elements

FX Rates Package

TARMS Inc.

September 07, 2000

Typeset in LaTeX.

Contents

List of Figures

List of Tables

Package Description

Foreign Exchange rates, or *FX rates*, are rates used to exchange an amount of one currency for an amount of another currency.

European Monetary Union The EMU regulations require that the currencies within the EMU be treated in a particular manner. The regulations can be enforced by constructing a CommonCommodityConstraintGroup using the EUR as the common commodity and a set of FXCommodityConstraintModel §4.7 constraints.

1 Use Cases

1.1 Full Rate

A full rate is a complete FX rate. The rate is usually quoted as being between a *commodity currency* and a *counter currency*, with the rate expressing the number of variable units of the counter currency that one fixed unit of the commodity currency would buy.

Rates are usually quoted as bid/ask pairs. The *bid rate* gives the rate for buying the counter currency or selling the commodity currency. The *ask rate* gives the rate for selling the counter currency or buying the commodity currency. Bid/ask rates are often expressed as a pair, with only the differences in digits being shown — eg., 5.2314/20 means a bid rate of 5.2314 and an ask rate of 5.2320.

Rates may be *direct* or *indirect*. A direct rate is the usual quotation convention and gives the rate in terms of the number of counter currency units a unit of the commodity currency would be exchanged for. An indirect rate gives the rate in terms of the number of commodity currency units a unit of the counter currency would be exchanged for.

As an example, the GBP/USD rate might be 1.6437/47 This means that you could:

- Sell 1000000 pounds in exchange for 1643700 dollars.
- Buy 1000000 dollars in exchange for $1000000/1.6437 = 608383.52$ pounds.
- Buy 1000000 pounds in exchange for 1644700 dollars.
- Sell 1000000 dollars in exchange for $1000000/1.6447 = 608013.62$ pounds.

1.2 Margin Rate

A margin rate represents some additional rate over an existing rate. Margins are added for a number of reasons, discussed in specialized use cases.

Margins are usually expressed in terms of *points*, an addition or subtraction in terms of the smallest unit that the rate is expressed in.

As an example, suppose the spot rate for USD/ISK is 70.43/53 with 1 week forward points of +73/+83. The exchange rate at 1 week is 71.16/36.

1.3 Forward Margin

A *forward margin* is a margin rate that gives the difference between a rate at the spot date and a rate at some date before or after the spot date.

1.4 Corporate Points

The *corporate points* of an FX deal represent the additional margin added (or subtracted) to a deal between a bank and a corporate customer, as opposed to a deal between two banks, to act as a profit margin. Corporate points are a common profit margin; other profit margins might include margins between local and regional trading centers and funding margins.

1.5 Spot Date

FX rates are usually quoted for a date two business days ahead of the current day — the *spot date*. Rates quoted for the spot date are called *spot rates*.

When calculating spot dates, the holidays and weekends of the following locations are considered:

- Any holidays and weekends associated with the location of the trader.
- Any holidays and weekends associated with the location of the counterparty.
- Any holidays and weekends associated with the commodity or counter currency's country of origin.
- If the rate is a cross rate, any holidays and weekends associated with the crossing currency's country of origin.

For quoted rates, the trader and counterparty are usually not considered.

As an example, consider an GBP/EUR trade on 24-Dec-2002. Two business days from 24-Dec-2002 is 28-Dec-2002, as the 25th is a holiday in both the UK and Europe generally and the 26th is a holiday in the UK.

1.6 USD Spot Date

In trades involving the USD, the spot date is two non-weekend days after the trade date, rather than two business days.

1.7 CAD Spot Date

In spot trades involving the CAD, the spot date is one business day after the trade date, rather than two business days.

2 Interfaces

2.1 CurrencyPair

Currency pairs provide information on the standard quotation conventions for an FX rate. Currency pairs are useful when unadorned rate information is supplied (for example, in a trade entry screen) or when sensible defaults are needed.

Generally FX rates for a pair of currencies are quoted using one currency as the commodity currency and one as the counter currency. A currency pair indicates the order in which a pair of currencies is conventionally quoted. Currency pairs also provide information on the quotation mechanism used: the number of decimal points that the rate is normally quoted to, the point size, etc.

2.1.1 Relationships

	Class	Description	Notes
↓	CurrencyPairModel §4.6		
↔	CurrencyPairAuthorityModel §4.4	specifiedCurrencyPairs 1..1	

↓:Realized by ↔:Association →:Navigable ◊:Aggregate ⧫:Composite

2.1.2 Operations

Currency commodityCurrency()

commodityCurrency

The commodity currency.

Return the currency which is usually quoted first in a rate pair. This currency usually forms the denominator in the rate fraction.

Currency counterCurrency()

counterCurrency

The counter currency.

Return the currency which is usually quoted second in a rate pair. This currency usually forms the numerator in the rate fraction.

Number accuracy()

accuracy

The smallest amount to which an FX rate is conventionally quoted to. For example, an accuracy of 0.0001 means that rates are conventionally quoted to 4 decimal places.

Number bigFigureAccuracy()

bigFigureAccuracy

The smallest change that constitutes a change in the *big figure*. A change in the big figure is usually interpreted as a significant change in the exchange rate.

The big figure is commonly 100 times the point size. For example, if the EUR/USD is 1.0137, then a change to 1.0245 represents a “big figure” change; 0.01 is the big figure accuracy.

Number pointSize()

pointSize

The size of a single point. For example, a point size of 0.0001 means that each point adds or subtracts 0.0001 to or from a full rate.

Boolean isDirect()

isDirect

Return true if this rate is quoted directly (ie. in terms of the amount of counter currency one unit of the commodity currency will buy). Return false if this rate is quoted indirectly (ie. in terms of the amount of commodity currency one unit of the counter currency will buy).

Period spotPeriod()

spotPeriod

Return the period that is used to calculate the spot date for this currency pair.

2.2 CurrencyPairAuthority

A currency pair authority provides currency pair (quotation convention) information for pairs of currencies. An instance of this class is a purely internal “authority”; there is normally no reference to an external authority.

Pairs of currencies are generally quoted according to a fixed set of conventions. The commodity currency is, in order of preference, the EUR, a commonwealth currency, the USD or any other currency. Currency pairs are usually arranged so that the commodity currency is the stronger currency.[1]

Most locales, however, usually have a convention that the local currency comes before any other currency. As a result, many currency pair authorities are possible, with each local area having its own conventions.

[1] And, therefore, the exchange rate will be greater than 1.

2.2.1 Relationships

	Class	Description	Notes
⇑	Identifiable		
↓	CurrencyPairAuthorityModel §4.4		
↓	CurrencyPairAuthorityReferenceDataModel §4.5		
↔	CurrencyPairAuthorityReferenceDataModel §4.5	model 0..1	
↔	FXRateSpecifierModel §4.16	authority 0..n	
↔	FXCurveSpecificationModel §4.8	currencyPairAuthority	

⇑:Inherits ↓:Realized by ↔:Association →:Navigable ◊:Aggregate ⧫:Composite

2.2.2 Operations

CurrencyPair pairFor(Currency first, Currency second) pairFor
first: Currency The first currency that needs a currency pair.
second: Currency The second currency that needs a currency pair.

Return the currency pair that is to be used for the two argument currencies.

«Static Method» CurrencyPairAuthority local() local
Return the currency pair authority that is used locally within this system.

2.3 FXCurve

An exchange rate curve is a specialization of a rate curve for FX rates. FXCurves are specific to two currencies and a series of forward dates and return an exchange rate for the two currencies at a specific date.

