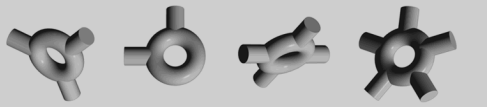
elements



Securities Package

TARMS Inc.

September 07, 2000

Typeset in LaTeX.

Contents

List of Figures

List of Tables

Package Description

Governments, corporations, banks or other companies looking to raise funds may issue securities to do so. These can be debt securities of a short-term (eg. bills) or long-term nature (eg. bonds), or equity securities (eg. shares).

This package defines these issued securities. As their attributes tend to change infrequently, they are a part of reference data.

The securities currently modeled are fixed interest bond issues and equities. Fixed interest bonds pay a fixed rate of interest and mature at a fixed point in time. The interest or coupon is paid at regular intervals. The financial component of a bond issue can be defined by the InstrumentWithSpecifier.

1 Interfaces

1.1 DebtSecurityType

When securities, in particular bonds, are issued there are certain features of these securities which can denote the 'type' of security that they are.

1.1.1 Relationships

	Class	Description	Notes
↓	DebtSecurityTypeModel §2.2		
↔	DebtSecurityModel §2.5	type	

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

1.1.2 Operations

String type() type

Returns the type of security, for example for Bonds: EuroBond, Flirbs, YankieBonds and for Bills: Certificate of Deposit, Bank accepted, etc

1.2 Security

This is an abstract interface designed to collate the common behavior of securities.

1.2.1 Relationships

	Class	Description	Notes
⇑	Instrument		
⇑	Identifiable		
⇓	DebtSecurity §1.3		
⇓	Equity §1.5		
↓	SecurityModel §2.4		

⇑:Inherits ⇓:Inherited by ↓:Realized by

1.2.2 Operations

Organization issuer() issuer

Returns the writer of the security being issued.

String identifier() identifier

Returns the unique name of the security. This is usually an abbreviation of the full name of the security issue.

String description() description

Returns the unabbreviated name of the security.

InstrumentWithSpecifier instrument() instrument

Returns the financial information relating to the security being created. For example, the cashflow(s) associated with a bill or a bond.

Location country() country

Returns the country of the security. This is used for credit risk purposes.

Currency currency() currency

Returns the currency in which the security is denominated. This can be different from the country of the issuer.

Location legalCountry() legalCountry

Returns the country under whose laws the security will be interpreted. This can be different from the issuer's country.

Collection listings() listings

A security can be listed on more than one exchange. This operation returns the collection of exchanges on which a security is listed. This may be an empty collection.

Number lotSize() lotSize

Returns the size unit in which the security can be traded. For example, this identifies the size of the parcel of bonds to be traded, usually 1000. The face value of bonds $= n \times lotSize$ where n is an integer.

Number minimumLot() minimumLot

Returns the minimum amount of securities that can be traded. If m is the minimum lot, l is the lot size then the trade amounts must be $m > n \times l$ where n is an integer. For example, the lot size may be set at 100, so the securities can only be traded in bundles of 100, but the minimumLot may be set at 500.

Date valueDate() valueDate

This is the addition of the value period to the start date. valueDate $=$ valuePeriod$+$

startDate. Return the date. The start date can be found in the InstrumentWithSpecifier.

Integer paymentFrequency() paymentFrequency

Returns the frequency of coupon or dividend payments per annum.

Period valuePeriod() valuePeriod

Returns the period from the start date on which settlement is supposed to occur. There are situations where the settlement date has been agreed upon, but is rescheduled. Accrued interest calculations start from the originally agreed date, the value date, rather than the rescheduled settlement date.

Boolean canLend() canLend

Returns whether the security can be used as collateral.

1.3 DebtSecurity

DebtSecurity is an abstract interface where the general information relevant to both bills and bonds can be found. For example, bill and bond contracts have some attributes in common. These can be found on the Security §1.2 interface.

