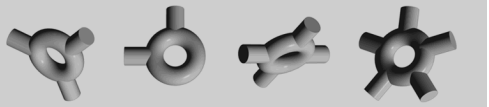
elements



Basic Commodities Package

TARMS Inc.

September 07, 2000

Typeset in LaTeX.

Contents

List of Figures

List of Tables

Package Description

This package deals with abstract commodities and their most visible financial instance, currencies. Commodities represent some tradeable asset that can not be defined in terms of some other, more primitive, instrument. Example commodities are copper, oranges, stocks or amounts of cash denominated in some currency.

Commodities are, generally, exchangeable in some way. In the simplest terms, this represents a barter economy. For example, “I trade 2 oxen for a cat, a copper knife with gold inlay and a statue of Rameses II. Bargain!” One characteristic of commodities is immediately apparent from the above example: the trades are all one-of-a-kind events in the sense that the price of a commodity fluctuates from trade to trade.

Barter trading tends to be complex and extremely inefficient; it is difficult to find someone who wishes to exchange what you want for what you have. Currencies allow the conversion of commodities into a common base, so that exchanges

can occur more easily. Rather than find an exact barter, commodities can be exchanged for some currency, which can then be split, joined or stored. Naturally, prices tend to be expressed in terms of some common currency, rather than in exchange terms. Since currencies are tradeable for other commodities, currencies are themselves commodities. In particular, currencies can be exchanged for other currencies, either through fixed ratios within a single currency system (eg. pounds, shillings and pence or the EMU) or via some sort of ordinary trade (foreign exchange).

Whatever form the currency takes, currencies are usually associated with governments or other organizations. These organizations act to maintain the value of the currency either through guaranteeing weights for commodity currencies, maintaining reserves for fiduciary currencies or enforcing the legality of the currency in the case of fiat currencies. The geographical area of control of a government tends to also reflect the area in which a fiat currency is acceptable, although there are exceptions in the form of generally recognized currencies such as the US dollar or German mark.

1 Use Cases

1.1 United States Dollar

The United States Dollar is the legal tender of the United States of America.

The dollar is broken into 100 cents. A single cent is a legal unit of exchange. 20109 US dollars can be expressed unambiguously as USD 20109.00 or, more commonly as $20,109. 0.25 US dollars is often expressed as 25c.

The value of the USD is maintained by the Federal Reserve Bank of the United States, a branch of the US Federal Government.

1.2 AUD

The Australian Dollar is the legal tender of Australia. The dollar is broken into 100 cents, with units of 5 cents being the smallest legal retail exchange. In general retail use, amounts under 5 cents are rounded to the nearest 5 cent unit.

2 Interfaces

2.1 Commodity

A commodity is an abstract tradeable thing. More detailed sub-interfaces specify the various types of commodity.

2.1.1 Relationships

	Class	Description	Notes
⇓	PrimitiveCommodity §2.2		

⇓:Inherited by

2.1.2 Operations

Number smallestPhysicalAmount() smallestPhysicalAmount

The smallest physical amount of this commodity. Returns the smallest amount of this commodity that can be physically represented. In the case of a physical commodity, such as wheat or wool, this would be in terms of the smallest weight or volume exchangeable. In the case of a currency, this would be the smallest cash denomination available.

Number smallestExchangableAmount() smallestExchangableAmount

The smallest exchangeable amount of this commodity. Returns the smallest amount of this commodity that can be exchanged in an agreement. This amount may be different from the physical amount if trading conventions allow smaller amounts than is physically possible.

Number smallestIntermediateAmount() smallestIntermediateAmount

The smallest amount of this commodity to use as an intermediate value in calculations. If the definition of some calculation requires an intermediate value to be rounded, then use this value as the rounding amount.

CommodityRoundingEnum roundingConvention() roundingConvention

How to round amounts of this commodity. This convention is used in conjunction with the smallest amount values to round off any calculations.

Integer decimals() decimals

The number of decimal places to print an amount of this commodity to. Returns the number of decimal places that this commodity is conventionally printed to.

printAmountFormal(OutputStream stream, Number amount, Language language)

printAmount-Formal

stream: OutputStream The stream to print onto.
amount: Number The amount of the commodity.
language: Language The language to use for numeric formatting conventions. The default value is Language.local().