Non-spot FX rates are conventionally expressed as a spot rate and forward points. For this reason, FX curves are built by getting the forward points for a particular date from the curve and then adding a separate spot rate to build a complete rate.

2.3.1 Relationships

	Class	Description	Notes
⇑	RateCurve		
⇓	BasicFXCurve §2.4		
⇓	CrossFXCurve §2.5		
↔	FXCommodityConstraintModel §4.7	fixing 0..n	
↔	CrossFXCurveModel §4.3	firstCurve 0..n	
↔	CrossFXCurveModel §4.3	secondCurve 0..n	

⇑:Inherits ⇓:Inherited by ↔:Association →:Navigable ◊:Aggregate ⧫:Composite

2.3.2 Operations

Collection<FormalRateParameter> formalParameters() formalParameters

Return the first and second currency formal parameters and the date basis formal parameter. See the FXRateSpecifier §2.13 class for details.

Date spotDate() spotDate

The date at which spot occurs.

Return the spot date for this currency pair, calculated by adding the spot period returned by the currency pair to the current processing date. The currency pair is derived from the curve's specifier.

FXPointRate at(Date date) at

date: Date Date for the exchange rate.

The exchange rate for a forward date.

Return the rate that this curve supplies at the given date. The BasicFXCurve §2.4 interface is designed to always return the rate as full rate in spot+forward margin form.

PointRate value(Collection<ActualRateParameter> parameters) value

parameters: Collection<ActualRateParameter> The parameters that fix the rate.

Get the value at some point on the curve.

Return the result of the at(), using the "date" parameter as an argument.

2.4 BasicFXCurve

A basic rate curve that is an FX curve. An FX curve is a curve that plots the relationship between two currencies over time. The curve is defined by a spot rate and a serious of forwardPoints (derived from a rate feed or calculated from the respective interest rates of the two currencies) and margins (eg. corporate margin, profit margin, etc.). This curve allows an FX value to be calculated for any point in time. These points can then be added to a separate spot rate.

2.4.1 Relationships

	Class	Description	Notes
⇑	FXCurve §2.3		
⇑	BasicRateCurve		
↓	BasicFXCurveModel §4.1		

⇑:Inherits ↓:Realized by

2.4.2 Operations

FXPointRate spotRate() spotRate

The rate on the spot date.

The rate returned must be a full rate with no forward points. The date of the rate must correspond to the date returned by the spotDate method.

FXPointRate forwardPoints(Date date) forwardPoints
date: Date The date for the forward points.

Forward points for a date.

Return the forward points from spot for this date.

FXPointRate at(Date date) at
date: Date Date for the exchange rate.

The exchange rate for a forward date.

Return the rate built by combining the result of the forwardPoints operation for this date with the result of the spotRate operation.

2.5 CrossFXCurve

An FX curve that is built by crossing the result of two other FX curves.

Note that there is no specification static data, as with BasicFXCurve §2.4. Cross FX rate curves are constructed on an as-needs basis by combining two rate curves with a common currency.

2.5.1 Relationships

	Class	Description	Notes
⇑	FXCurve §2.3		
⇑	ImpliedRateCurve		
↓	CrossFXCurveModel §4.3		

⇑:Inherits ↓:Realized by

2.5.2 Operations

FXCurve firstCurve()

firstCurve

The first curve to use when building the cross rate.

FXCurve secondCurve()

secondCurve

The second curve to use when building the cross rate.

FXPointRate at(Date date)

at

date: Date Date for the exchange rate.

The exchange rate for a forward date.

Return a cross rate constructed by combining the rate returned by the first curve with the rate returned by the second curve at the supplied date.

RateConstructor constructor()

constructor

The constructor used to build the cross rate.

Return nil. Cross rate curves have a consistent construction method.

Collection<Rate> sources()

sources

The source rates.

Return a collection consisting of the first curve and the second curve.

2.6 FXCurveSpecification

FX curve specifications consist of a list of the elements needed to build the curve.

2.6.1 Relationships

	Class	Description	Notes
⇑	Identifiable		
⇑	Validatable		
⇓	BasicFXCurveSpecification §2.7		
↓	FXCurveSpecificationModel §4.8		
↓	FXCurveSpecificationReferenceDataModel §4.10		
↔	FXCurveSpecificationReferenceDataModel §4.10	model	
↔	BasicFXCurveConstructorService §5.1	specification	

⇑:Inherits ⇓:Inherited by ↓:Realized by ↔:Association →:Navigable ◊:Aggregate ♦:Composite

2.6.2 Operations

Currency firstCurrency() firstCurrency

The first currency of the curve.

Return the currency that acts as the commodity currency in the rates supplied to the curve.

Currency secondCurrency() secondCurrency

The second currency of the curve.

Return the currency that acts as the counter currency in the rates supplied to the curve.

CurrencyPairAuthority currencyPairAuthority() currencyPairAuthority

The currency pair authority for this curve.

CurrencyPair currencyPair() currencyPair

The currency pair.

Return the currency pair derived from the first and second currencies and the currency pair authority.

Collection<RateDefinitionSpecifier> sources() sources

The source rates.

Return the collection of rate specifiers that are used to build this curve. The curve may be constructed from a collection of point rates, using interpolation, or it may be an implied rate curve, constructed from other rate curves. The rate specifiers in the sources collection may specify either points or curves.

DateBasis dateBasis() dateBasis

The date basis for the curve.

Return the date basis used for constructing interpolations and extrapolations on the curve.

Period spotPeriod() spotPeriod

The spot period.

Return the period which, when added to the current processing date, gives the spot date.

2.7 BasicFXCurveSpecification

Basic FX curves are constructed from a series of point FX rates. This interface provides a means of listing the components such as point rates, along with the currencies and date bases used in building the curve.

2.7.1 Relationships

	Class	Description	Notes
⇑	FXCurveSpecification §2.6		

⇑:Inherits

2.8 FXPointRate

A specialization of the PointRate interface to handle FX rates.

2.8.1 Relationships

	Class	Description	Notes
⇑	PointRate		
⇓	BasicFXRate §2.9		

⇑:Inherits ⇓:Inherited by

2.9 BasicFXRate

The basic FX rate interface specifies the functionality expected for single point FX rates ie. the rate between two currencies on a particular date.

Some definition of terms is required to understand the operation buy() and sell() methods of this interface.

- *Rate is mine.* Rates value can be derived internally (but from an external feed) and "belong" to us. This is termed "mine". If the rate is supplied by the counterparty (e.g. when we ask what it will cost to buy XXX and they say 0.6652) then the rate is termed "yours".
- *Quantity > zero*. This is the quantity of commodity being exchanged. If we are selling the commodity then the quantity is < 0, if we are buying it then it is > 0.
- *Quantity Currency = Quote Commodity currency*. Rates are quoted as Commodity Currency/Counter Currency. This tests the currency of the quantity being traded with the currency of the commodity of the quote.
- *Rate is Direct*. See notes on FXRateQuotationMethod interface on the isDirect() operation for full details.

2.9.1 Relationships

	Class	Description	Notes
⇑	BasicPointRate		
⇑	FXPointRate §2.8		
↓	BasicFXRateModel §4.2		
↔	BasicFXCurveModel §4.1	spotRate 0..n	

⇑:Inherits ↓:Realized by ↔:Association →:Navigable ♢:Aggregate ⧫:Composite

2.9.2 Operations

Instrument buy(Instrument quantity) buy
quantity: Instrument The quantity to convert.
Raises: RateConversionException

Buy one quantity of an instrument by paying a quantity of some other instrument.

See PointRate for an overview of this operation.

