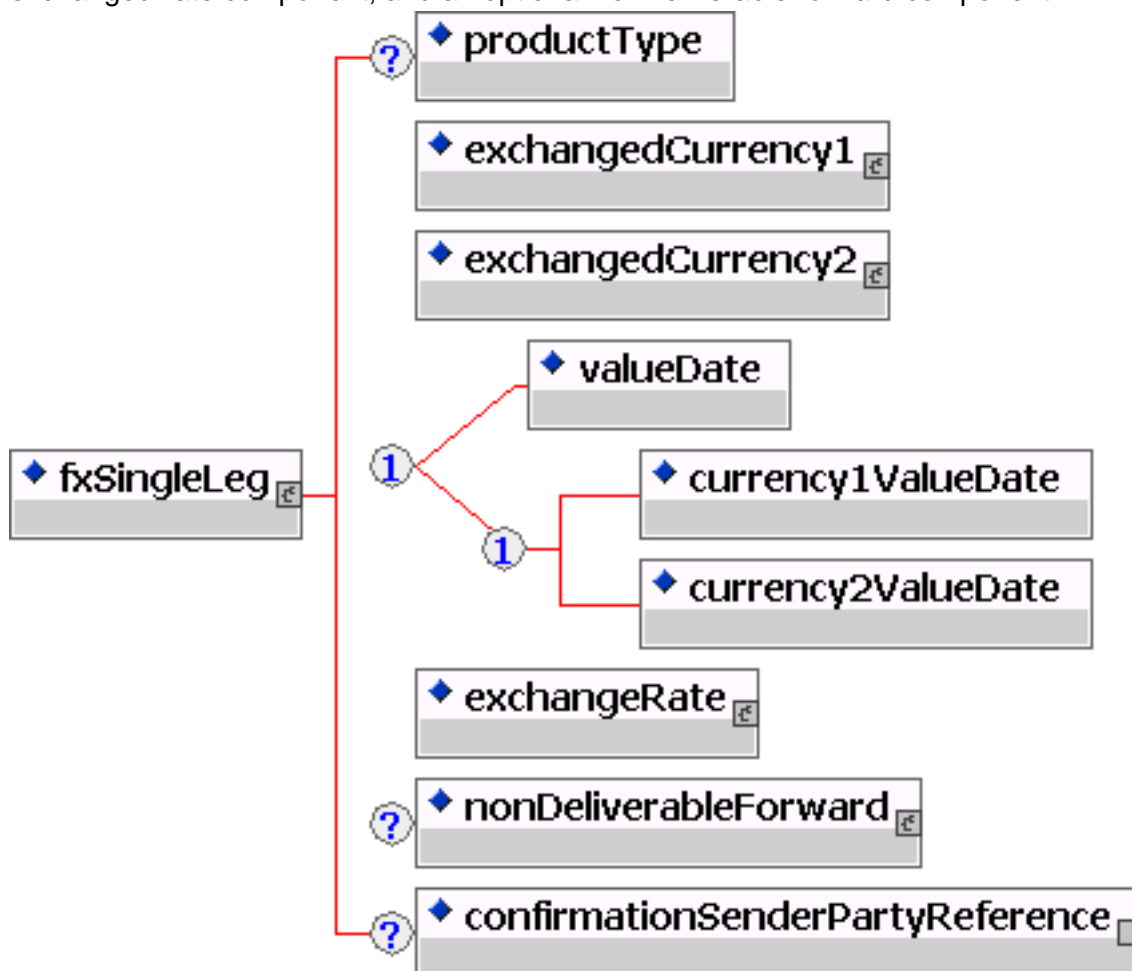

The simple FX transaction contains two currencies which are exchanged between parties. The characteristics of each currency exchange: the currency, the amount, and optionally settlement instructions, are described in the exchangedCurrency structure. An optional payment date is allowed per currency, if there is a requirement to provide for date adjustments for each currency based upon business day conventions to accommodate unscheduled holidays.
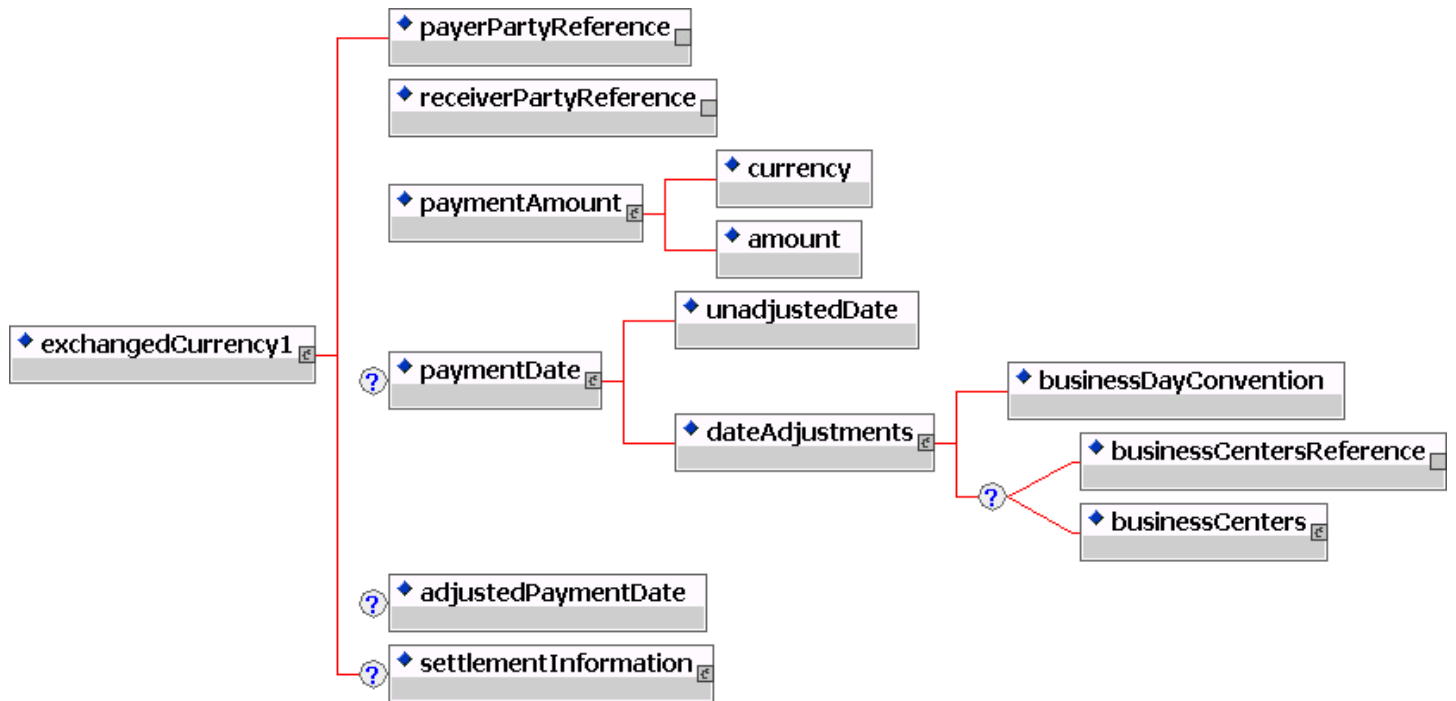
The rate of exchange is required for a foreign exchange trade. The rate of exchange includes a reusable entity (FpML_QuotedCurrencyPair) that describes the underlying composition of the rate: the currencies and the method in which the rate is quoted. The actual trade rate is required, but other rate information such as spot rate and forward points are also accommodated. For non-base currency trades, side rates (or rates to base) are provided for.