Print some amount of this commodity in a formal manner. Print the amount of this commodity using formal conventions, designed to eliminate ambiguity. Format the number according to the language's formatting conventions.

printAmountInformal(OutputStream stream, Number amount, Language language)

printAmountIn-formal

stream: OutputStream The stream to print onto.
amount: Number The amount of the commodity.
language: Language The language to use for numeric formatting conventions. The default value is Language.local().

Print some amount of this commodity in an informal manner. Print the amount of this commodity using informal conventions, designed to facilitate reading rather than prevent ambiguity. Format the number according to the language's formatting conventions.

2.2 PrimitiveCommodity

An interface for all primitive commodities. Primitive commodities represent tradeable entities that cannot be further defined in terms of sub-components. Subinterfaces define concrete behavior for the various types of primitive commodity.

Primitive commodities inherit from the ValueSemantics interface, making them suitable attributes.

2.2.1 Relationships

	Class	Description	Notes
⇑	Commodity §2.1		
⇑	ValueSemantics		
⇑	Identifiable		
↑	Validatable		
⇓	Currency §2.3		
↓	PrimitiveCommodityModel §3.2		

⇑:Inherits ⇓:Inherited by ↑:Realizes ↓:Realized by

2.2.2 Operations

StandardizedIdentifier code() code

The standardized code for this commodity. Returns the standards-based identifier that is used to represent this commodity.

String name() name

The full name of the commodity. Returns the full name of the commodity.

CommodityClassificationEnum classification() classification

The classification of the commodity. Returns the classification of the commodity. For example, fiat, fiduciary or commodity.

Organization authority() authority

The controlling authority. Returns the organization that is the authority that issues or controls the trading of this commodity. Returns nil if there is no controlling authority.

Location location() location

The location in which the commodity is traded. Returns the location in which the commodity has value. Returns nil if this commodity has some universal value.

String identifier() identifier

The unique identifier for this commodity. Returns the identifier from the StandardizedIdentifier returned by code().

2.3 Currency

Currencies are liquid commodities that can be easily exchanged for other commodities. Currencies are usually regulated by a national government and have a number of conventions for expressing amounts in the currency.

2.3.1 Relationships

	Class	Description	Notes
⇑	PrimitiveCommodity §2.2		
↓	CurrencyModel §3.3		
↓	CurrencyReferenceDataModel §3.1		
↔	CurrencyReferenceDataModel §3.1	model	

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

2.3.2 Operations

StandardizedIdentifier code() code

The standardized code for this commodity. Currency codes are usually ISO 4217 currency codes.[1]

String symbol() symbol

The currency symbol. Returns the symbol used to normally denote this currency, eg. $ for US dollars or F for French francs.

CurrencySymbolPlacementEnum symbolPlacement() symbolPlacement

Place the currency symbol before or after the amount.

String smallSymbol() smallSymbol

The symbol used for fractional amounts of the currency. Returns the symbol used to denote fractional amounts of the currency. For example, c for cents, p for pence, etc. Returns nil if this currency has no small symbol.

CurrencySymbolPlacementEnum smallSymbolPlacement() smallSymbolPlacement

Place the symbol for small amounts before or after the amount. Returns nil if this currency has no small symbol.

printAmountFormal(OutputStream stream, Number amount, Language language) printAmountFormal

stream: OutputStream The stream to print onto.
amount: Number The amount of the commodity.
language: Language The language to use for numeric formatting conventions. The default value is Language.local().

Print some amount of this commodity in a formal manner. Print the currency code, a space and then the amount, formatted without three-digit groupings and with a number of decimal places given by the result of the decimals() operation.

printAmountInformal(OutputStream stream, Number amount, Language language)

printAmountInformal

stream: OutputStream The stream to print onto.
amount: Number The amount of the commodity.
language: Language The language to use for numeric formatting conventions. The default value is Language.local().

Print some amount of this commodity in an informal manner.

If the amount is less that 1, there is a small currency symbol and decimals() returns a number, d, greater than 0, then multiply the amount by 10^d and round it by the currency's rounding convention, as defined in roundingConvention(). Print the resulting number using the supplied language's number printing conventions and prefixed or postfixed, as specified by smallSymbolPlacement(), by the result of smallSymbol().