If this rate is mine, the quantity is greater than zero and the currency in quantity is the commodity currency and the quotation direction is direct then use the bid rate. Any change in a characteristic flips from bid to ask. Another change flips from ask to bid. See description above for full details on the terms used here.

The quantity, in this case, must be a SimpleCashflow with a currency equal to either the commodity or counter currency and a date equal to the date of the rate. If the currency of the supplied quantity is the commodity currency and the direction is direct, then return the quantity multiplied by the rate. Changing the currency to the counter currency, or the direction from direct to indirect changes the multiplication to a division.

As an example, suppose the rate was AUD/USD of 0.6236 at 12-Dec-2002 (direct) and the quantity was 1000000 USD at 12-Dec-2002. The returned quantity would be 1603592.046218 USD at 12-Dec-2002.

Instrument sell(Instrument quantity) sell
quantity: Instrument The quantity to convert.
Raises: RateConversionException

Sell one quantity of an instrument in exchange for a quantity of some other instrument.

See PointRate for an overview of this operation.

If this rate is mine, the quantity is greater than zero and the currency in quantity is the commodity currency and the quotation direction is direct then use the ask rate. Any change in a characteristic flips from ask to bid. Another change flips from bid to ask. See description above for full details on the terms used here.

The quantity, in this case, must be a SimpleCashflow with a currency equal to either the commodity or counter currency and a date equal to the date of the rate. If the currency of the supplied quantity is the commodity currency and the direction is direct, then return the quantity multiplied by the rate. Changing the currency to the counter currency, or the direction from direct to indirect changes the multiplication to a division.

As an example, suppose the rate was AUD/USD of 0.6236 at 12-Dec-2002 (direct) and the quantity was 1000000 AUD at 12-Dec-2002. The returned quantity would be 623600 USD at 12-Dec-2002.

2.10 FXRatePiece

A rate piece specialized to handle FX rates.

2.10.1 Relationships

	Class	Description	Notes
⇑	RatePiece		
↓	FXRatePieceModel §4.11		

⇑:Inherits ↓:Realized by

2.11 FXRateQuotationMethod

Full rates represent a complete FX rate. Full rates are assumed to follow the same direct/indirect and currency order conventions as whatever specifier supplies.[2]

2.11.1 Relationships

	Class	Description	Notes
⇑	QuotationMethod		
⇑	ValueSemantics		
↓	FXRateQuotationMethodModel §4.12		

⇑:Inherits ↓:Realized by

2.11.2 Operations

String type() type

The type of rate that this quotation method is for.
Return "FX Rate".

Boolean isCanonical(FXRateSpecifier specifier) isCanonical

specifier: FXRateSpecifier The specifier to use when checking for canonicality.

Is this rate in canonical form?
Return true.

Boolean isDirect() isDirect

Is this rate quoted directly?
Return true is this rate is a direct quotation, false if indirect.

A rate is quoted directly if the commodity and counter currency are in the same order as that returned by the CurrecyAuthority for this pair of currencies. For example: If the quote is CHF/USD = 0.66666 and the CurrencyAuthority returns

[2] Some FX rates cannot be inverted, for legal reasons. For this reason, FX rates must follow the standard order and direct/indirect conventions.

USD/CHF for this pair of currencies then the rate is "indirect" and this method returns false. If the quote was USD/CHF = 1.50000 then the rate quote is "direct" and the method returns true.

2.12 FXRateQuote

A rate quote specialized to handle FX rates.

2.12.1 Relationships

	Class	Description	Notes
⇑	RateQuote		
↓	FXRateQuoteModel §4.15		

⇑:Inherits ↓:Realized by

2.13 FXRateSpecifier

Foreign exchange rates (FX rates) specify the rate at which two currencies are exchanged. Generally, the spot date, the date at which exchanges are conventionally made, is given special attention.

2.13.1 Relationships

	Class	Description	Notes
⇑	RateFunctionSpecifier		
↓	FXRateSpecifierModel §4.16		
↔	FXCurveSpecificationModel §4.8	sources	◊

⇑:Inherits ↓:Realized by ↔:Association →:Navigable ◊:Aggregate ⧫:Composite

2.13.2 Operations

Collection<FormalRateParameter> formalParameters()

formalParameters

The possible parameters for this rate specifier.
Return the following set of parameters:

Name	Type		Description
firstCurrency	Commodity	discrete	The first currency to be exchanged
secondCurrency	Commodity	discrete	The second currency to be exchanged
date	Date	continuous	The date of exchange
period	Period	discrete	The period of exchange
dateBasis	DateBasis	discrete	The date basis for period calculations

The date is a LogicalRateDateFormalParameter.

Currency firstCurrency() firstCurrency

One of the currencies that this exchange rate exchanges.

Currency secondCurrency() secondCurrency

One of the currencies that this exchange rate exchanges. This currency is a currency other than the first currency.

Currency currencyOtherThan(Currency currency) currencyOtherThan

currency: Currency The currency to compare against.

Raises: NotFoundException

The currency other than the supplied currency.

If the first currency is the same as the argument currency, then return the second currency. If the second currency is the same as the argument currency, then return the first currency. If neither currency matches the argument currency, then raise a NotFoundException.

CurrencyPair currencyPair() currencyPair

Return the currency pair for the first and second currencies, using the currency pair authority returned by the currencyPairAuthority method.

Date date() date

The date at which the rate occurs.

If the period is defined, then the date is equal to the current processing date with the period added to it.

DateBasis dateBasis() dateBasis

The date basis for use when calculating day and year counts.

This parameter is used when calculating interpolations in exchange rate curves.

Period period()

period

The period at which the rate occurs.

Collection<ActualRateParameter> actualParameters()

actualParameters

The set parameters for this rate specifier.

Return the collection of actual parameters corresponding to the formal parameters by matching the formal parameters against the operations of the same name. Do not include any parameters whose operations return nil.

CurrencyPairAuthority currencyPairAuthority()

currencyPairAuthority

The currency pair authority.

Return the CurrencyPairAuthority §2.2 used to derive standard quotation conventions for this rate.

3 Service Interfaces

3.1 FXRateCurveConstructor

An FX rate curve constructor is used to build a basic FX rate curve out of a series of FX point rates. A spot rate must be supplied as one of the point rates.

Once collected, these rates can be used to build a rate curve by stripping the spot rate from the supplied rates, leaving a set of forward points. These forward points can then be interpolated by a suitable curve constructor.

3.1.1 Relationships

	Class	Description	Notes
⇑	RateConstructor		
↓	BasicFXCurveConstructorService §5.1		

⇑:Inherits ↓:Realized by

4 Classes

4.1 BasicFXCurveModel

An implementation of the BasicFXCurve interface using the BasicRateCurveModel as a constructor. The curve is used to build the forward points for the rate. This is then added to a separate spot rate.

The elements of the curve are restricted to being BasicFXRates with no base pieces in their components (ie. forward margins only). The spot rate provides the base piece for the final rate.

4.1.1 Relationships

	Class	Description	Notes
⇑	BasicRateCurveModel		
↑	BasicFXCurve §2.4		
↔	BasicFXRate §2.9	spotRate 1..1	→

⇑:Inherits ↑:Realizes ↔:Association →:Navigable ◊:Aggregate ♦:Composite

4.1.2 Operations

FXPointRate forwardPoints(Date date) forwardPoints

date: Date The date for the forward points.

Forward points for a date.

Return the value of the rate curve, with the date restricted to the supplied parameter.

4.2 BasicFXRateModel

A concrete implementation of the BasicFXRate interface. This class is restricted to components that implement the FXRateQuote §2.12 interface.

