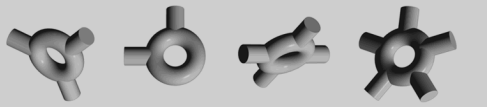
elements



Payment Rates Package

TARMS Inc.

September 07, 2000

Typeset in LaTeX.

Contents

1 Use Cases **2**
- 1.1 Loan (Actual Payments) . . . 2
- 1.2 Loan (Level Payments) . . . 3
- 1.3 Bond Coupons . . . 3
- 1.4 Discounted FRA Payments . . . 3
- 1.5 Accrued Interest . . . 4

2 Interfaces **4**
- 2.1 PaymentRate . . . 4
  - 2.1.1 Relationships . . . 4
  - 2.1.2 Operations . . . 4

3 Classes **9**
- 3.1 PaymentRateModel . . . 9
  - 3.1.1 Relationships . . . 9
  - 3.1.2 Attributes . . . 9
  - 3.1.3 Operations . . . 9
- 3.2 PaymentRateActualModel . . . 11
  - 3.2.1 Relationships . . . 11
  - 3.2.2 Operations . . . 11
- 3.3 PaymentRateLevelModel . . . 13
  - 3.3.1 Relationships . . . 13
  - 3.3.2 Operations . . . 13

4 Associations **15**
- 4.1 rate . . . 16

5 Extensions to the Interest Rates Package **19**
- 5.1 PointInterestRate . . . 19
  - 5.1.1 Operations . . . 19

List of Figures

1 Class Diagram— Examples . . . 17
2 Class Diagram— Payment Rates . . . 18

List of Tables

1 Payment Rates— Associations 15
1 . . . continued 16

Package Description

Payment rates represent amounts of money that need to be paid under contracts such as bonds or loans.

The interest rates package is largely concerned with interest rates representing the time value of money — discount factors. This package is directed towards the use of interest rates as payment rates; prices indicating the interest payments on a loan or the coupons on a bond.

Payment rates may be based on the actual time elapsed between payment dates, or it may be based on a *level payment* convention. Level payments usually quote an annual rate paid at some regular interval; irrespective of the actual period, the amount paid is constant, based on the payment frequency.

1 Use Cases

1.1 Loan (Actual Payments)

A loan has a principal of 1,000,000 USD and a 6% per annum interest rate, with the principal being returned at the end of the loan. Payments are made on a monthly basis, using an Actual/365 date basis. The loan starts on 15-May-1999 and the first payment is due on the following month. The payments for the first year are:

Date	Days	Term	Payment (USD)
15-Jun-1999	31	0.0849315	5095.89
15-Jul-1999	30	0.0821916	4931.51
15-Aug-1999	31	0.0849315	5095.89
15-Sep-1999	31	0.0849315	5095.89
15-Oct-1999	30	0.0821916	4931.51
15-Nov-1999	31	0.0849315	5095.89
15-Dec-1999	30	0.0821916	4931.51
15-Jan-2000	31	0.0849315	5095.89
15-Feb-2000	31	0.0849315	5095.89
15-Mar-2000	29	0.0794521	4767.12
15-Apr-2000	31	0.0849315	5095.89
15-May-2000	30	0.0821916	4931.51

1.2 Loan (Level Payments)

A loan has a principal of 1,000,000 USD and a 6% per annum interest rate, with the principal being returned at the end of the loan. Payments are made on a monthly basis, with all payments being level. The payments are calculated as $1000000 \times \frac{6\%}{12} = 5000$.

1.3 Bond Coupons

A bond with a face value of 1000000 USD pays a coupon of 10% semi-annually. Each coupon is calculated as a level payment of $1000000 \times \frac{10\%}{2} = 50000$.

1.4 Discounted FRA Payments

Australian dollar bank bill FRAs are paid in advance.[3]. Given a notional principal P, a period of m days, a fixed interest rate for m days of r_f and a spot floating interest rate of r_s, then the net settlement payment is given as:

$$\frac{P}{1 + \frac{m}{365} \times r_f} - \frac{P}{1 + \frac{m}{365} \times r_s}$$

1.5 Accrued Interest

Bonds which are bought part-way through a coupon pay *accrued interest*.[4, 2] The accrued amount is calculated from the start of the coupon date to the date of exchange. This accrued amount is paid to the seller of the bond to compensate for the lost interest from the coupon.