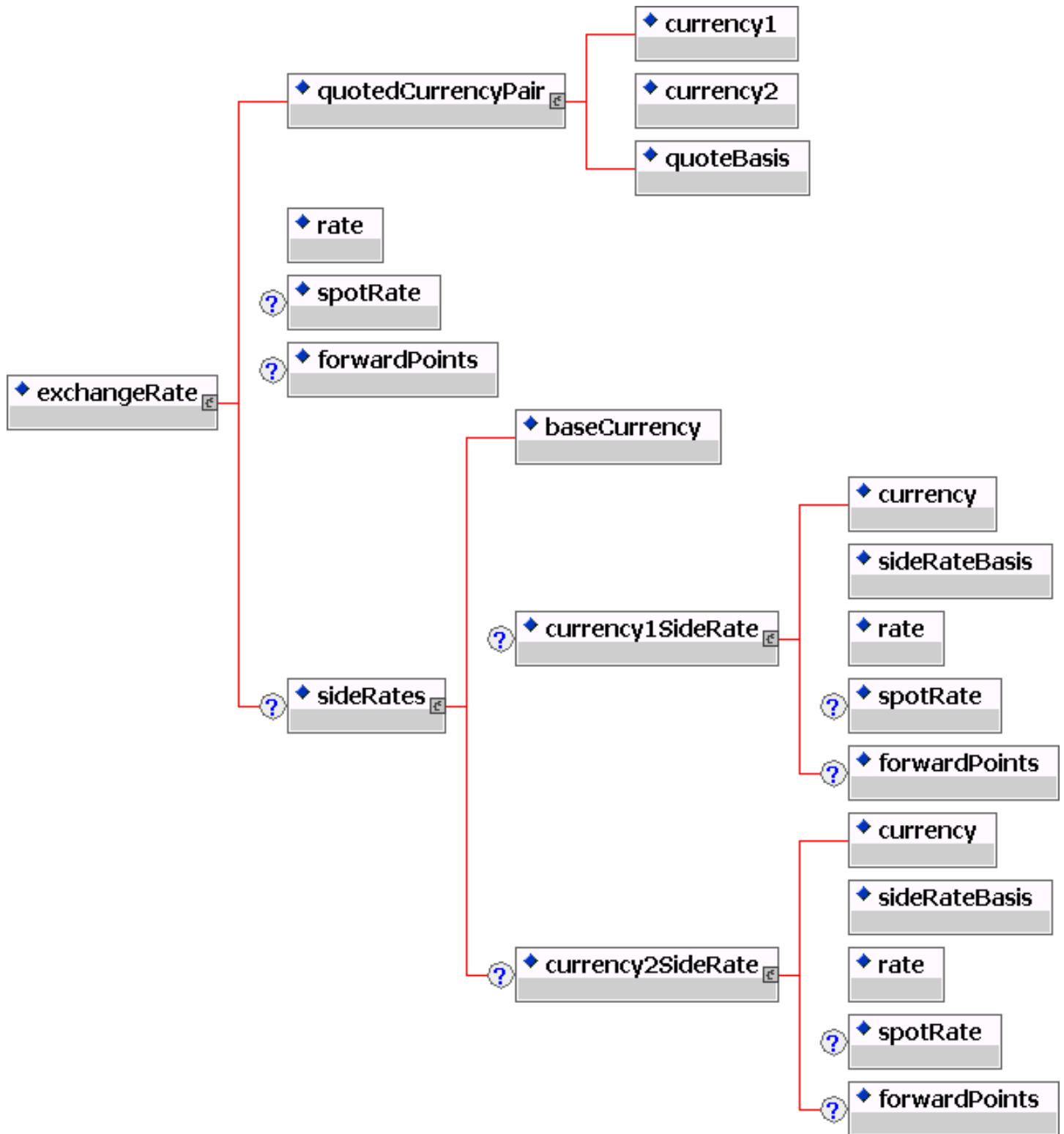
Non-deliverable forwards are catered for within the conventional FX single leg structure by including an optional non-deliverable information structure. This contains identifies the agreed-upon settlement currency and describes the fixing date and time, as well as the settlement rate source that the fixing will be based upon. The non-deliverable structure is shown below.
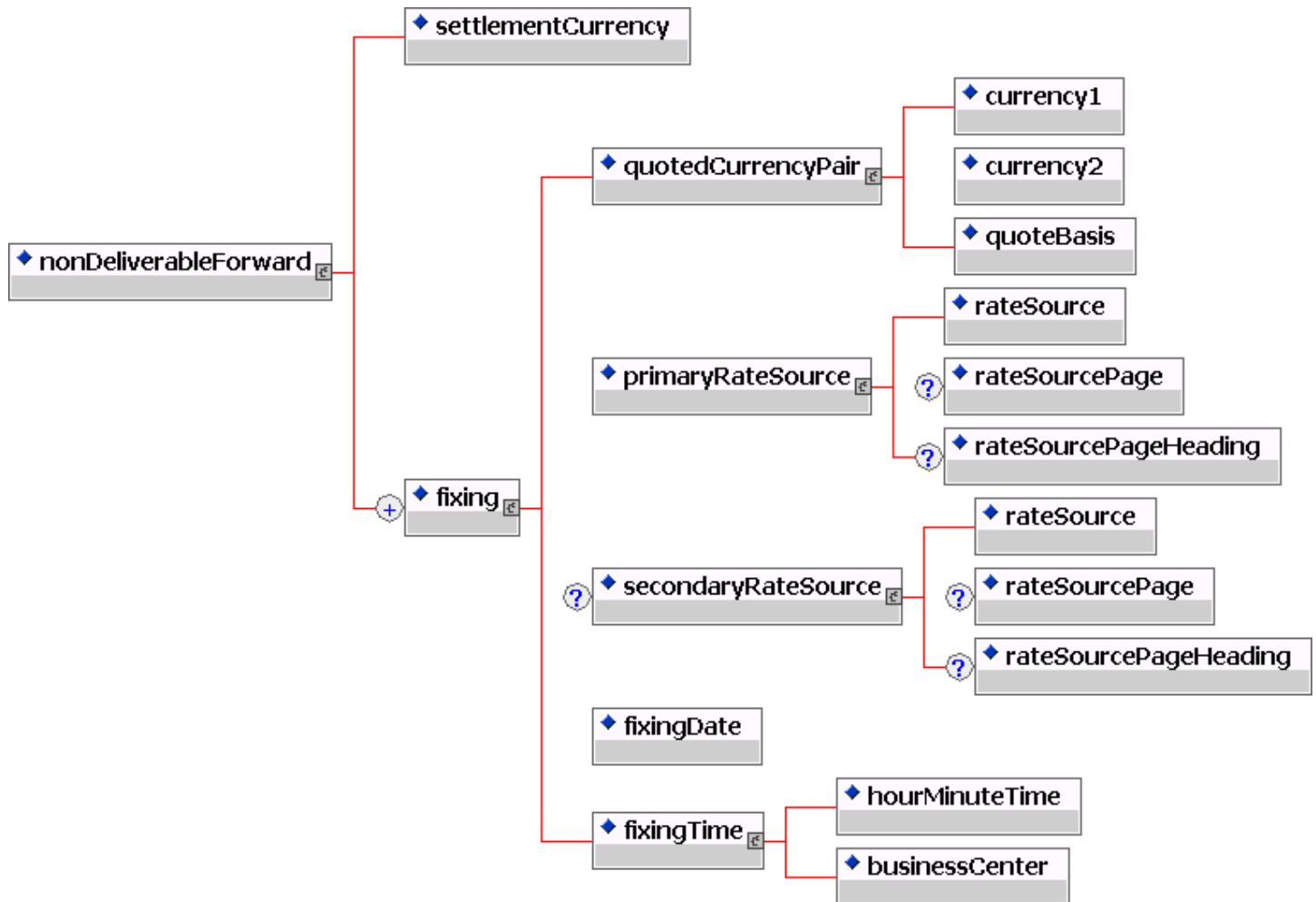
Significant effort has been spent in the development of FX to incorporate the appropriate information required for trade confirmation and settlement. An optional settlementInformation structure has been included for each exchanged currency. This can be used in a variety of ways: not at all, flagging a trade for standard settlement, flagging a trade for settlement netting, or specifying the detailed settlement instructions for that particular currency flow.
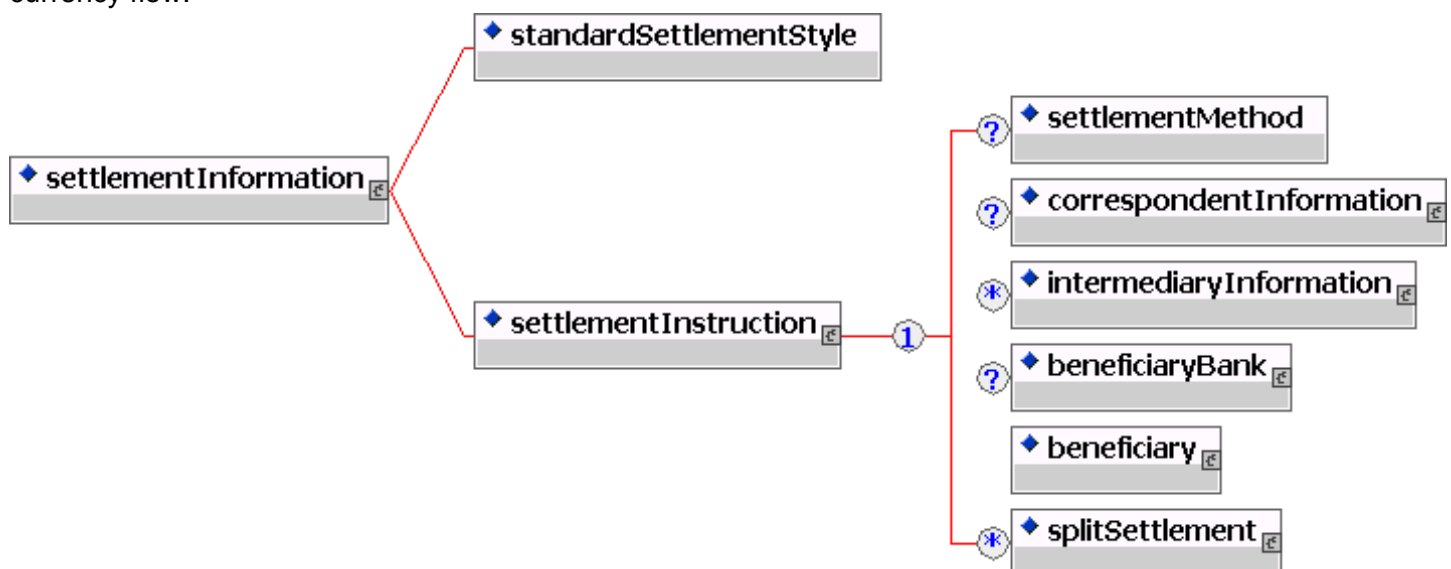