4.2.1 Relationships

	Class	Description	Notes
⇑	BasicPointRateModel		
↑	BasicFXRate §2.9		

⇑:Inherits ↑:Realizes

4.3 CrossFXCurveModel

An implementation of the CrossFXCurve interface. The two constructor curves are held as associations.

4.3.1 Relationships

	Class	Description	Notes
↑	CrossFXCurve §2.5		
↔	FXCurve §2.3	firstCurve 1..1	→
↔	FXCurve §2.3	secondCurve 1..1	→

↑:Realizes ↔:Association →:Navigable ◊:Aggregate ♦:Composite

4.4 CurrencyPairAuthorityModel

A concrete implementation of the CurrencyPairAuthority interface. Currency pair information is usually built algorithmically, with exceptions allowed for difficult cases.

4.4.1 Relationships

	Class	Description	Notes
↑	CurrencyPairAuthority §2.2		
↔	CurrencyPair §2.1	specifiedCurrencyPairs 0..n	→

↑:Realizes ↔:Association →:Navigable ◊:Aggregate ♦:Composite

4.4.2 Attributes

identifier: String The unique identifier for the authority.

4.4.3 Operations

CurrencyPair pairFor(Currency first, Currency second) pairFor
first: Currency The first currency that needs a currency pair.
second: Currency The second currency that needs a currency pair.

Return the currency pair that is to be used for the two argument currencies.

If the first and second currencies match the commodity and counter currencies of an associated exception currency pair, then return that currency pair.

Otherwise, construct a currency pair according to the following algorithm:

- The commodity currency is the currency with the highest priority (lowest number). See the extensions to Currency in this package for a discussion of priority. The counter currency is the currency with the lowest priority. If the two currencies have equal priorities, then the commodity currency is the currency with the smallest alphabetical identifier.

- If both currencies have a smallest exchangeable amount less than 1, then the accuracy is 0.0001. Otherwise, the accuracy is 0.01.
- The bigFigureAccuracy is the accuracy times 100.
- The pointSize is the same as the accuracy.
- The quotation method is direct.
- If one of the currencies is CAD, then the spot period is a period of one business day. Otherwise, if one of the currencies is USD, then the spot period is a period of two non-weekend days. Otherwise, the spot period is a period of two business days. No rolling convention is used.[3]

4.5 CurrencyPairAuthorityReferenceDataModel

An implementation of the CurrencyPairAuthority interface as reference data. The CurrencyPairAuthority operations are delegated to the associated model.

4.5.1 Relationships

	Class	Description	Notes
⇑	ReferenceDataModel		
↑	CurrencyPairAuthority §2.2		
↔	CurrencyPairAuthority §2.2	model 1..1	→

⇑:Inherits ↑:Realizes ↔:Association →:Navigable ◊:Aggregate ⧫:Composite

4.6 CurrencyPairModel

A concrete implementation of the CurrencyPair interface, with the various elements being stored as attributes.

4.6.1 Relationships

	Class	Description	Notes
↑	CurrencyPair §2.1		
↔	Period	spotPeriod 1..1	→

↑:Realizes ↔:Association →:Navigable ◊:Aggregate ⧫:Composite

[3] These periods may be specially distinguished reference data periods.

4.6.2 Attributes

commodityCurrency: Currency The commodity currency.

counterCurrency: Currency The counter currency.

accuracy: Number The accuracy to quote an FX rate to.

bigFigureAccuracy: Number The accuracy to quote a big figure rate to.

pointSize: Number The size of a single point.

isDirect: Boolean = true This is a direct quotation.

4.7 FXCommodityConstraintModel

FX commodity constraints allow the fixing of currency pairs to fixed exchange rates.

Although the examples of FX constraints (EMU, the USD/HKD peg) are time-independent, in the general case this may not be so. FX commodity constraints, therefore, are expressed in terms of curves, rather than point rates.

4.7.1 Relationships

	Class	Description	Notes
⇑	CommodityConstraintModel		
↔	FXCurve §2.3	fixing 1..1	→

⇑:Inherits ↔:Association →:Navigable ◊:Aggregate ⧫:Composite

4.7.2 Attributes

isSoft: Boolean = false Is this a soft (non-mandatory) constraint?

4.7.3 Operations

Commodity firstCommodity() firstCommodity

The first commodity in the constraint.
Return the first currency of the associated FX curve.

Commodity secondCommodity() secondCommodity

The second commodity in the constraint.
Return the second currency of the associated FX curve.

Boolean isComposable()

isComposable

Can this constraint be composed with other constraints?

Return true.

Boolean isFunctional()

isFunctional

Is this a functional constraint?

Return true.

Boolean predicate(SimpleCashflow amount1, SimpleCashflow amount2)

predicate

amount1: SimpleCashflow The first amount to compare.
amount2: SimpleCashflow The second amount to compare.
Raises: CommodityConstraintException

The constraint.

Amount1 must be a simple cashflow in the currency corresponding to the first commodity. Amount2 must be a simple cashflow in the currency corresponding to the second commodity. Both amount1 and amount2 must share the same date. Raise a CommodityConstraintException is these conditions are not met.

Let r be the rate supplied by the associated FX curve at the date of amount1 and amount2.

Return true if amount1 = r.buy(amount2). Return false otherwise.

SimpleCashflow firstAmountFrom(SimpleCashflow amount)

firstAmountFrom

amount: SimpleCashflow The amount to convert..
Raises: CommodityConstraintException

Convert an amount of the first commodity into an amount of the second commodity.

Amount must be a simple cashflow in the currency corresponding to the second commodity. Raise a CommodityConstraintException if this is not true.

Let r be the rate supplied by the associated FX curve at the date of amount.

Return r.buy(amount).

SimpleCashflow secondAmountFrom(SimpleCashflow amount)

secondAmountFrom

amount: SimpleCashflow The amount to convert..
Raises: CommodityConstraintException

Convert an amount of the first commodity into an amount of the second commodity.

Amount must be a simple cashflow in the currency corresponding to the first commodity. Raise a CommodityConstraintException if this is not true.

Let r be the rate supplied by the associated FX curve at the date of amount.

Return r.sell(amount).

4.8 FXCurveSpecificationModel

A concrete implementation of the FXCurveSpecification interface.

4.8.1 Relationships

	Class	Description	Notes
↑	FXCurveSpecification §2.6		
⇓	BasicFXCurveSpecificationModel §4.9		
↔	Period	spotPeriod	→
↔	CurrencyPairAuthority §2.2	currencyPairAuthority	→
↔	FXRateSpecifier §2.13	sources	→

⇓:Inherited by ↑:Realizes ↔:Association →:Navigable ◊:Aggregate ♦:Composite

4.8.2 Attributes

firstCurrency: Currency The first (commodity) currency of the curve.

secondCurrency: Currency The second (counter) currency of the curve.

dateBasis: DateBasis The common date basis for the curve.

identifier: String The unique identifier for the curve definition.

4.8.3 Operations

Period spotPeriod() spotPeriod

The spot period.

If there is an associated spot period, return that period. Otherwise, return the spot period derived from the currency pair for this curve.

CurrencyPairAuthority currencyPairAuthority() currencyPairAuthority

The currency pair authority for this curve.

If there is an associated authority, return that authority. Otherwise, return the local authority.

4.9 BasicFXCurveSpecificationModel

An FX curve specification where the curve is built from individual point rates. The associated sources must specify point rates.

4.9.1 Relationships

	Class	Description	Notes
⇑	FXCurveSpecificationModel §4.8		

⇑:Inherits

4.9.2 Operations

Reportable validate() validate

Validate the specification.