1.3.1 Relationships

	Class	Description	Notes
⇑	Security §1.2		
⇓	Bond §1.4		
↓	DebtSecurityModel §2.5		
↓	SecurityModel §2.4		
↓	DebtSecurityReferenceData-Model §2.1		
↔	DebtSecurityReferenceData-Model §2.1	model	

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

1.3.2 Operations

SecurityType type() type

Returns the type of security. This allows for the easy identification of particular types of securities. For example, Brady bonds can be identified by a type = 'Brady'.

OrderedCollection guarantor() guarantor

Returns the organization that will pay for the bond, in the case of an issuer defaulting on payment. There can be more than one guarantor associated with a security. They are order dependent.

String ticker() ticker

Returns a string that is used in conjunction with the series to identify the security.

String series() series

Returns a string that is used in conjunction with the ticker to identify the security.

DateBasis dateBasis() dateBasis

Returns the day and year count that applies to the security.

1.4 Bond

A bond specifies the characteristics of a particular bond primary issue. These bonds can then be traded on secondary markets. In this case a deal will be done for a specified amount of a given bond. For example: A 10 year USD Treasury bond @ 6.625%. The deal will be for $1 million face value of the 10 year USD Treasury bond @ 6.625%.

1.4.1 Relationships

	Class	Description	Notes
⇑	DebtSecurity §1.3		
↓	BondModel §2.6		

⇑:Inherits ↓:Realized by

1.4.2 Operations

CommodityHolding calculateAccruedInterestToDate(Date aDate) calculateAccruedInterestToDate

aDate: Date This is the date for which accrued interest is calculated to, from the last coupon payment date.

Accrued interest is the interest accrued, but not paid, to a date.

1.5 Equity

An equity specifies the characteristics of shares, whether they be a primary or secondary issue. (Note: The term equity can be used synonymously with share.) Deals will be executed for a specified number of shares. For example, a broker may place an order for 500 IBM shares.

One reason shareholders may hold shares is for capital growth. Shareholders may also derive income from dividends associated with their shares. Dividends are the amount of a company's after-tax earnings which are paid to shareholders. A regular dividend is usually paid on a semi-annual basis. A company may also pay an extra dividend to shareholders in times of prosperity. With the declaration of an extra dividend, a company is warning investors that the dividend is not an increase in the established dividend rate. An increase in the dividend rate can be a meaningful signal that management has increased its assessment of the future earnings of a company.

Dividends can provide meaningful information about shares. For these reasons, dividend information is being modeled with shares.

1.5.1 Relationships

	Class	Description	Notes
⇑	Security §1.2		
↓	EquityModel §2.7		
↓	EquityReferenceDataModel §2.3		
↔	EquityReferenceDataModel §2.3	model	

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

1.5.2 Operations

String classOfShare() classOfShare

Returns the class of the issue that specifies the voting rights associated with a share. For example "Class A" or "Class B" shares can have different voting rights.

CommodityHolding issuedCapital() issuedCapital

Returns the amount of capital that has been issued. From the issuedCapital and faceValue the number of shares can be calculated as follows:

$$NumberofShares = \frac{Totalissuedcapital}{FaceValue}$$

This can be used to determine the percentage of the company that you own. This is the total nominal, or face value of the shares of the company that have actually been issued or allocated to shareholders.

CommodityHolding authorizedCapital() authorizedCapital

Returns the total amount of capital that can be offered to the public for subscription. Not all of the capital available may be issued at an initial offering.

CommodityHolding dividend() dividend

Returns the expected amount of the next dividend payment (per share), expressed as an amount of currency.

Double dividendYield() dividendYield

The dividend yield is the return on a share investment. This method will return the *expected* initial dividend yield for a share.
$dividendYield = \frac{dividend}{market\ price} \times 100$
The market price is derived from a feed of some sort.

The dividend is expressed as cents per share. The dividendYield is expressed as a percentage value.

CommodityHolding faceValue() faceValue

Returns the issued value of a share.

CommodityHolding sharePremiumReserve() sharePremiumReserve

When shares are issued and sold to new subscribers, they are often issued at a value above the face value of the share. The share premium reserve is the difference between the issue price (the price new subscribers pay for the shares) and the par value of the share. If a company is going to do a rights issue or reserves are going to be recapitalized, this would be used as a check against the new nominal value of the shares rather than the nominal value being input manually.