If the specific settlement instruction is included, then this is broken out into correspondent, intermediary, and beneficary information. This includes the identificaiton of the routing mechaism (e.g., SWIFT, Fedwire, etc.)

that the trade will settle via and the id and account that the trade will settle via. Routing can be handled either via purely a routing id (e.g., SWIFT code), routing details (a customer name, address, and account number), or a combination of routing id and details. The following diagrams show the correspondent, intermediary, and beneficiary structures.
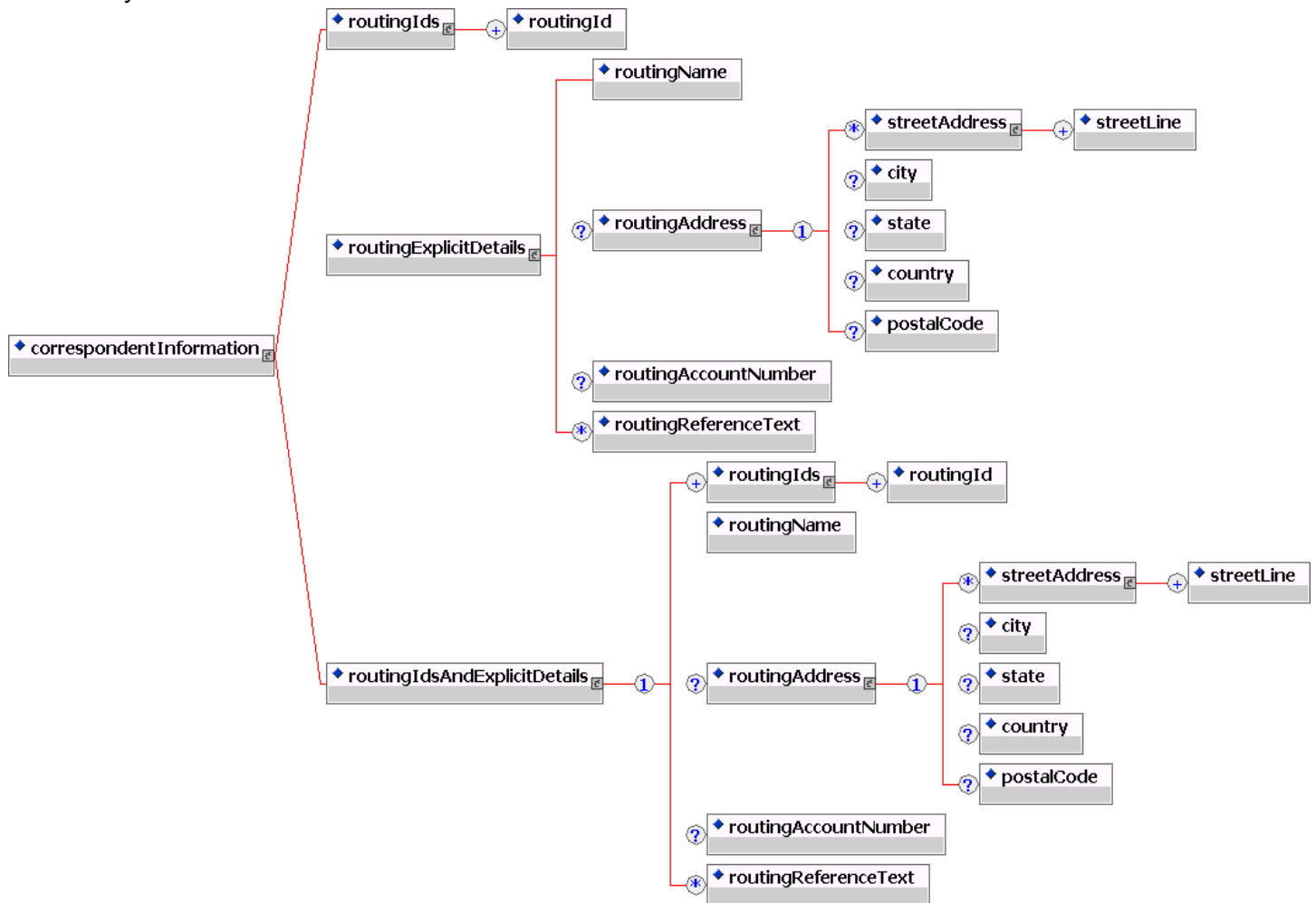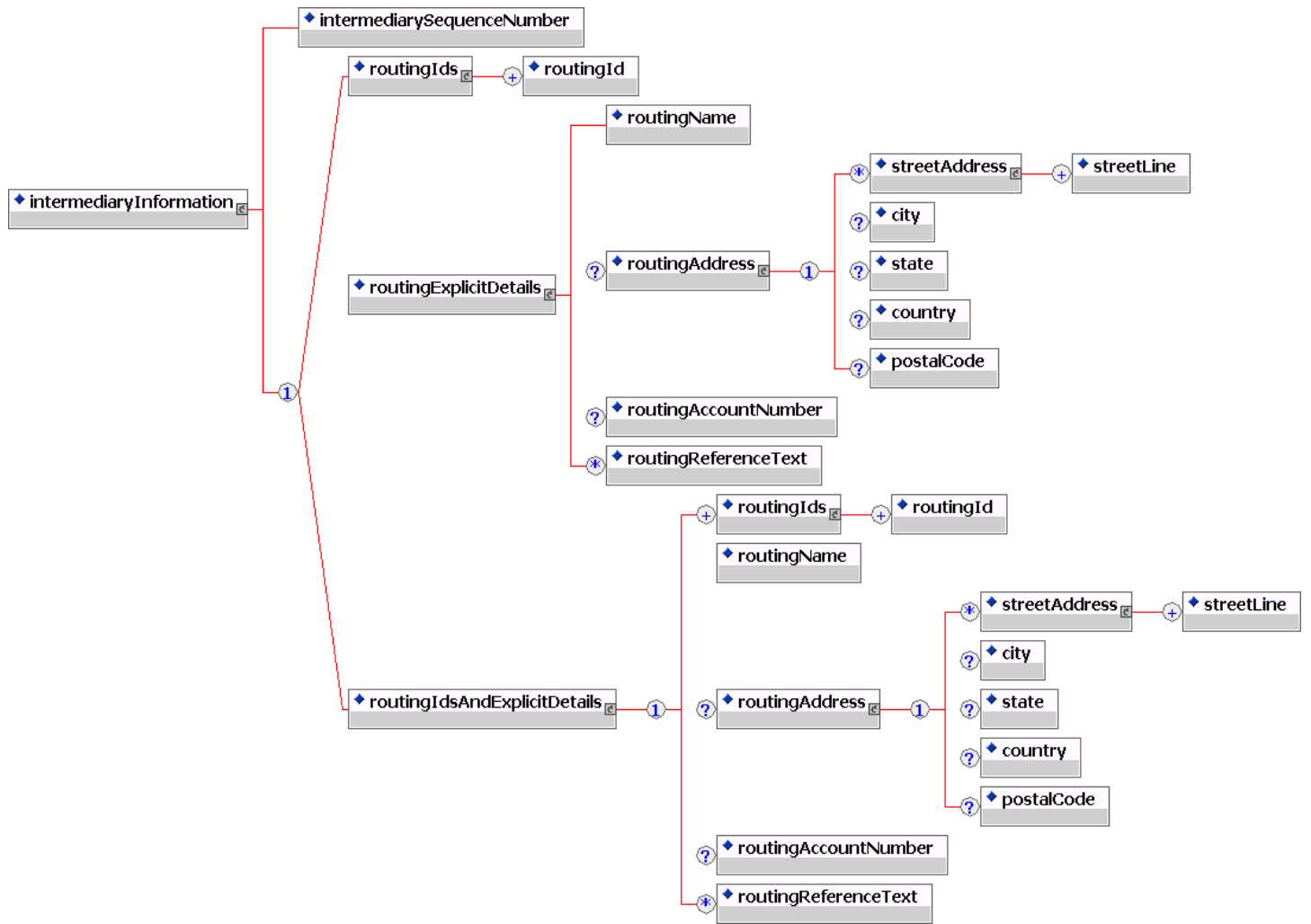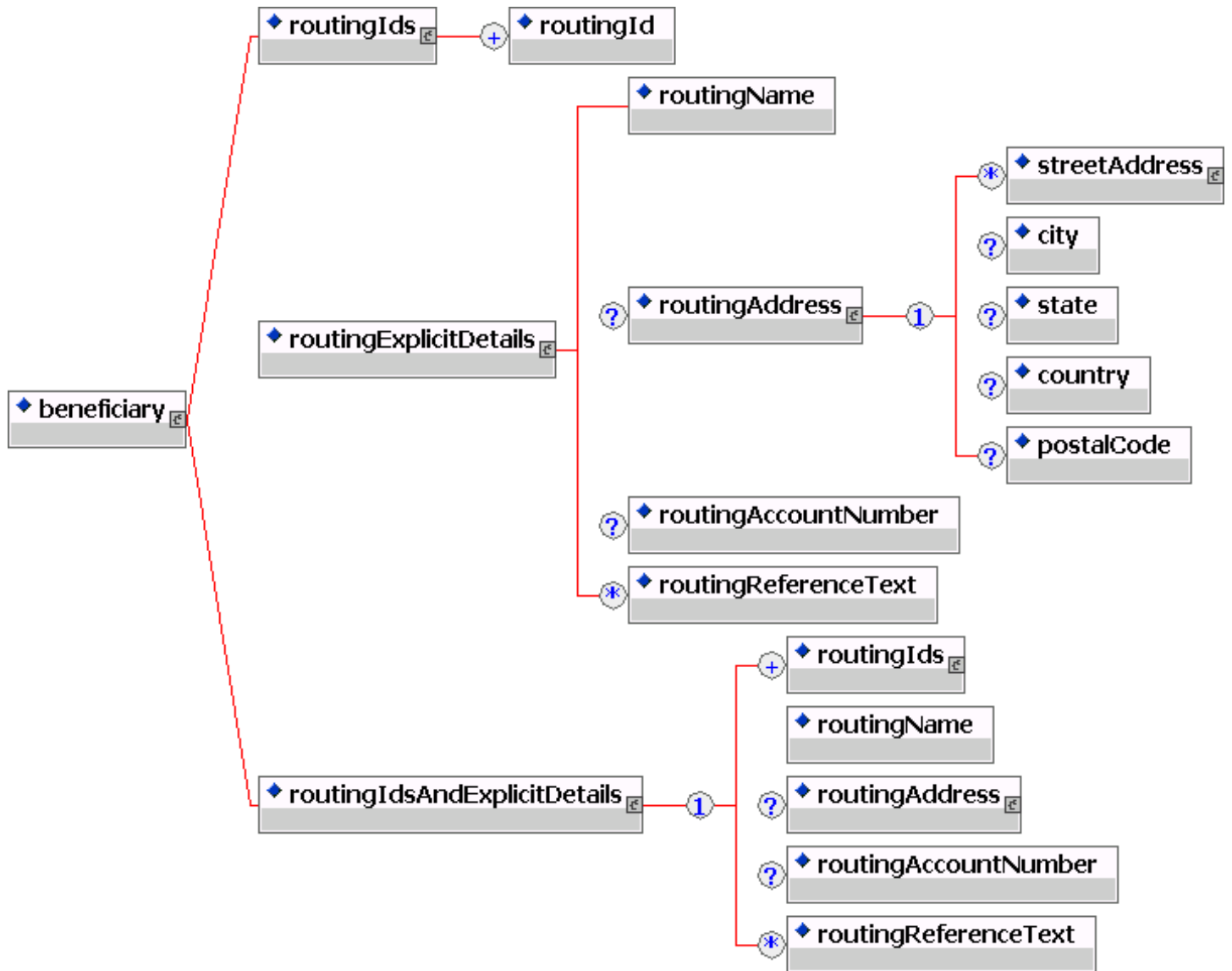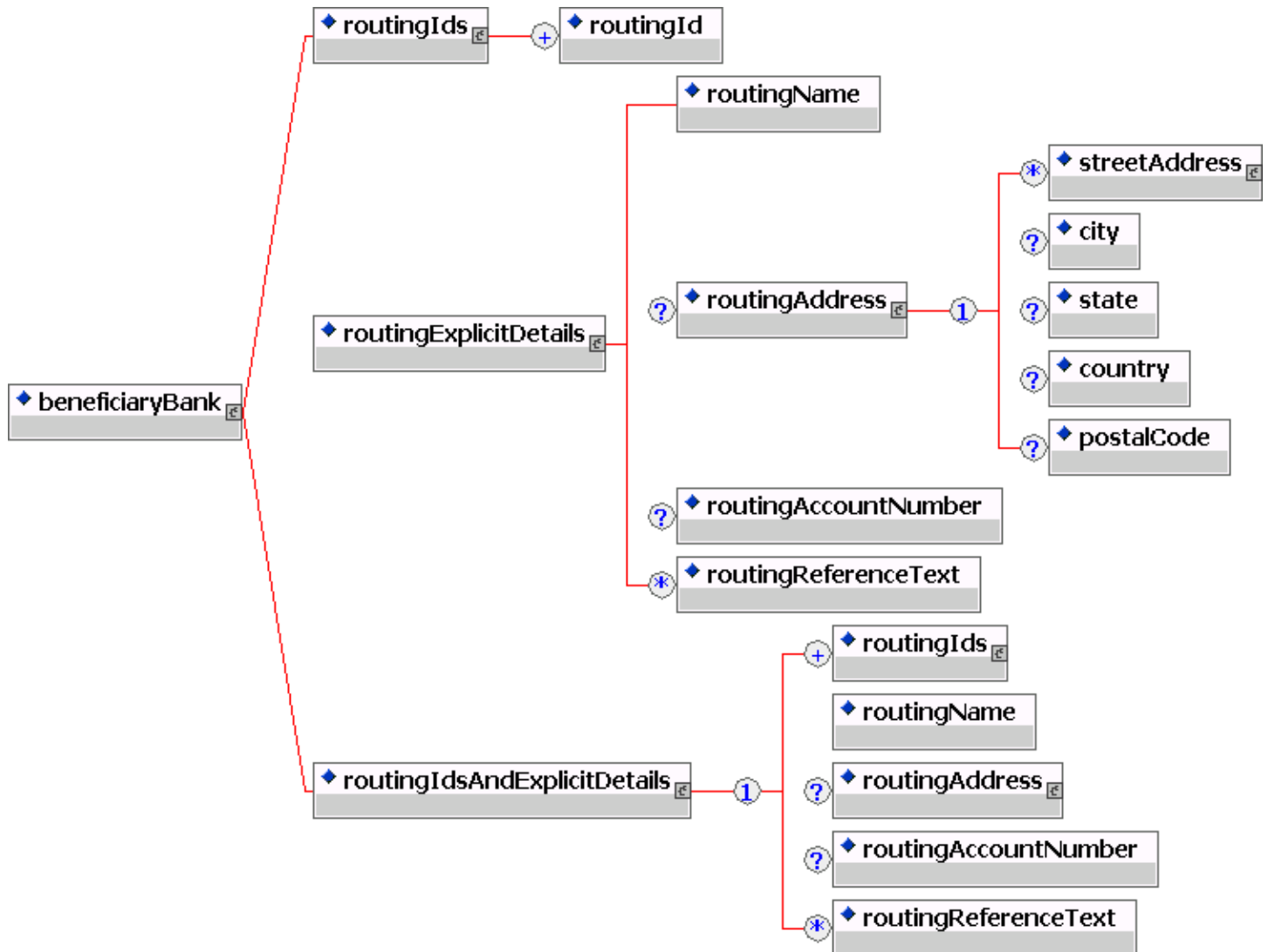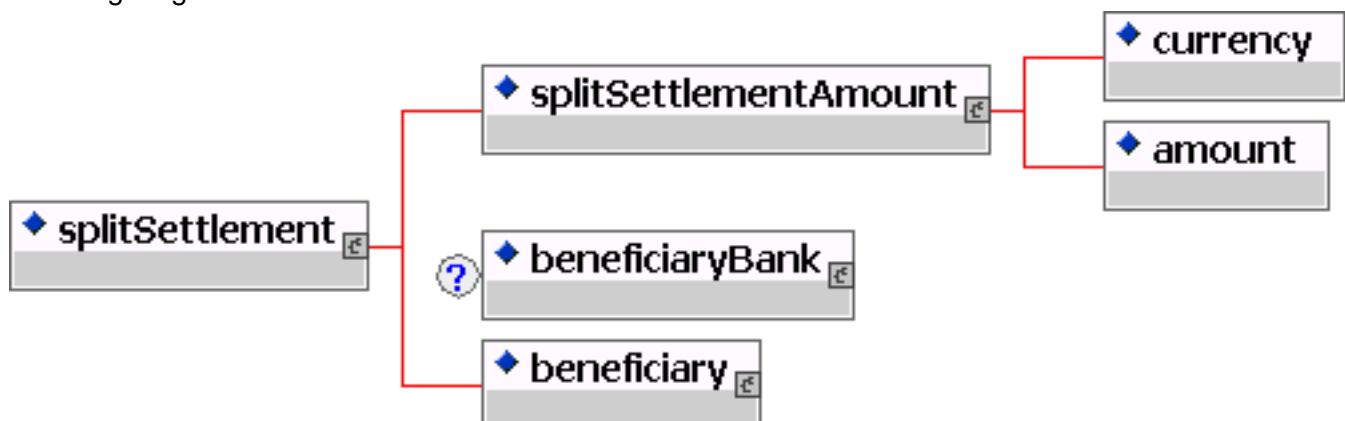
Split settlement is also accommodated. Split settlement will mean that there will be multiple beneficiaries associated with a single trade, where the payment amounts are broken down between beneficiaries. The following diagram shows how this has been modeled:

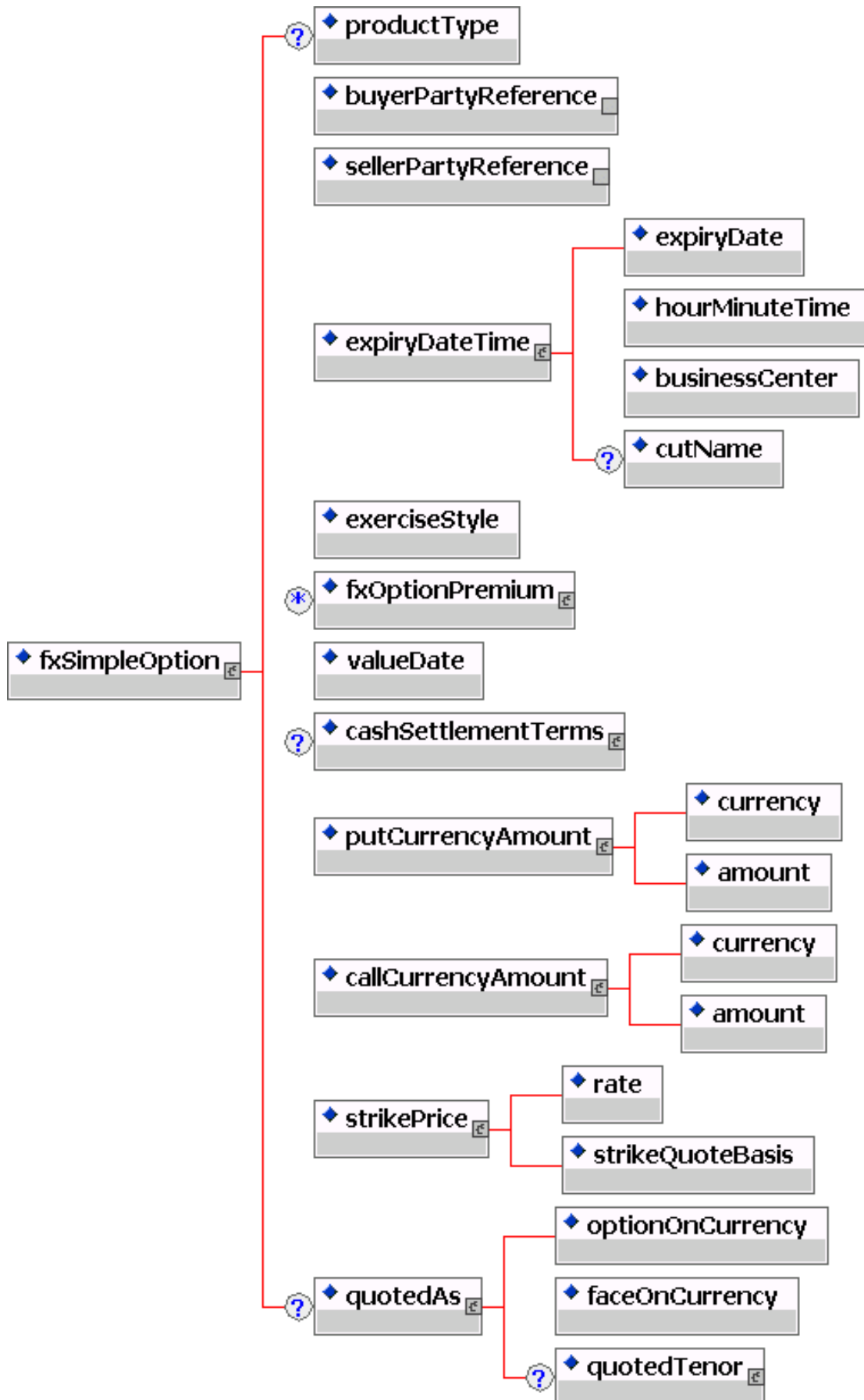

11.2 Foreign Exchange Swap

Foreign exchange swaps are a very simple extension of FpML_FXLeg, whereby the FX swap is simply multiple legs. A standard FX swap contains two legs, whereby the second leg has a value date that is greater than the value date on the first leg. There are a variety of different types of FX swaps in the marketplace:

standard (round-amount) swaps, overnight swaps, unequal-sided swaps, forward-forward swaps. Therefore, all of the features that are available within a standard spot or forward trade (described previously) can be utilized in describing an FX swap as well.
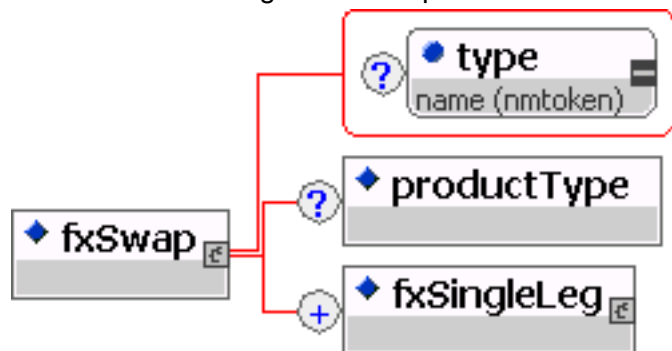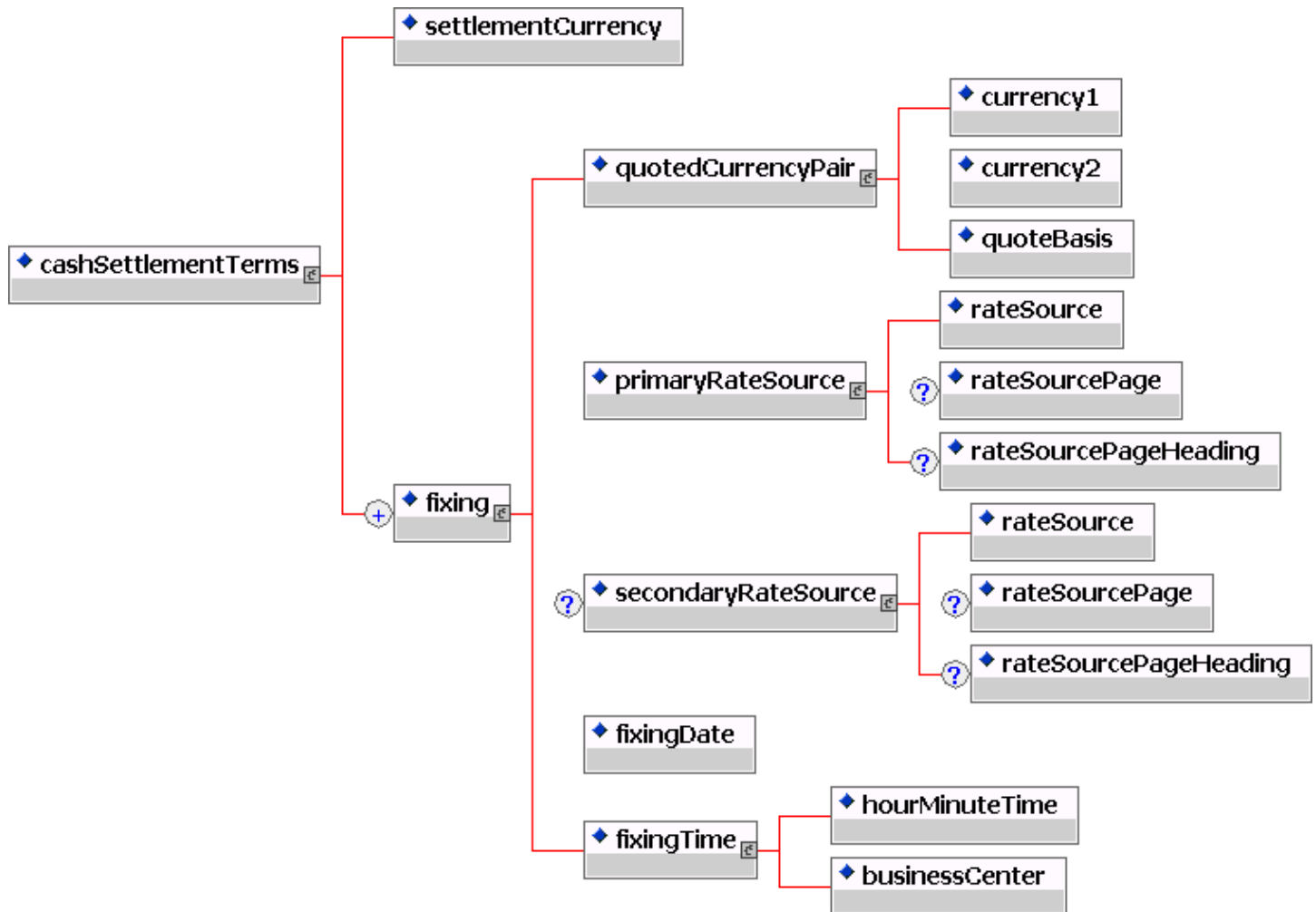