A specification is valid if:

- The two currencies and date basis are supplied.
- The two currencies are different.
- One source specifier has a period of the spot period.
- No two source specifiers have the same period.
- Each source specifier must satisfy the following requirements:
  - Each source specifier is an FXRateSpecifier §2.13.
  - The first and second currencies of the specifier match the first and second currencies of the specification. The order of the first and second currencies need not be the same.
  - If the derivation is a basic derivation, then there is a quotation method.
  - The date is present.
  - If the specifier has a period that corresponds to the spot date, then the quotation method for that specifier is a full quotation method.
  - The source and the specification have the same currency pair authority.

In addition, add a warning if:

- The specifier is not at the spot period and the quotation method is not a marginal quotation method.
- The period is not defined and a date supplied instead.
- The date basis does not match the date basis of the curve.

4.10 FXCurveSpecificationReferenceDataModel

An FX curve that is managed as a piece of reference data. Operations that are part of the FXCurveSpecification interface are delegated to a held model.

4.10.1 Relationships

	Class	Description	Notes
⇑	ReferenceDataModel		
↑	FXCurveSpecification §2.6		
↔	FXCurveSpecification §2.6	model	→

⇑:Inherits ↑:Realizes ↔:Association →:Navigable ◊:Aggregate ♦:Composite

4.11 FXRatePieceModel

A concrete implementation of the FXRatePiece interface. Quotation methods used by this class must implement FXRateQuotationMethod §2.11 interface.

4.11.1 Relationships

	Class	Description	Notes
⇑	RatePieceModel		
↑	FXRatePiece §2.10		

⇑:Inherits ↑:Realizes

4.12 FXRateQuotationMethodModel

A concrete implementation of the FXFullRateQuotationMethod interface.

4.12.1 Relationships

	Class	Description	Notes
↑	FXRateQuotationMethod §2.11		
⇓	FXFullRateQuotationMethodModel §4.13		
⇓	FXMarginRateQuotationMethodModel §4.14		

⇓:Inherited by ↑:Realizes

4.12.2 Attributes

isDirect: Boolean = true Is this rate a direct quotation?

4.13 FXFullRateQuotationMethodModel

An FX rate that is quoted in full form.

4.13.1 Relationships

	Class	Description	Notes
⇑	FXRateQuotationMethodModel §4.12		

⇑:Inherits

4.13.2 Operations

Boolean isMargin() isMargin

Is this rate in margin form?
Return false.

4.14 FXMarginRateQuotationMethodModel

An FX rate that is quoted in margin form.

4.14.1 Relationships

	Class	Description	Notes
⇑	FXRateQuotationMethodModel §4.12		

⇑:Inherits

4.14.2 Operations

Boolean isMargin() isMargin

Is this rate in margin form?
Return true.

4.15 FXRateQuoteModel

A concrete implementation of the FXRateQuote interface. Instances of this model are only composed of FXRatePiece §2.10 components.

4.15.1 Relationships

	Class	Description	Notes
⇑	RateQuoteModel		
↑	FXRateQuote §2.12		

⇑:Inherits ↑:Realizes

4.16 FXRateSpecifierModel

A concrete implementation of the FXRateSpecifier interface. All relevant parameters are held in (possibly nil) attributes.

4.16.1 Relationships

	Class	Description	Notes
↑	FXRateSpecifier §2.13		
↔	Period	period 0..1	→
↔	CurrencyPairAuthority §2.2	authority 0..1	→

↑:Realizes ↔:Association →:Navigable ◊:Aggregate ♦:Composite

4.16.2 Attributes

firstCurrency: Currency The first currency of the rate.

secondCurrency: Currency The second currency of the rate.

date: Date The date of the exchange.

dateBasis: DateBasis The date basis to use.

4.16.3 Operations

CurrencyPairAuthority currencyPairAuthority() currencyPairAuthority

The currency pair authority.

If there is an associated CurrencyPairAuthority §2.2 then return that. Otherwise return the local CurrencyPairAuthority.

5 Services

5.1 BasicFXCurveConstructorService

An abstract class providing the basis for more specific construction services. This constructor service is associated with an FX specification which lists a series of point rates used to build the curve.

The associated specification must be a BasicFXCurveSpecificationModel §4.9.

5.1.1 Relationships

	Class	Description	Notes
↑	FXRateCurveConstructor §3.1		
⇓	LinearFXCurveConstructorService §5.2		
↔	FXCurveSpecification §2.6	specification	→

⇓:Inherited by ↑:Realizes ↔:Association →:Navigable ◊:Aggregate ♦:Composite

5.1.2 Operations

OrderedCollection<RateFunctionSpecifier> sources() sources

The source rates.

Return the source rates from the associated specification.

Rate construct(OrderedCollection<Rate> sources) construct

sources: OrderedCollection<Rate> The point rates needed to build the curve.

Raises: RateConstructorException

Build the exchange rate curve.

Construction operations are defined in subclasses.

RateFunctionSpecifier result() result

The output rate specification.

Build an FXRateSpecifierModel §4.16 with the first currency being the commodity currency of the associated specifier, the second currency being the counter currency of the associated specifier, the date basis of the associated specifier and a date and period of nil.

5.2 LinearFXCurveConstructorService

An example FX curve constructor. This constructor simply provides linear interpolation between the supplied rates and flattens the curve outside the range of the supplied rates.

5.2.1 Relationships

	Class	Description	Notes
⇑	BasicFXCurveConstructorService §5.1		

⇑:Inherits

5.2.2 Operations

Rate construct(OrderedCollection<Rate> sources) construct

sources: OrderedCollection<Rate> The point rates needed to build the curve.

Raises: RateConstructorException

Build the exchange rate curve.

1. Identify the rate which has a spot period. If there is no spot rate, raise a RateConstructorException.
2. Convert all non-spot rates in full form into margin form by subtracting the spot rate from these rates.
3. Divide the supplied rates into two parts: those before the spot date and those after the spot date. Sort these two collections into date order and add a zero forward point rate at the spot date to each collection.
4. Remove any rate from each collection where the day count between the rate and the previous rate is 0, using the date basis of the specification.

5. Choose the earliest rate, r_0 from the rates before the spot date. Use this rate to construct a flat curve for dates before the date of r_0.

For each rate $r_i : i \geq 1$:

1. Construct a linear interpolation for each component and piece by

$$r_{i-1} + \frac{r_i - r_{i-1}}{d_i - d_{i-1}}(d - d_{i-1})$$

Where $d_2 - d_1$ is the term in years between the dates d_1 and d_2 calculated using the date basis of the specification.