Collection<Transaction> dividendCalendar() dividendCalendar

A dividend calendar returns a collection of dividend payments with the future dividends from data entry showing.

The dividend for a given year is determined by the company. The amount of a dividend in the early years tends to change frequently. In later years, a regular dividend tends to be paid. Dividends are paid on a discretionary basis. They are not known in advance. The dividend streams will be used in giving estimates for analytics and for the calculation of risk premiums.

Double dividendGrowthRate() dividend-GrowthRate

Returns the rate at which the dividend is expected to grow.

2 Classes

2.1 DebtSecurityReferenceDataModel

This class allows debt securities to be set up as reference data by implementing a reference data wrapper.

2.1.1 Relationships

	Class	Description	Notes
⇑	ReferenceDataModel		
↑	DebtSecurity §1.3		
↔	DebtSecurity §1.3	model	→

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

2.2 DebtSecurityTypeModel

The DebtSecurityTypeModel realizes the DebtSecurityType interface.

2.2.1 Relationships

	Class	Description	Notes
↑	DebtSecurityType §1.1		

↑:Realizes

2.2.2 Attributes

type: String This identifies the type of security.

2.3 EquityReferenceDataModel

This class allows Equities to be set up as reference data by implementing a reference data wrapper.

2.3.1 Relationships

	Class	Description	Notes
⇑	ReferenceDataModel		
↑	Equity §1.5		
↔	Equity §1.5	model	→

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

2.4 SecurityModel

This is an abstract class containing the common characteristics of different types of security models. This class realizes the Security interface.

2.4.1 Relationships

	Class	Description	Notes
↑	DebtSecurity §1.3		
↑	Security §1.2		
⇓	DebtSecurityModel §2.5		
⇓	EquityModel §2.7		
↔	Location	legalCountry	→
↔	Period	valuePeriod	→
↔	Organization	listings	→
↔	InstrumentWithSpecifier	instrument	→

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

2.4.2 Attributes

identifier: String The unique name of the security. This is usually an abbreviation of the full name of the bond issue.

description: String The long (unabbreviated) name of the security.

lotSize: Number The smallest granularity in which the security can be traded. That is, all trades must be in multiples of the lot size. For example, this identifies the size of the parcel of bonds to be traded, usually 1000.

minimumLot: Number The minimum amount of securities that can be traded. If m is the minimum lot, l is the lot size then the trade amounts must be $m > n \times l$ where n is an integer.

canLend: Boolean Specifies if the security can be used as collateral.

2.4.3 Operations

Organization issuer() issuer

Returns the counterparty found on the InstrumentWithSpecifier.

Location country() country

Returns the country of the issuer. This is found by the location with a type of 'country' being returned.

Currency currency() currency

This value will be determined from within the instrument. The instrument will be of type InstrumentWithSpecifier, and the specifier contains a currency, accessible by the 'currency' method on the specifier. This currency should be returned by this method.

Reportable validate() validate

The validation for this would be subject to business rules, and hence is left incomplete.

Integer paymentFrequency() paymentFrequency

Returns the number of coupon or dividend payments per annum. It is stored on the specifier of the associated instrument.

2.5 DebtSecurityModel

DebtSecurityModel is an abstract class which realizes DebtSecurity. Examples of debt securities are bills or bonds.

The DebtSecurityModel class contains the common features of bills and bonds.

2.5.1 Relationships

	Class	Description	Notes
⇑	SecurityModel §2.4		
↑	DebtSecurity §1.3		
⇓	BondModel §2.6		
↔	Organization	guarantor	→
↔	DebtSecurityType §1.1	type	→

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

2.5.2 Attributes

ticker: String This string is used in conjunction with the series to identify a debt security.

series: String This string is used in conjunction with the ticker to identify the debt security.

2.6 BondModel

The BondModel realizes the Bond interface. This class specifies a bond.

2.6.1 Relationships

	Class	Description	Notes
⇑	DebtSecurityModel §2.5		
↑	Bond §1.4		

⇑:Inherits ↑:Realizes

2.6.2 Operations

validate() validate

The validation for this would be subject to business rules, and hence is left incomplete.