11.3 Foreign Exchange Simple Option

Foreign exchange simple options include standard European and American options. These are commonly referred to as "vanilla," or non-exotic, options. fxSimpleOption identifes the put currency and amount and call currency and amount, as well as the exercise style and premium information. The premium is structured similar to an exchanged currency for a conventional FX trade, where optional settlement information for the premium can be attached. In addition, there is an optional quotedAs structure that contains information about how the option was originally quoted, which can be useful. Below are the structures for a conventional FX OTC option.
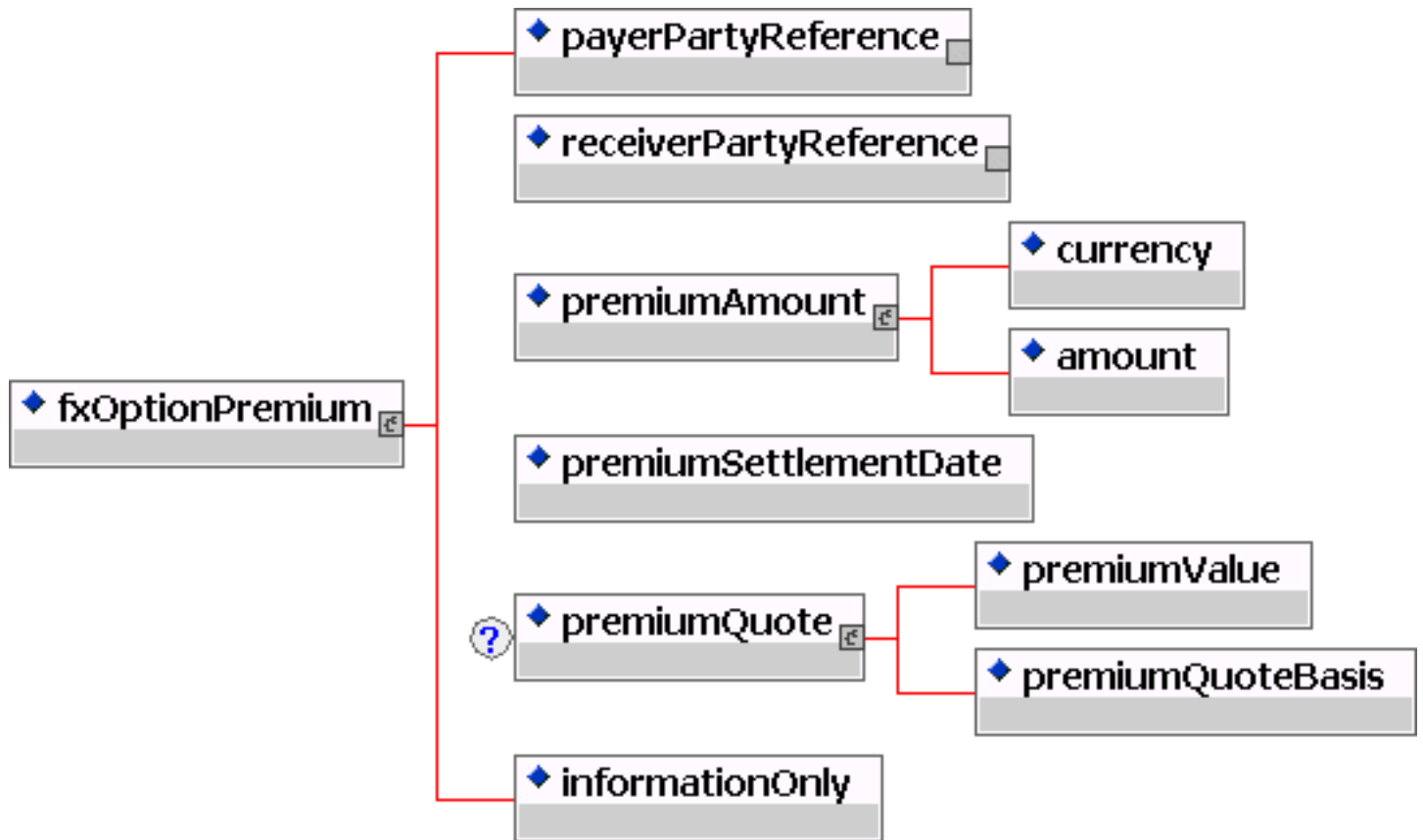
Non-deliverable options are also supported by including the FpML_CashSettlement, which is the identical structure used within non-deliverable forwards.

cashSettlementTerms

- settlementCurrency
- fixing
  - quotedCurrencyPair
    - currency1
    - currency2
    - quoteBasis
  - primaryRateSource
    - rateSource
    - rateSourcePage
    - rateSourcePageHeading
  - secondaryRateSource
    - rateSource
    - rateSourcePage
    - rateSourcePageHeading
  - fixingDate
  - fixingTime
    - hourMinuteTime
    - businessCenter

11.4 Foreign Exchange Barrier Option

A conventional option except that it is changed in a predetermined way when the underlying trades at predetermined barrier levels. A knock-in option pays nothing at expiry unless at some point in its life the underlying reaches a pre-set barrier and brings the option to life as a standard call or put. A knock-out option is a conventional option until the price of the underlying reaches a pre-set barrier price, in which case it is extinguished and ceases to exist. Barrier options have both a strike price and a barrier price.

An optionalbarrierTypeScheme is used to allow for differentiation between knockin, knockout, reverse knockin, and reverse knockout options. One or more barriers are supported. The reference spot rate, while optional, is recommended, as it determines whether the option need to go up or down (or is "out-of-the-money" or "in-the-money") in order to hit the barrier. Additionally, the payout is utilized to accommodate rebates when a barrier is hit. Below are the structures for an FX OTC barrier option.
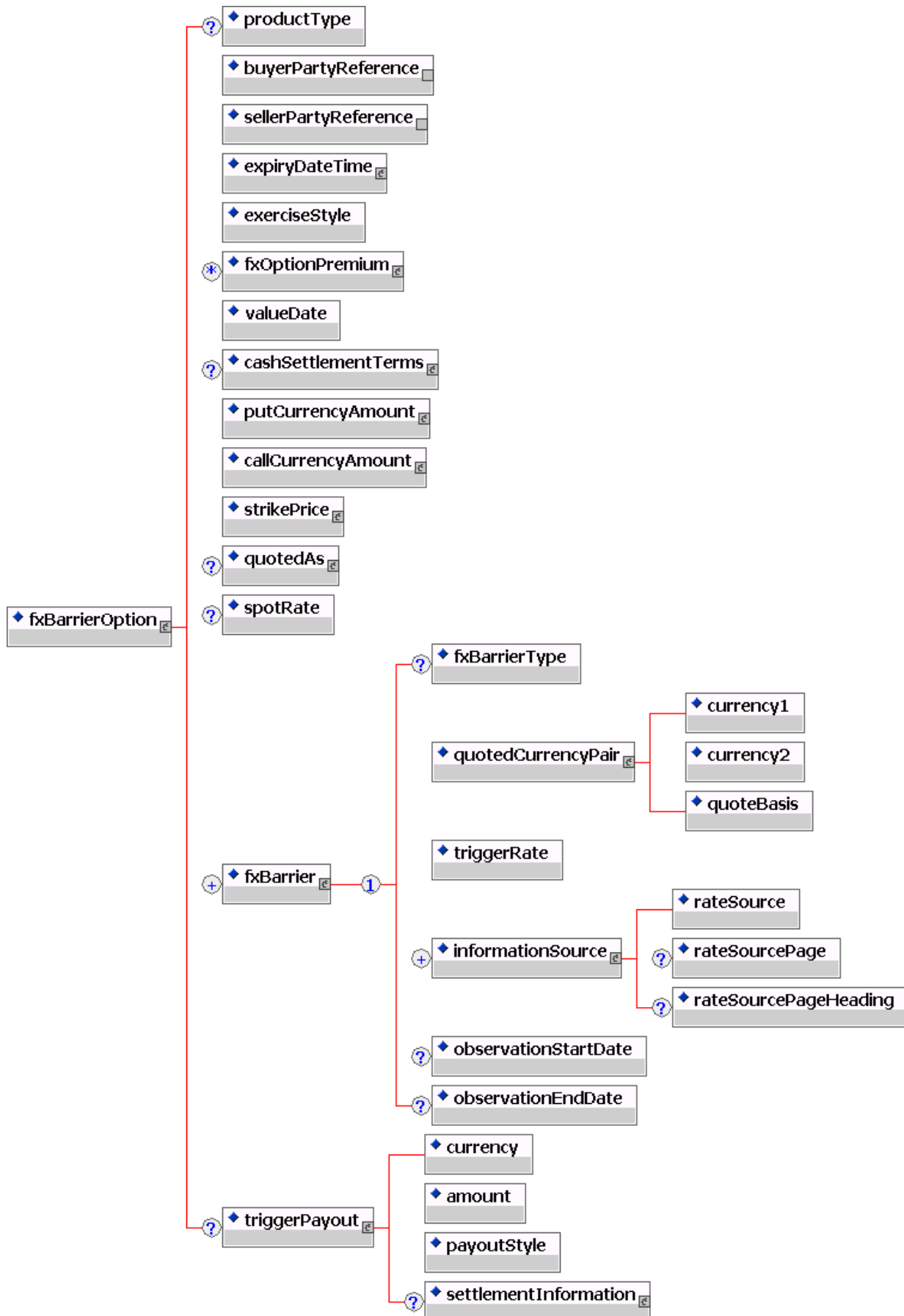
11.5 Foreign Exchange Binary and Digital Options

The terms binary and digital are not clearly defined in the FX markets and can occasionally be synonomous. This is used to define an option that has a discontinuous payout profile. It typically pays out a fixed amount if the underlying satisfies a predetermined trigger condition or else pays nothing. Unlike the standard option, the amounts quoted are the payout amounts as opposed to a notional underlying amount.