6 Associations

Table 1: FX Rates— Associations

Association				
	Role	Class	Card.	Notes
specifiedCurrencyPairs				
	exception	CurrencyPair §2.1	0..n	→
	authority	CurrencyPairAuthorityModel §4.4	1..1	
model				
	model	CurrencyPairAuthority §2.2	1..1	→
	reference data	CurrencyPairAuthorityReferenceDataModel §4.5	0..1	
period				
	period	Period	0..1	→
	fx rate specifier	FXRateSpecifierModel §4.16	0..n	
authority				
	authority	CurrencyPairAuthority §2.2	0..1	→
	fx rate specifier	FXRateSpecifierModel §4.16	0..n	
spotPeriod				
	spot period	Period	1..1	→
	currency pair	CurrencyPairModel §4.6	0..n	
spotRate				
	spot rate	BasicFXRate §2.9	1..1	→

Table 1: . . . continued

Association				
	Role	Class	Card.	Notes
	rate curve	BasicFXCurveModel §4.1	0..n	
spotPeriod				
	spot period	Period		→
	fx curve specification	FXCurveSpecificationModel §4.8		
currencyPairAuthority				
	authority	CurrencyPairAuthority §2.2		→
	fx curve specification	FXCurveSpecificationModel §4.8		
sources				
	source	FXRateSpecifier §2.13		→
	fx curve specification	FXCurveSpecificationModel §4.8		◊
model				
	model	FXCurveSpecification §2.6		→
	reference data	FXCurveSpecificationReferenceDataModel §4.10		
specification				
	specification	FXCurveSpecification §2.6		→
	constructor	BasicFXCurveConstructorService §5.1		
fixing				
	fixing curve	FXCurve §2.3	1..1	→
	constraint	FXCommodityConstraintModel §4.7	0..n	
firstCurve				
	first curve	FXCurve §2.3	1..1	→
	cross curve	CrossFXCurveModel §4.3	0..n	
secondCurve				
	second curve	FXCurve §2.3	1..1	→
	cross curve	CrossFXCurveModel §4.3	0..n	

→:Navigable ◊:Aggregate ◆:Composite

6.1 specifiedCurrencyPairs

Role: exception *Navigable* CurrencyPair, 0..n.
Role: authority CurrencyPairAuthorityModel, 1..1.

The list of specially defined currency pairs. Other currency pairs are built algorithmically (see pairFor on CurrencyPairAuthorityModel for more details).

6.2 model

Role: model *Navigable* CurrencyPairAuthority, 1..1.
Role: reference data CurrencyPairAuthorityReferenceDataModel, 0..1.
The held CurrencyPairAuthority model for the reference data model.

6.3 period

Role: period *Navigable* Period, 0..1.
Role: fx rate specifier FXRateSpecifierModel, 0..n.
The period that an exchange takes place at.

6.4 authority

Role: authority *Navigable* CurrencyPairAuthority, 0..1.
Role: fx rate specifier FXRateSpecifierModel, 0..n.
The option authority to use when deriving currency pair information.

6.5 spotPeriod

Role: spot period *Navigable* Period, 1..1.
Role: currency pair CurrencyPairModel, 0..n.
The spot period that this currency pair uses.

6.6 spotRate

Role: spot rate *Navigable* BasicFXRate, 1..1.
Role: rate curve BasicFXCurveModel, 0..n.
The spot rate for a basic rate curve.

6.7 spotPeriod

Role: spot period *Navigable* Period.
Role: fx curve specification FXCurveSpecificationModel.
The period that gives the spot date for a FX curve specification.

6.8 currencyPairAuthority

Role: authority *Navigable* CurrencyPairAuthority.
Role: fx curve specification FXCurveSpecificationModel.
The currency pair authority to use for a rate curve.

6.9 sources

Role: source *Navigable* FXRateSpecifier.
Role: fx curve specification *Aggregate* FXCurveSpecificationModel.
The rate specifiers which make up an FX curve.

6.10 model

Role: model *Navigable* FXCurveSpecification.
Role: reference data FXCurveSpecificationReferenceDataModel.
The held model for the reference data.

6.11 specification

Role: specification *Navigable* FXCurveSpecification.
Role: constructor BasicFXCurveConstructorService.

6.12 fixing

Role: fixing curve *Navigable* FXCurve, 1..1.
Role: constraint FXCommodityConstraintModel, 0..n.
The curve used to fix the commodity pair.

6.13 firstCurve

Role: first curve *Navigable* FXCurve, 1..1.
Role: cross curve CrossFXCurveModel, 0..n.
The first curve that makes up the cross rate.

6.14 secondCurve

Role: second curve *Navigable* FXCurve, 1..1.
Role: cross curve CrossFXCurveModel, 0..n.
The second curve that makes up the cross curve.