In certain cases, Spanish and Italian bonds, the accrual includes the trade date.[1]

Accrual amounts can be calculated using a different date basis to the date basis used for coupon calculations. In this case, there is a potential mismatch between the amount calculated by accrual and the amount of the coupon. Any adjustments are assumed to occur on the coupon date — a date on which accrual calculations are not made.

2 Interfaces

2.1 PaymentRate

A payment rate is an interest rate used to indicate the amount of interest to be paid on a loan, bond, etc.

Payment rates are not as completely specified as interest rates — they have no currency, from-date, to-date or other such information. Instead, payment rates reflect a minimal interest rate specification; enough to generate an interest rate from a supplied specifier.

2.1.1 Relationships

	Class	Description	Notes
↓	PaymentRateModel §3.1		

↓:Realized by

2.1.2 Operations

Boolean isLevel() isLevel

Does this payment use level payments?

Return true if this payment rate specifies level payments, false otherwise.

Integer paymentsPerAnnum() paymentsPer-Annum

The number of payments per annum that are made. Return the number of payments/coupons per year.

DateBasis dateBasis() dateBasis

The date basis used to calculate interest payments.

DateBasis accrualDateBasis() accrualDateBasis

The date basis used to calculate interest accrual.

Cases which include the final day in accrual calculations can be modeled by choosing an appropriate date basis.

SimpleCashflow interestOnPrincipal(CommodityHolding principal, Date fromDate, Date toDate, Date paymentDate, Boolean round, Boolean discount) interestOnPrincipal

principal: CommodityHolding The principal amount.
fromDate: Date The start date of the payment.
toDate: Date The end date of the payment.
paymentDate: Date The date on which the payment is made. The default value is if(discount, fromDate, toDate).
round: Boolean Round amounts to payable amounts? The default value is false.
discount: Boolean Discount the payment to the start of the period? The default value is false.

The interest on a principal amount for *one payment*.

The result returned is a simple cashflow giving the interest amount on the principal. The value returned should equal the sum of the values of the dictionary returned by the interestPiecesOnPrincipal operation.

Dictionary<String, SimpleCashflow> interestPiecesOnPrincipal(CommodityHolding principal, Date fromDate, Date toDate, Date paymentDate, Boolean round, Boolean discount) interestPiecesOnPrincipal

principal: CommodityHolding The principal amount.
fromDate: Date The start date of the payment.
toDate: Date The end date of the payment.
paymentDate: Date The date on which the payment is made. The default value is if(discount, fromDate, toDate).
round: Boolean Round amounts to payable amounts? The default value is false.
discount: Boolean Discount the payment to the start of the period? The default value is false.

The interest on a principal amount, split into the pieces that make up the pay-

ment rate for *one payment*. The pieces are the rate pieces that make up the associated rate. Return the interest payable on a principal amount, broken into a dictionary of piece names mapping onto amounts.

SimpleCashflow accruedInterestOnPrincipal(CommodityHolding principal, Date fromDate, Date toDate, Date startDate, Date endDate, Date paymentDate, Boolean round, Boolean discount)

accruedInterestOnPrincipal

principal: CommodityHolding The principal amount.
fromDate: Date The start date of the payment.
toDate: Date The end date of the payment.
startDate: Date The start date of the accrual.
endDate: Date The end date of the accrual.
paymentDate: Date The date on which the accrual is made. The default value is endDate.
round: Boolean Round amounts to payable amounts? The default value is false.
discount: Boolean Discount the payment to the start of the period? The default value is false.

The accrued interest on a principal amount within *one payment*.

The result returned is a simple cashflow giving the accrued interest on the principal between two dates. The value returned should equal the sum of the values of the dictionary returned by the accruedInterestPiecesOnPrincipal operation.

Dictionary<String, SimpleCashflow> accruedInterestPiecesOnPrincipal(CommodityHolding principal, Date fromDate, Date toDate, Date startDate, Date endDate, Date paymentDate, Boolean round, Boolean discount)

accruedInterestPiecesOnPrincipal

principal: CommodityHolding The principal amount.
fromDate: Date The start date of the payment.
toDate: Date The end date of the payment.
startDate: Date The start date of the interest accrual.
endDate: Date The end date of the interest accrual.
paymentDate: Date The date on which the accrual payment is made. The default value is endDate.
round: Boolean Round amounts to payable amounts? The default value is false.
discount: Boolean Discount the payment to the start of the period? The default value is false.