2.7 EquityModel

The EquityModel realizes the Equity interface.

2.7.1 Relationships

	Class	Description	Notes
⇑	SecurityModel §2.4		
↑	Equity §1.5		

⇑:Inherits ↑:Realizes

2.7.2 Attributes

classOfShare: String

dividendYield: Double

issuedCapital: CommodityHolding

authorizedCapital: CommodityHolding

2.7.3 Operations

Double dividendGrowthRate() dividend-GrowthRate

Returns the paymentGrowthRate found on the CashflowSeriesSpecifier found within the instrument.

Collection<Transaction> dividendCalendar() dividendCalendar

Returns the instrument's transaction sequence.

CommodityHolding dividend() dividend

The dividend is converted to a dividend rate by dividing it by the face value of the share. This dividend rate is then stored in the 'fixedInterestRate' of the specifier on the receiver's instrument.

The inverse operation is performed to return the dividend.

3 Associations

Table 1: Securities— Associations

Association Role	Class	Card.	Notes
legalCountry			
	Location		→

Table 1: . . . continued

Association Role	Class	Card.	Notes
	SecurityModel §2.4		
guarantor			
	Organization		→
	DebtSecurityModel §2.5	0..n	
type			
	DebtSecurityType §1.1		→
	DebtSecurityModel §2.5		
valuePeriod			
	Period		→
	SecurityModel §2.4		
listings			
	Organization		→
	SecurityModel §2.4		
instrument			
	InstrumentWithSpecifier		→
	SecurityModel §2.4		
model			
	DebtSecurity §1.3		→
	DebtSecurityReferenceData-Model §2.1		
model			
	Equity §1.5		→
	EquityReferenceDataModel §2.3		

→:Navigable ◊:Aggregate ⧫:Composite

3.1 legalCountry

Role: *Navigable* Location.
Role: SecurityModel.

3.2 guarantor

Role: *Navigable* Organization.
Role: DebtSecurityModel, 0..n.

3.3 type

Role: *Navigable* DebtSecurityType.
Role: DebtSecurityModel.

3.4 valuePeriod

Role: *Navigable* Period.
Role: SecurityModel.

3.5 listings

Role: *Navigable* Organization.
Role: SecurityModel.

3.6 instrument

Role: *Navigable* InstrumentWithSpecifier.
Role: SecurityModel.

3.7 model

Role: *Navigable* DebtSecurity.
Role: DebtSecurityReferenceDataModel.

3.8 model

Role: *Navigable* Equity.
Role: EquityReferenceDataModel.