Digital options typically are defined as being European, meaning the observation occurs only if the spot rate trades above (or below) the trigger level on expiry date. The two examples that have been included in the specification are the digital and the range digital.

Binary options, on the other hand, are more like American options, meaning that the payout occures if the spot rate trades through the trigger level at any time up to and including the expiry date. The four examples that have been included in the specification are the one-touch, no-touch, double one-touch, and double no-touch binary options. Below are the structures for FX OTC binary and digital options.
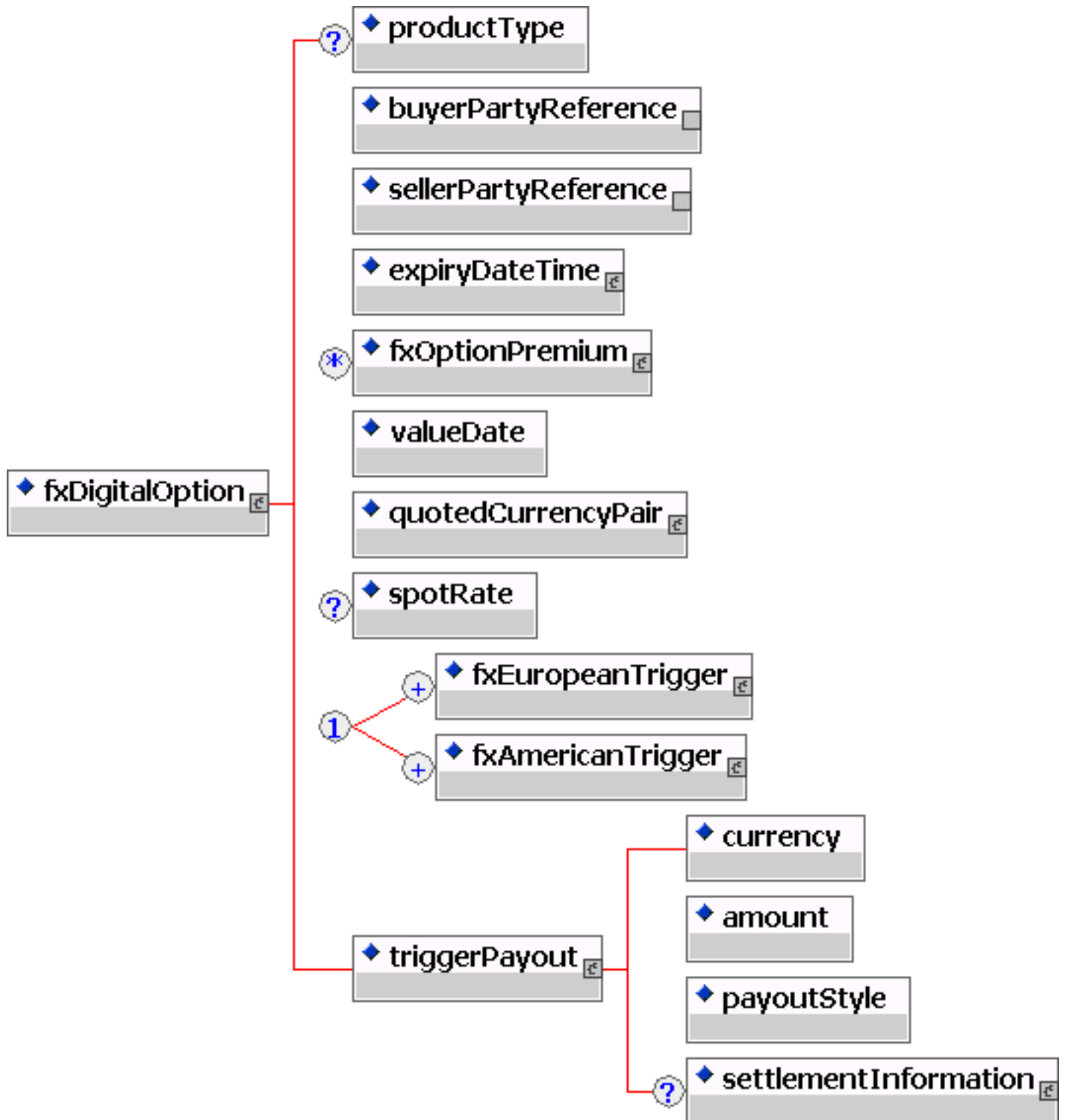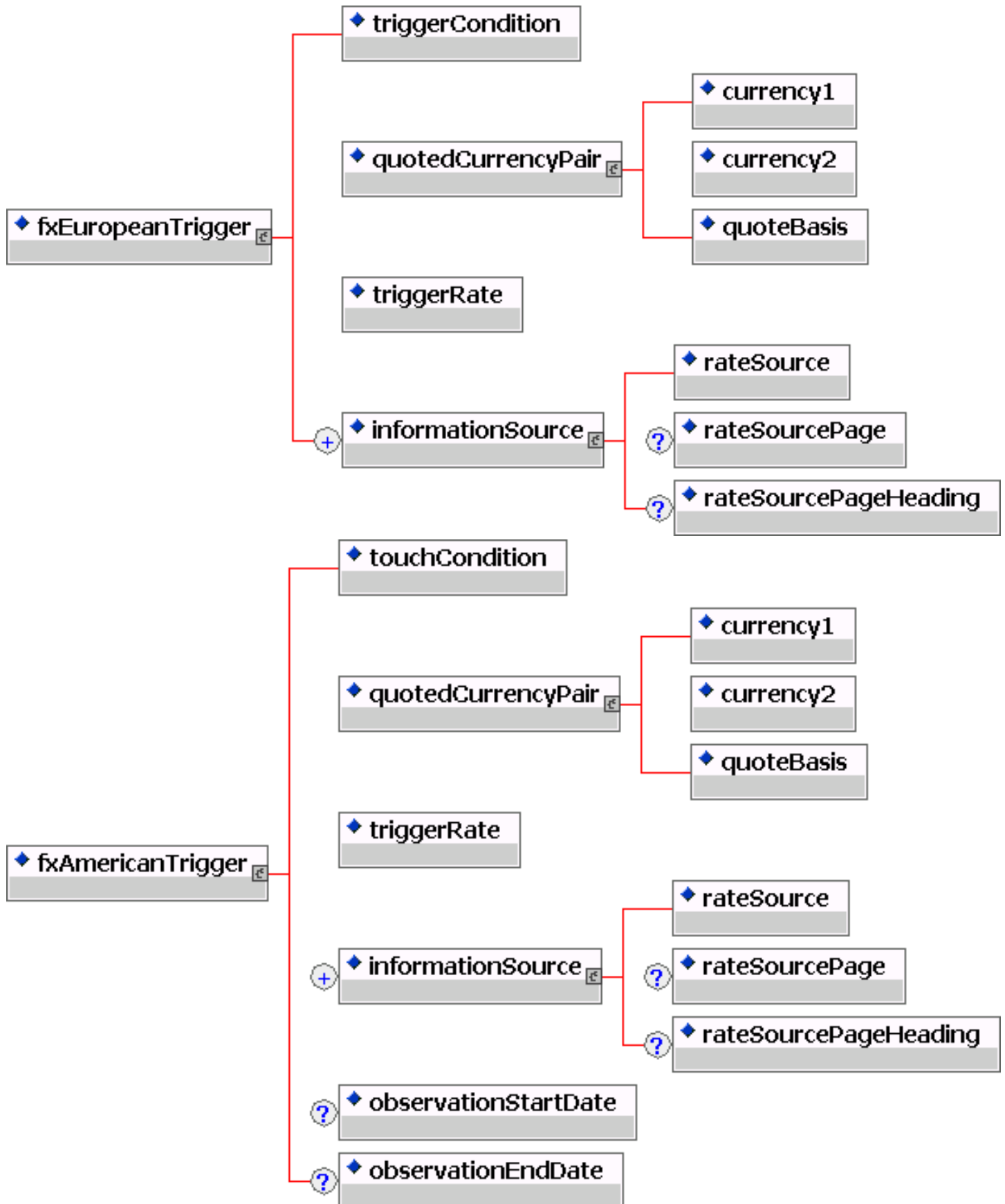
11.6 Foreign Exchange Average Rate Option

Foreign exchange average rate options (sometimes referred to as Asian options) are options whose payout is based upon the average of the price of the underlying, typically (but not necessarily) over the life of the option. Average rate options are popular because they are usually cheaper than conventional options because the averaging process over time reduces volatility.