The accrued interest on a principal amount, split into the pieces that make up the payment rate. The pieces are the rate pieces that make up the associated rate. Return the accrued interest payable on a principal amount, broken into a dictionary of piece names mapping onto amounts.

The specifications for accrual assume that the end-date never reaches the to-date of the payment, where any adjustment in the discrepancy between the accrual calculation and the interest payment are assumed to occur.

SimpleCashflow discountedPrincipal(CommodityHolding principal, Date fromDate, Date toDate, Date paymentDate, Boolean round)

discountedPrincipal

principal: CommodityHolding The principal amount.
fromDate: Date The start date of the payment.
toDate: Date The end date of the payment.
paymentDate: Date The date on which the payment is made. The default value is fromDate.
round: Boolean Round amounts to payable amounts? The default value is false.

The discounted principal amount for *one payment*.

The result returned is a simple cashflow giving the discounted principal for the period covering the payment. The value returned should equal the sum of the values of the dictionary returned by the discountedPrincipalPieces operation.

Dictionary<String, SimpleCashflow> discountedPrincipalPieces(CommodityHolding principal, Date fromDate, Date toDate, Date paymentDate, Boolean round)

discountedPrincipalPieces

principal: CommodityHolding The principal amount.
fromDate: Date The start date of the payment.
toDate: Date The end date of the payment.
paymentDate: Date The date on which the payment is made. The default value is fromDate.
round: Boolean Round amounts to payable amounts? The default value is false.

The discounted principal amount for *one payment*. The pieces are the rate pieces that make up the associated rate.

Return the discounted principal amount, broken into a dictionary of piece names mapping onto amounts.

BasicInterestRate asInterestRate(Currency currency, Date fromDate, Date toDate) asInterestRate

currency: Currency The currency for the interest rate.
fromDate: Date The from-date of the interest rate.
toDate: Date The end-date of the interest rate.

Convert this rate into a BasicInterestRate with the following specifier:

Name	Value
commodity	currency
location	nil
party	nil
date basis	dateBasis()
from-date	fromDate
to-date	toDate
from-period	nil
to-period	nil

The bid and ask components of the interest rates are set to be renamed copies of the associated rate component.

BasicInterestRate asInterestRate(Currency currency, Period fromPeriod, Period toPeriod) asInterestRate

currency: Currency The currency for the interest rate.
fromPeriod: Period The start period for this rate.
toPeriod: Period The end period for this rate.

Convert this rate into a BasicInterestRate with the following specifier:

Name	Value
commodity	currency
location	nil
party	nil
date basis	dateBasis()
from-date	nil
to-date	nil
from-period	fromPeriod
to-period	toPeriod

The bid and ask components of the interest rates are set to be renamed copies of the associated rate component.

3 Classes

3.1 PaymentRateModel

A concrete implementation of the PaymentRate interface. This model holds a single InterestRateQuote which is used to express the rate.

3.1.1 Relationships

	Class	Description	Notes
↑	PaymentRate §2.1		
⇓	PaymentRateLevelModel §3.3		
⇓	PaymentRateActualModel §3.2		
↔	InterestRateQuote	rate 1..1	→

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

3.1.2 Attributes

dateBasis: DateBasis

accrualDateBasis: DateBasis The date basis under which accrued interest is calculated. This may be a nil value, indicating that the payment date basis is used instead.

paymentsPerAnnum: Integer The number of payments that are made per annum for this rate.

3.1.3 Operations

DateBasis accrualDateBasis() accrualDateBasis

The date basis used to calculate interest accrual. If the accrualDateBasis attribute is non-nil then return that attribute, otherwise return the result of the dateBasis() operation.