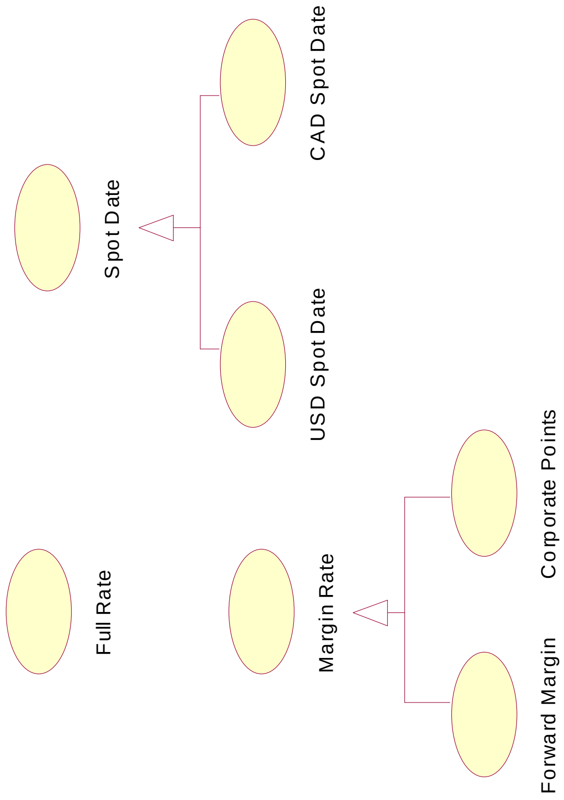

Figure 1: Class Diagram— Examples

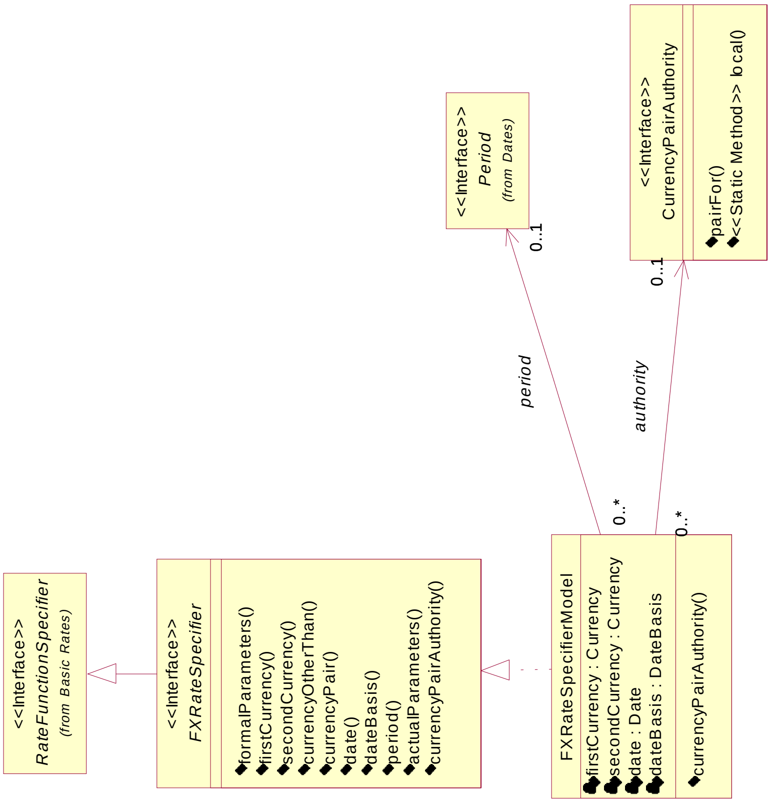

Figure 2: Class Diagram— FX Rate Specification1

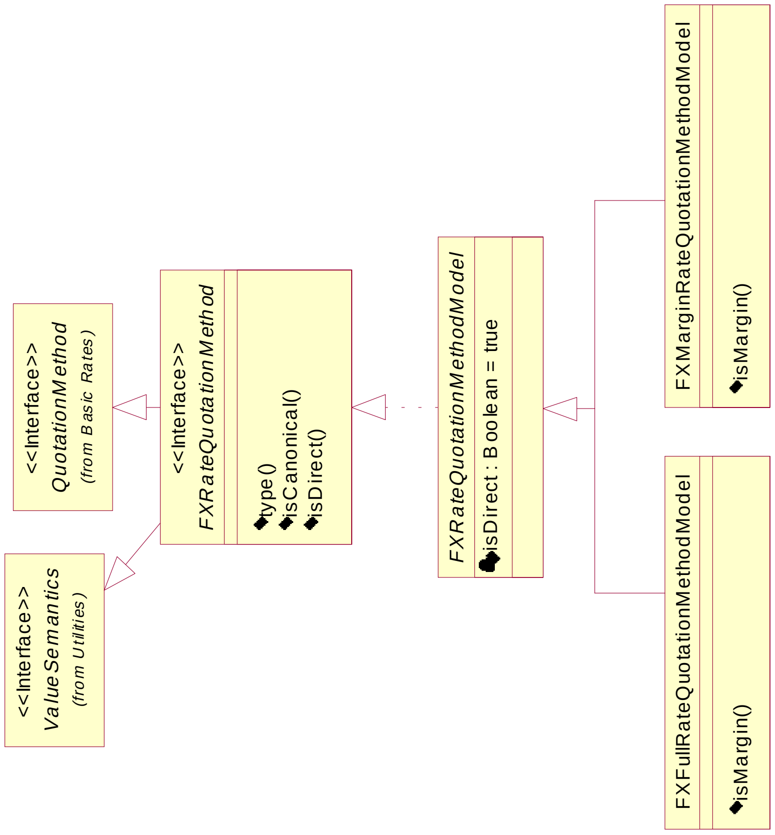

Figure 3: Class Diagram— FX Rate Specification2

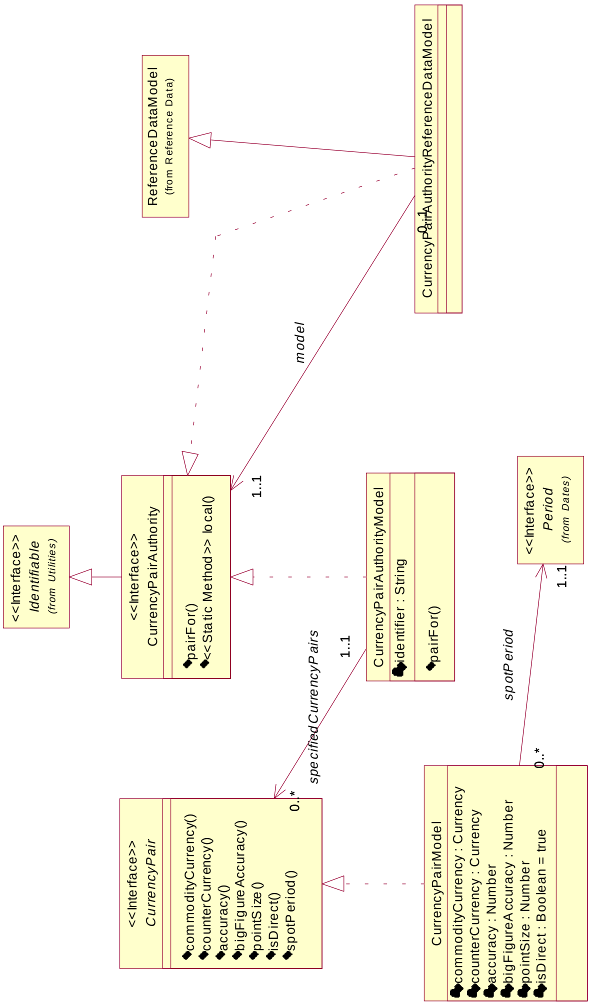

Figure 4: Class Diagram— Currency Pairs

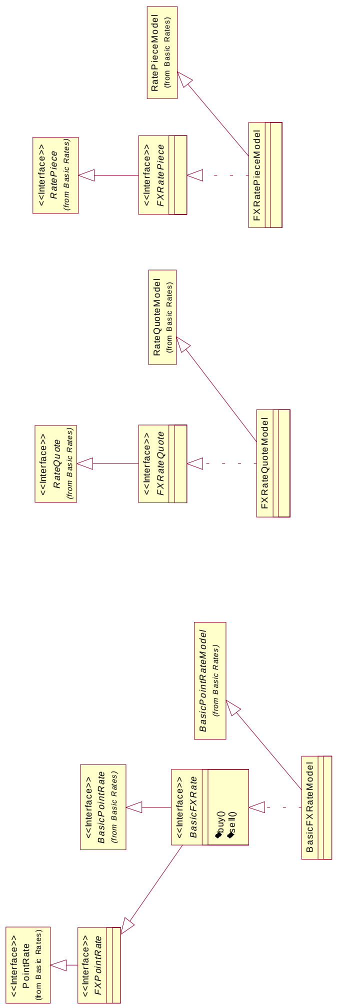

Figure 5: Class Diagram— Point Rates

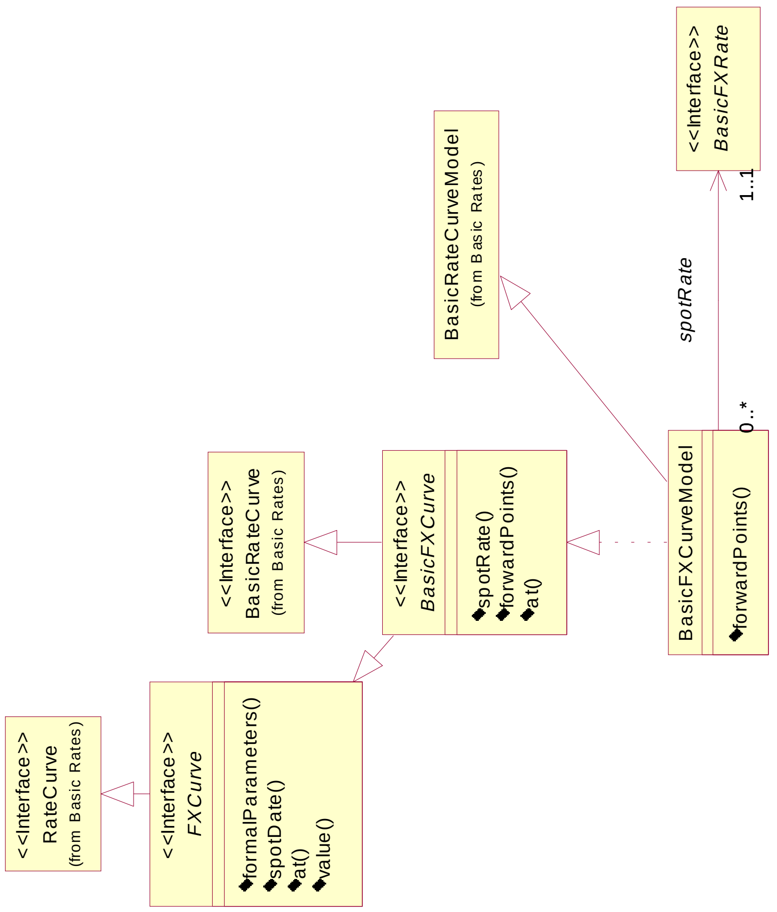

Figure 6: Class Diagram— Basic FX Curves1

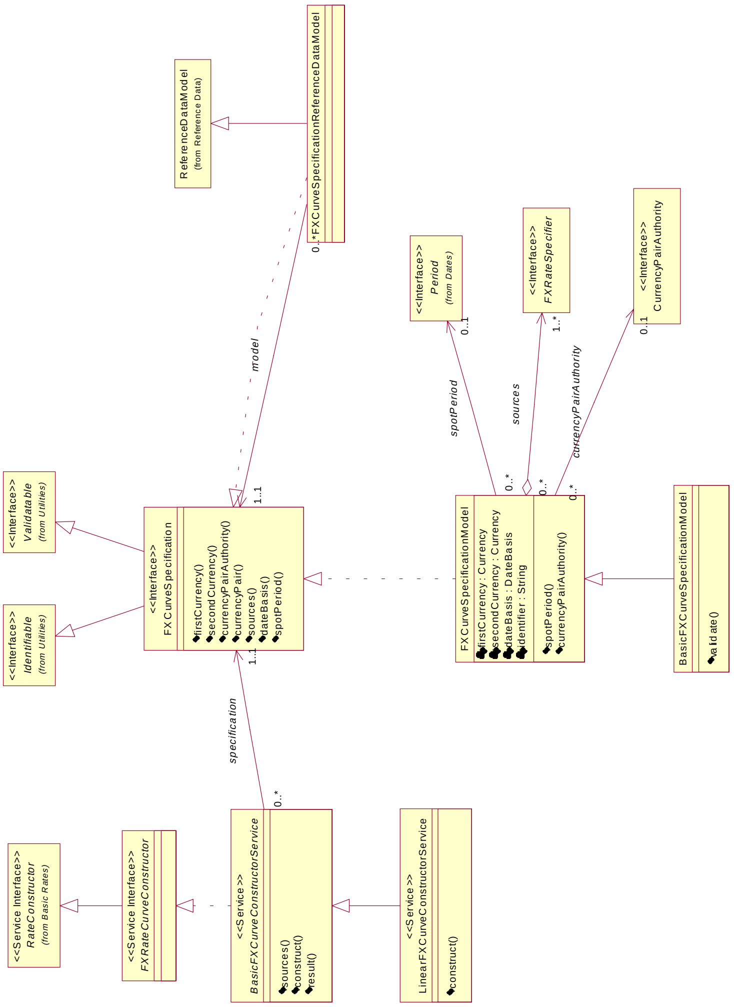

Figure 7: Class Diagram— Basic FX Curves2

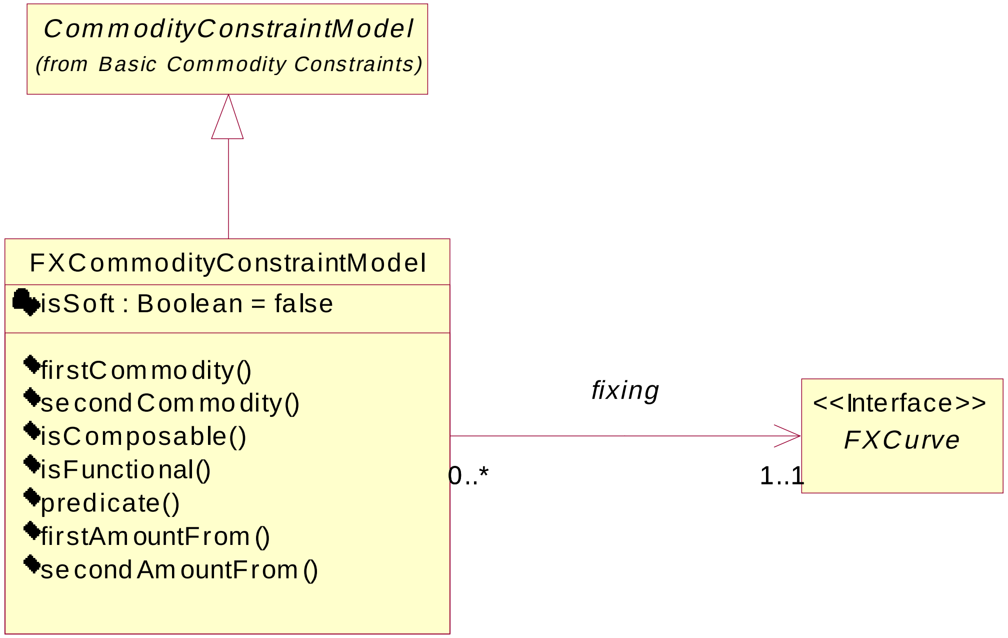