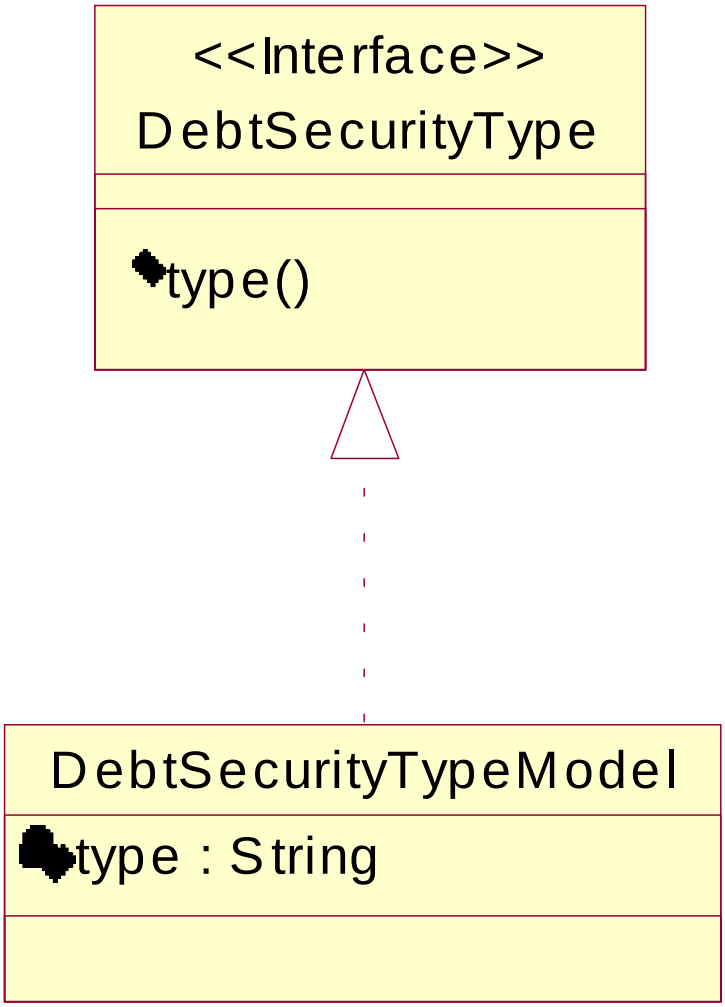


Figure 1: Class Diagram— DebtSecurityType

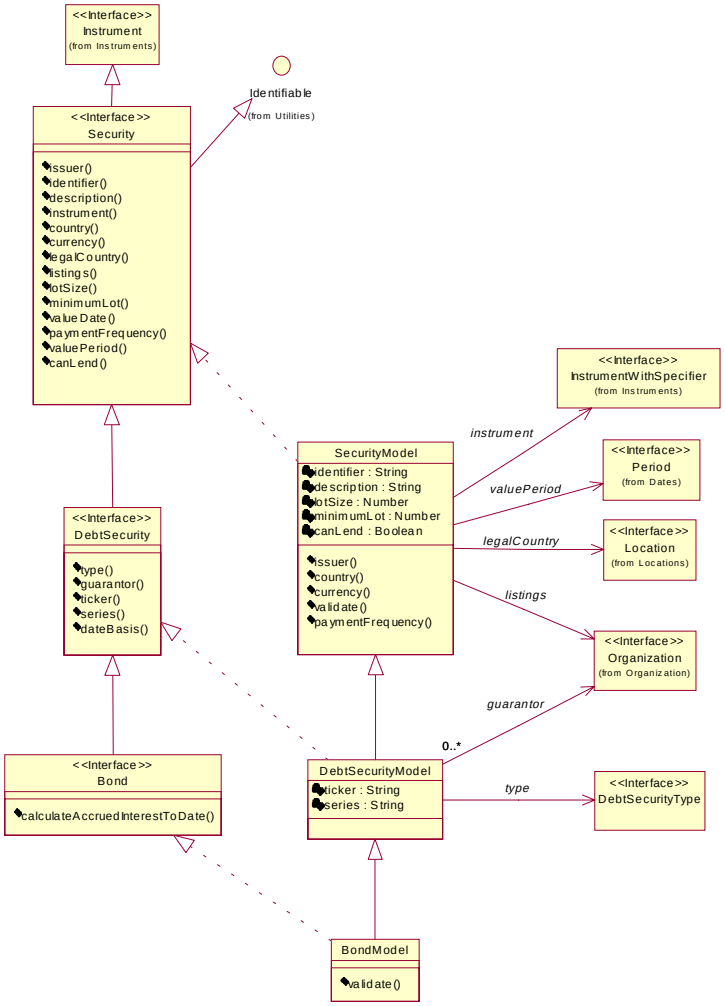


Figure 2: Class Diagram— DebtSecurities