Dictionary<String, SimpleCashflow> accruedInterestPiecesOnPrincipal(CommodityHolding principal, Date fromDate, Date toDate, Date startDate, Date endDate, Date paymentDate, Boolean round, Boolean discount) accruedInterestPiecesOnPrincipal

principal: CommodityHolding The principal amount.
fromDate: Date The start date of the payment.
toDate: Date The end date of the payment.
startDate: Date The start date of the interest accrual. The default value is fromDate.

endDate: Date The end date of the interest accrual.
paymentDate: Date The date on which the accrual payment is made. The default value is endDate.
round: Boolean Round amounts to payable amounts? The default value is false.
discount: Boolean Discount the payment to the start of the period? The default value is false.

The accrued interest on a principal amount, split into the pieces that make up the payment rate.

If discount is true, then let D be the discount factor between the start and end dates, using the accrual date basis and the associated rate. If discount is false, $D = 1$.

Let $\{(n_i, r_i)\}$ be the names and interest rates supplied by the associated price, with each r_i converted into an annualized rate with the accrual date basis of this payment rate. Assume that the base piece is (n_1, r_1). Let p be the supplied principal. Let $t_i = \sum_{j=1}^{i} r_j$. Let $\lceil x \rceil$ represent the rounding convention of the commodity of the principal. Let y be the term in years between the start-date and end-date, according to the supplied date basis.

If round is true, return a dictionary of

$$\{(n_i, \lceil Dpt_i y \rceil - \lceil Dpt_{i-1} y \rceil)\}$$

If round is false, return a dictionary of

$$\{(n_i, Dp(t_i - t_{i-1})y)\}$$

As an example, suppose we have a rate of 5.6%(base) + 100bp(risk) + 120bp(profit).

Accrual date basis	30/360
Principal	USD 1000000
fromDate	20-Feb-2001
toDate	20-Aug-2001
startDate	20-Feb-2001
endDate	23-Mar-2001
paymentDate	23-Mar-2001
round	true
discount	false

The parameters we have are $D = 1$, $y = 33/360$ and $t_1 = 0.056$, $t_2 = 0.066$ and $t_3 = 0.078$. We then have $rate = \lceil 1 \times 1000000 \times 33/360 \times 0.056 \rceil = 5133.33$, $risk = \lceil 1 \times 1000000 \times 33/360 \times 0.066 \rceil - 5133.33 = 916.67$ and $risk = \lceil 1 \times 1000000 \times 33/360 \times 0.078 \rceil - 6050.00 = 1100.00$.

3.2 PaymentRateActualModel

A payment rate that uses calculated day counts for calculating interest payments, as opposed to assuming a set of level payments.

3.2.1 Relationships

	Class	Description	Notes
⇑	PaymentRateModel §3.1		

⇑:Inherits

3.2.2 Operations

Boolean isLevel() isLevel

Does this payment use level payments?
Return false.

Dictionary<String, SimpleCashflow> interestPiecesOnPrincipal(SimpleCashflow principal, Date fromDate, Date toDate, Date paymentDate, Boolean round, Boolean discount) interestPiecesOnPrincipal

principal: SimpleCashflow The principal amount.
fromDate: Date The start date of the payment.
toDate: Date The end date of the payment.
paymentDate: Date The date on which the payment is made. The default value is toDate.
round: Boolean Round amounts to payable amounts? The default value is false.
discount: Boolean Discount the payment to the start of the period? The default value is false.

The interest on a principal amount, split into the pieces that make up the payment rate for *one payment*.

If discount is true, let D be the discount factor derived from the associated rate, using the supplied from-date and to-date and using the dateBasis attribute as a date basis. If discount is false, $D = 1$.

Let $\{(n_i, r_i)\}$ be the names and interest rates supplied by the associated price, with each r_i converted into an annualized rate with the date basis of this payment rate. Assume that the base piece is (n_1, r_1). Let p be the supplied principal. Let $t_i = \sum_{j=1}^{i} r_j$. Let $\lceil x \rceil$ represent the rounding convention of the commodity of the principal. Let y be the term in years between the from-date and to-date, according to the dateBasis attribute.

If round is true, return a dictionary of

$$\{(n_i, \lceil Dpt_i y \rceil - \lceil Dpt_{i-1} y \rceil)\}$$

If round is false, return a dictionary of

$$\{(n_i, Dp(t_i - t_{i-1})y)\}$$

As an example, suppose we have a rate of 5.6%(base) + 100bp(risk) + 120bp(profit).