Figure 8: Class Diagram— FX Constraints

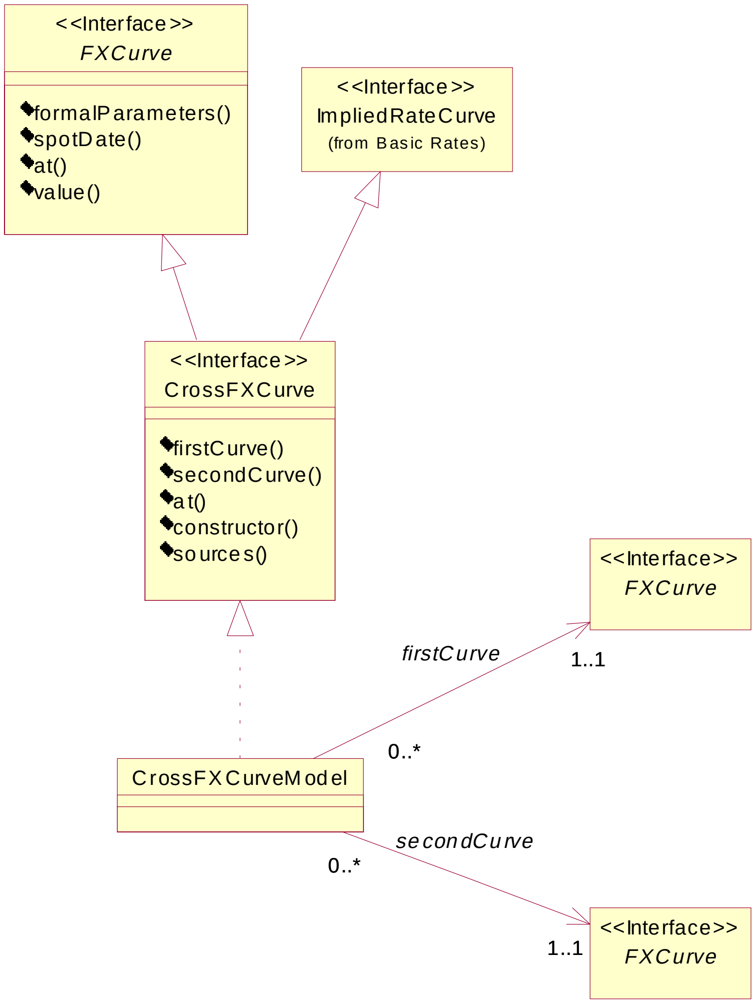

Figure 9: Class Diagram— Cross FX Curves

7 Extensions to the Basic Commodities Package

7.1 Currency

7.1.1 Operations

Integer priority() priority

Return a currency pair priority for this currency, with 1 being the highest priority and 0 being a special case.

For any pair of priorities, the larger number indicates a lower priority. An exception to this rule is zero, which is lower in priority than any non-zero priority.

7.2 CurrencyModel

7.2.1 Attributes

priority: Integer = 0 The priority of the currency.

References