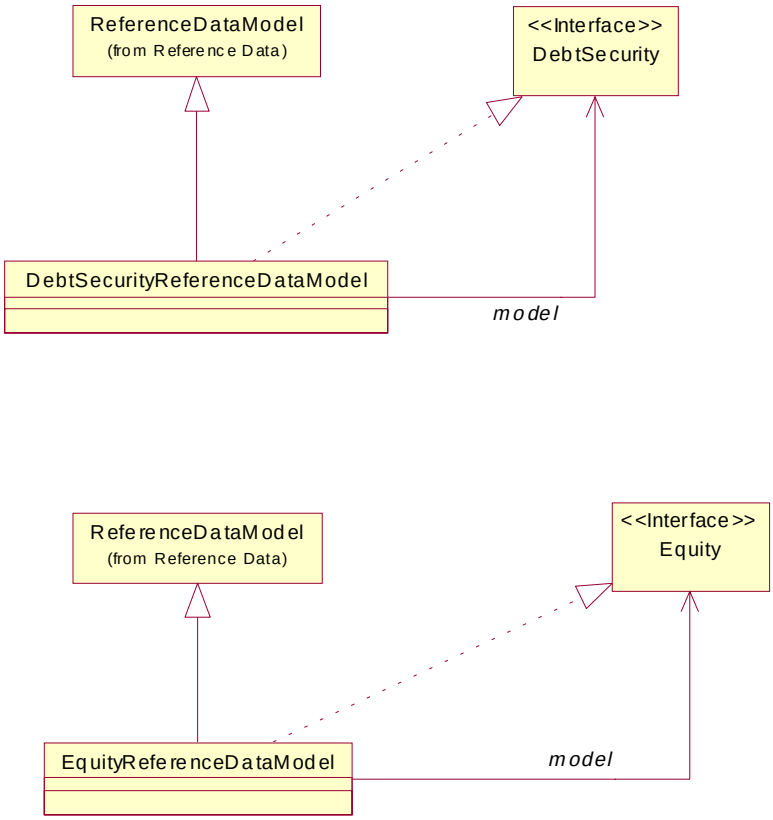


Figure 3: Class Diagram— SecurityReferenceData

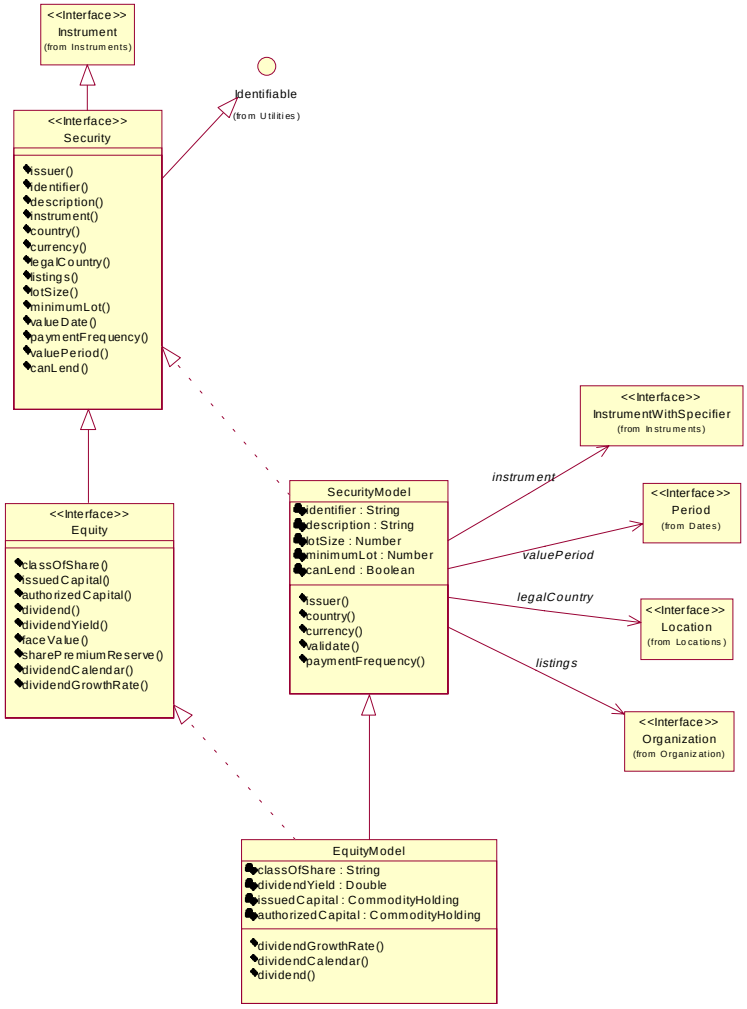


Figure 4: Class Diagram— Equities

References