Accrual date basis	30/Actual
Principal	USD 1000000
fromDate	20-Feb-2001
toDate	20-Aug-2001
paymentDate	20-Aug-2001
round	true
discount	true

The parameters we have are $D = 1/(1+0.078 \times \frac{180}{365}) = 0.9630$, $y = 180/365$ and $t_1 = 0.056$, $t_2 = 0.066$ and $t_3 = 0.078$. We then have $rate = \lceil 0.9630 \times 1000000 \times 180/365 \times 0.056 \rceil = 26593.45$, $risk = \lceil 0.9630 \times 1000000 \times 180/365 \times 0.066 \rceil - 26593.45 = 4748.89$ and $risk = \lceil 0.9630 \times 1000000 \times 180/365 \times 0.078 \rceil - 31342.34 = 5698.61$.

Dictionary<String, SimpleCashflow> discountedPrincipalPieces(CommodityHolding principal, Date fromDate, Date toDate, Date paymentDate, Boolean round)

discountedPrincipalPieces

principal: CommodityHolding The principal amount.
fromDate: Date The start date of the payment.
toDate: Date The end date of the payment.
paymentDate: Date The date on which the payment is made. The default value is fromDate.
round: Boolean Round amounts to payable amounts? The default value is false.

The discounted principal amount for *one payment*.

Let $\{(n_i, r_i)\}$ be the names and interest rates supplied by the associated price, with each r_i converted into an annualized rate with the date basis of this payment rate. Assume that the base piece is (n_1, r_1). Let p be the supplied principal and m be the paymentsPerAnnum attribute. Let $t_i = \sum_{j=1}^{i} r_j$. Let $\lceil x \rceil$ represent the rounding convention of the commodity of the principal. Let y be the term in years between the from-date and to-date, according to the dateBasis attribute.

If round is true, return a dictionary of

$$\{(n_i, \lceil \frac{p}{1+yt_i} \rceil - \lceil \frac{p}{1+yt_{i-1}} \rceil)\}$$

If round is false, return a dictionary of

$$\{(n_i, \frac{p}{1+yt_i} - \frac{p}{1+yt_{i-1}})\}$$

As an example, suppose we have a rate of 5.6%(base) + 100bp(risk) + 120bp(profit).

Date basis	30/Actual
Principal	USD 1000000
fromDate	20-Feb-2001
toDate	20-Aug-2001
paymentDate	20-Aug-2001
round	true

The parameters we have are $y = 180/365$ and $t_1 = 0.056$, $t_2 = 0.066$ and $t_3 = 0.078$. We then have $rate = \lceil 1000000/(1 + 180/365 \times 0.056) \rceil = 973125.73$, $risk = \lceil 1000000/(1 + 180/365 \times 0.066) \rceil - 973125.73 = -4647.70$ and $risk = \lceil 1000000/(1 + 180/365 \times 0.078) \rceil - 968478.03 = -5518.98$.

3.3 PaymentRateLevelModel

A payment rate where the payments are made in level amounts.

3.3.1 Relationships

	Class	Description	Notes
⇑	PaymentRateModel §3.1		

⇑:Inherits

3.3.2 Operations

Boolean isLevel() isLevel

Does this payment use level payments?
Return true.

Dictionary<String, SimpleCashflow> interestPiecesOnPrincipal(SimpleCashflow principal, Date fromDate, Date toDate, Date paymentDate, Boolean round, Boolean discount) interestPiecesOnPrincipal

principal: SimpleCashflow The principal amount.
fromDate: Date The start date of the payment.
toDate: Date The end date of the payment.
paymentDate: Date The date on which the payment is made. The default value is toDate.

round: Boolean Round amounts to payable amounts? The default value is false.
discount: Boolean Discount the payment to the start of the period? The default value is false.

The interest on a principal amount, split into the pieces that make up the payment rate for *one payment*.

If discount is true, let D be the discount factor derived from the associated rate, using the supplied from-date and to-date and using the dateBasis attribute as a date basis. If discount is false, $D = 1$.

Let $\{(n_i, r_i)\}$ be the names and interest rates supplied by the associated price, with each r_i converted into an annualized rate with the date basis of this payment rate. Assume that the base piece is (n_1, r_1). Let p be the supplied principal and m be the paymentsPerAnnum attribute. Let $t_i = \sum_{j=1}^{i} r_j$. Let $\lceil x \rceil$ represent the rounding convention of the commodity of the principal.