fxAverageRateOption allows for either a parametric representation of the averaging schedule (e.g., daily, 2nd Friday, etc.), utilizing the same rolling convention schemes as utilized within the interest rate derivatives area. Alternatively, each specific averaging period can be identified, along with a specific weighting factor for that period. In addition, average rate options on occasion, when struck, already have agreed-upon rate observations in the past; the structure accommodates this as well.
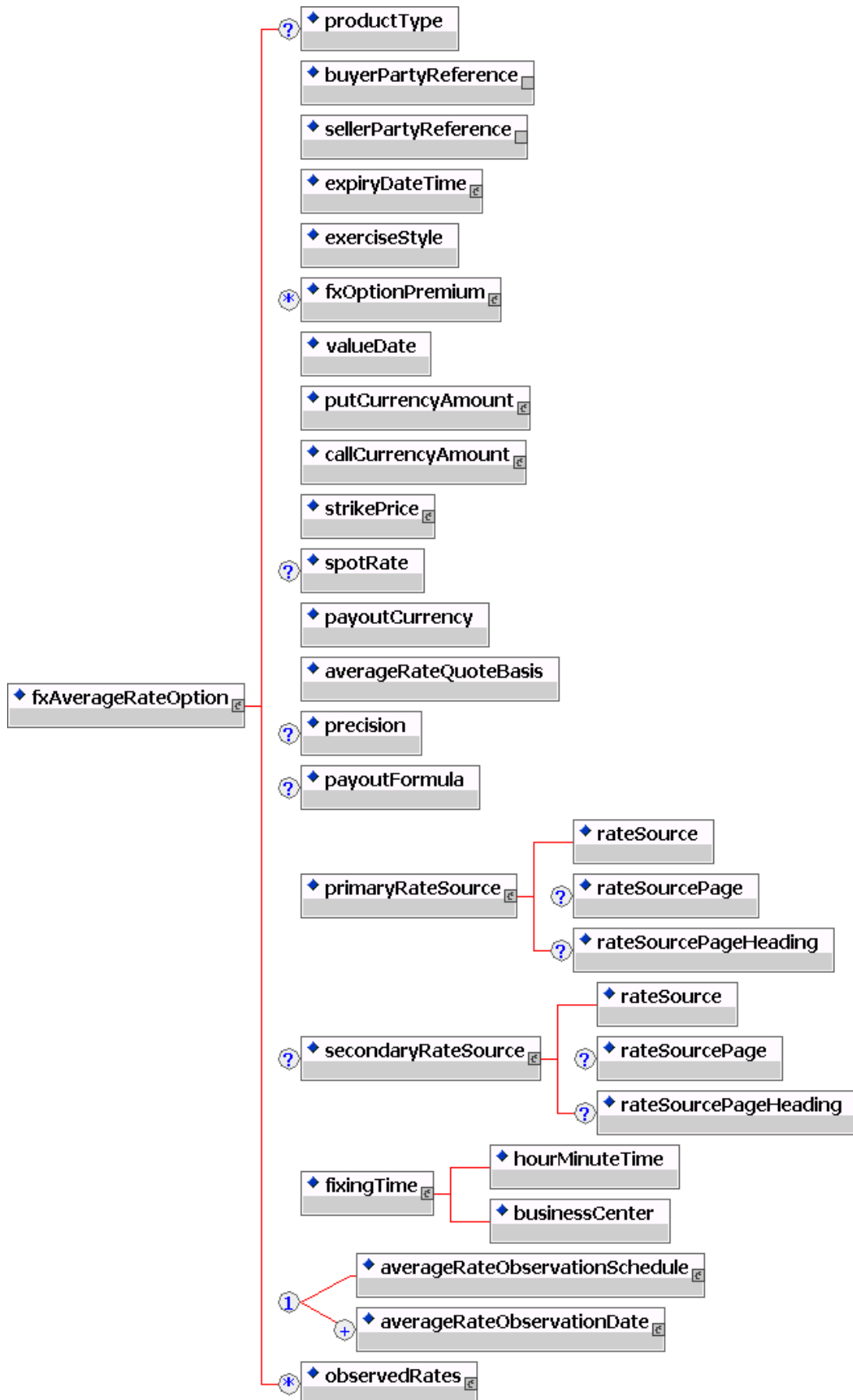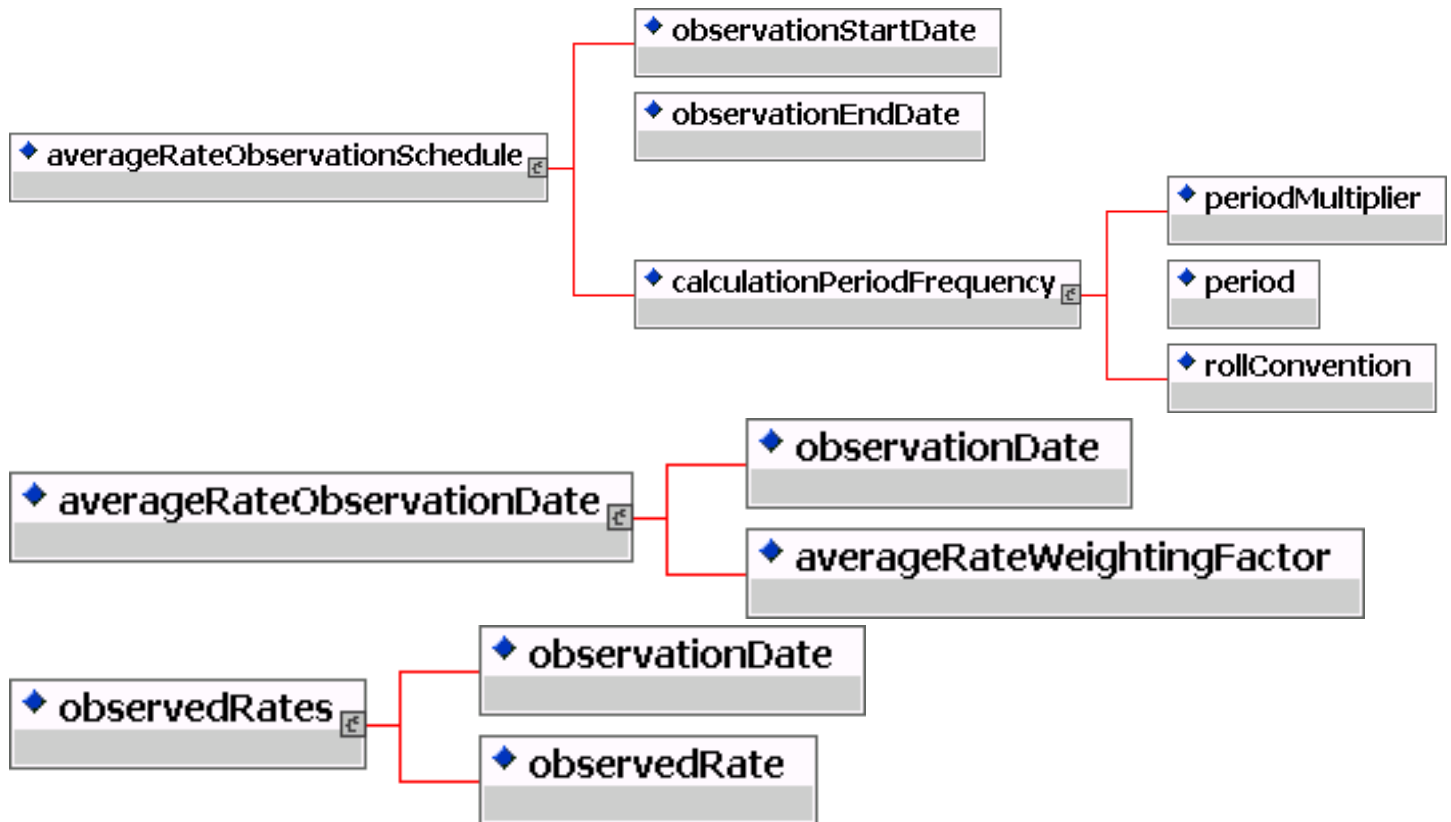
11.7 Trade Strategies

One or more financial instruments, of any sort that are supported by the FpML specification, can be combined to form what is called a strategy. This can include various a package of the same or different asset classes in a single trade. Typical examples of this would include option packages (e.g., straddles, strangles) or a delta hedge (FX OTC option with spot risk hedged by FX spot deal). Additionally, other asset classes can be combined in a strategy (e.g., interest rate swap with FX, etc.).

12 EQUITY DERIVATIVE PRODUCT ARCHITECTURE

Significant effort has been devoted to ensure that the Equity Derivative Product Architecture is fully aligned with ISDA definitions. This has resulted in a legible and compact technical expression of the supported product set.

All definitions are ISDA definitions unless explicity stated otherwise in documentation, and it is not the intention of this section to re-state them.

12.1 Equity Option

Equity Option products comprise Vanilla Put and Call options, with American or European Exercise styles, on Single Stocks or Indices, with delivery being either cash or physical stock.

It derives from a standard product (FpML_Product), which it extends to include buyerParty, sellerParty, optionType, underlying, strike, spotPrice (optional), notional (optional), numberOfOptions (optional), optionEntitlement, equityExercise, equityPremium, methodOfAdjustment (optional), extraordinaryEvents (optional)