If round is true, return a dictionary of

$$\{(n_i, \lceil \frac{Dpt_i}{m} \rceil - \lceil \frac{Dpt_{i-1}}{m} \rceil)\}$$

If round is false, return a dictionary of

$$\{(n_i, \frac{Dp(t_i - t_{i-1})}{m})\}$$

As an example, suppose we have a rate of 5.6%(base) + 100bp(risk) + 120bp(profit).

paymentsPerAnnum	2
Principal	USD 1000000
fromDate	20-Feb-2001
toDate	20-Aug-2001
paymentDate	20-Aug-2001
round	true
discount	false

The parameters we have are $D = 1$, $m = 2$ and $t_1 = 0.056$, $t_2 = 0.066$ and $t_3 = 0.078$. We then have $rate = \lceil 1 \times 1000000 \times 0.056/2 \rceil = 28000$, $risk = \lceil 1 \times 1000000 \times 0.066/2 \rceil - 28000 = 5000$ and $risk = \lceil 1 \times 1000000 \times 0.078/2 \rceil - 33000 = 6000$.

Dictionary<String, SimpleCashflow> discountedPrincipalPieces(CommodityHolding principal, Date fromDate, Date toDate, Date paymentDate, Boolean round)

discountedPrincipalPieces

principal: CommodityHolding The principal amount.
fromDate: Date The start date of the payment.

toDate: Date The end date of the payment.
paymentDate: Date The date on which the payment is made. The default value is fromDate.
round: Boolean Round amounts to payable amounts? The default value is false.

The discounted principal amount for *one payment*.

Let $\{(n_i, r_i)\}$ be the names and interest rates supplied by the associated price, with each r_i converted into an annualized rate with the date basis of this payment rate. Assume that the base piece is (n_1, r_1). Let p be the supplied principal and m be the paymentsPerAnnum attribute. Let $t_i = \sum_{j=1}^{i} r_j$. Let $\lceil x \rceil$ represent the rounding convention of the commodity of the principal.

If round is true, return a dictionary of

$$\{(n_i, \lceil \frac{p}{1 + \frac{t_i}{m}} \rceil - \lceil \frac{p}{1 + \frac{t_{i-1}}{m}} \rceil)\}$$

If round is false, return a dictionary of

$$\{(n_i, \frac{p}{1 + \frac{t_i}{m}} - \frac{p}{1 + \frac{t_{i-1}}{m}})\}$$

As an example, suppose we have a rate of 5.6%(base) + 100bp(risk) + 120bp(profit).

paymentsPerAnnum	2
Principal	USD 1000000
fromDate	20-Feb-2001
toDate	20-Aug-2001
paymentDate	20-Aug-2001
round	true

The parameters we have are $m = 2$ and $t_1 = 0.056$, $t_2 = 0.066$ and $t_3 = 0.078$. We then have $rate = \lceil 1000000/(1 + 0.056/2) \rceil = 972762.65$, $risk = \lceil 1000000/(1 + 0.066/2) \rceil - 972762.65 = -8708.44$ and $risk = \lceil 1000000/(1 + 0.078/2) \rceil - 968054.21 = -5590.30$.

4 Associations

Table 1: Payment Rates— Associations

Association Role	Class	Card.	Notes
rate			

Table 1: . . . continued

Association Role	Class	Card.	Notes
rate	InterestRateQuote	1..1	→
payment rate	PaymentRateModel §3.1	0..n	

→:Navigable ◊:Aggregate ♦:Composite

4.1 rate

Role: rate *Navigable* InterestRateQuote, 1..1.
Role: payment rate PaymentRateModel, 0..n.
The interest rate quote that makes up the payment rate.

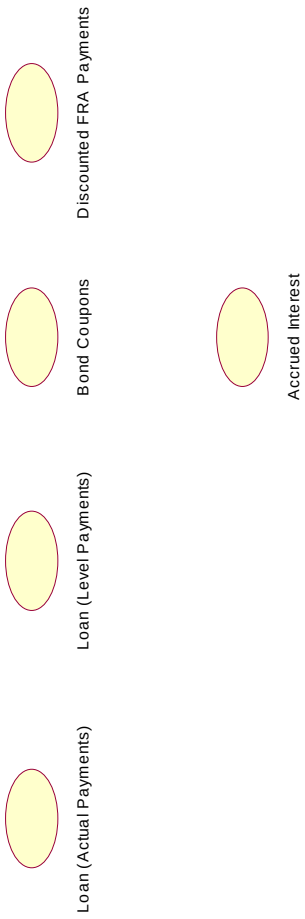


Figure 1: Class Diagram— Examples

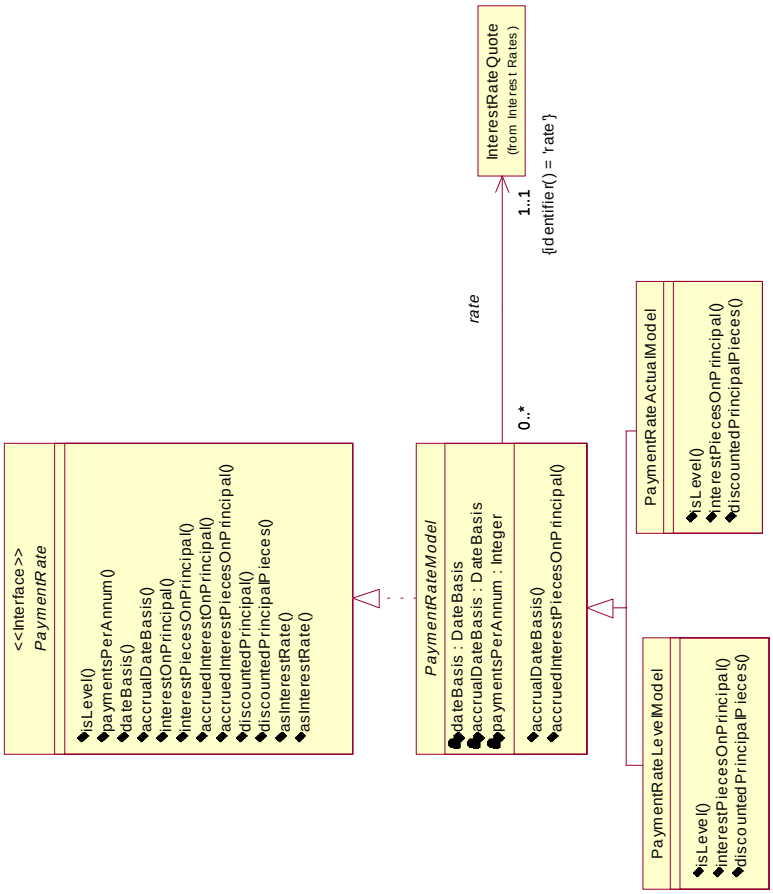


Figure 2: Class Diagram— Payment Rates

5 Extensions to the Interest Rates Package

5.1 PointInterestRate

5.1.1 Operations

PaymentRate asPaymentRate(DateBasis accrualDateBasis, Boolean isLevel, Boolean payments) asPaymentRate

accrualDateBasis: DateBasis The default value is nil.
isLevel: Boolean The default value is true.
payments: Boolean The number of payments per annum (if nil then derived from the quotation frequency of the interest rate). The default value is nil.

Return a payment rate with the the following characteristics:

rate The mid rate of this interest rate.

dateBasis The date basis specifier of this rate. If not specified, assume the date basis of the currency of the interest rate.

accrualDateBasis The accrualDateBasis argument. This argument may be nil, indicating an accrual date basis which is the same as the payment date basis.

paymentsPerAnnum If paymentsPerAnnum argument is non-nil, then use the argument. Otherwise, if the mid rate of this interest rate is quoted using a DiscreteLevelCompoundingFequency then use the frequency of that model. Otherwise, assume a value of 1.

References

[1] Patrick J. Brown. *Bond Markets: Structures and Yield Calculations*. Amacom, 1998.

[2] International Securities Market Association (ISMA), Zurich. *Statutes, Bye Laws, Rules and Recommendations*, 1997.

[3] Michael Sherris. *Money and Captial Markets*. Allen and Unwin, 1991.

[4] Robert Steiner. *Mastering Financial Calculations*. Pitman Publishing, 1998.