If the amount is an integer, then print the integer amount using the supplied language's number printing conventions and prefixed or postfixed by the result of symbol().

Otherwise print the amount, rounded according to the currency's rounding convention and with d decimal places. The amount is prefixed or postfixed by the result of symbol().

3 Classes

3.1 CurrencyReferenceDataModel

This model provides a reference data wrapper for currencies that allow them to be managed as a piece of reference data. This class implements the Currency §2.3 interface by delegating operations to the associated model.

3.1.1 Relationships

	Class	Description	Notes
⇑	ReferenceDataModel		
↑	Currency §2.3		
↔	Currency §2.3	model	→

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

3.1.2 Operations

String defaultDescription() defaultDescription

The default long description of the reference data. Return the name from the associated model.

Reportable validate() validate

This method returns an object of type Reportable which will contain all the errors and warnings generated from this method. Below is a list of the axioms that must hold for an instance of this class to be valid. Each statement is followed by the error or warning message that will be issued if the axiom is violated.

- Returns the validation results for the superclass, composed with the validation results from the associated model.

3.2 PrimitiveCommodityModel

An implementation of the Commodity §2.1 interface that models most of the behavior of the commodity in terms of attributes.

3.2.1 Relationships

	Class	Description	Notes
↑	PrimitiveCommodity §2.2		
⇓	CurrencyModel §3.3		
↔	Organization	authority 0..1	→
↔	Location	location 0..1	→

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

3.2.2 Attributes

code: StandardizedIdentifier The commodity code.

name: String The long name of the commodity.

classification: CommodityClassificationEnum = CommodityClassificationEnum.fiat() The type of commodity.

smallestPhysicalAmount: Number = 0.01 The smallest amount that can be exchanged in this commodity.

smallestExchangableAmount: Number = 0.01 The smallest tradeable amount of this commodity.

smallestIntermediateAmount: Number The amount to round intermediate values to in some calculations. This attribute may be nil.

roundingConvention: CommodityRoundingEnum = CommodityRoundingEnum.round() The rounding convention to use.

decimals: Integer = 2 Number of decimal places to print the commodity to.

3.2.3 Operations

Reportable validate() validate

This method returns an object of type Reportable which will contain all the errors and warnings generated from this method. Below is a list of the axioms that must hold for an instance of this class to be valid. Each statement is followed by the error or warning message that will be issued if the axiom is violated.

- The code is valid and unique.
- The name exists and it is unique.
- There is a valid classification.
- If the commodity is classified as a fiduciary or fiat commodity, there is an authority and a location.
- The smallest physical amount and smallest exchangeable amount exist and are > 0 and the physical amount is greater than or equal to the exchangeable amount.

- The smallest amount implied by the number of decimals is less than or equal to the smallest exchangeable amount. For example, if the smallest exchangeable amount = 0.005 and decimals = 2. The smallest amount implied by the number of decimals is 0.01, whicha is greater than 0.005 and therefore an exception should be generated.
- If the commodity is classified as a commodity type, then add a warning if there is no authority or location.

3.3 CurrencyModel

An implementation of the currency interface, using attributes to store most of the characteristics of the currency.

3.3.1 Relationships

	Class	Description	Notes
⇑	PrimitiveCommodityModel §3.2		
↑	Currency §2.3		
↔	LocationModel	currency 0..n	

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

3.3.2 Attributes

symbol: String The currency symbol.

symbolPlacement: CurrencySymbolPlacementEnum = CurrencySymbolPlacementEnum.before()
Where to place the currency symbol.

smallSymbol: String The small amount currency symbol.

smallSymbolPlacement: CurrencySymbolPlacementEnum = CurrencySymbolPlacementEnum.after
Where to place the small amount currency symbol.

3.3.3 Operations

Reportable validate() validate

This method returns an object of type Reportable which will contain all the errors and warnings generated from this method. Below is a list of the axioms that must hold for an instance of this class to be valid. Each statement is followed by the error or warning message that will be issued if the axiom is violated.

In addition to the validation rules for the PrimitiveCommodityModel class, the following conditions apply:

- The symbol is defined and the symbolPlacement is defined. The symbol must not contain a number.
- If the smallSymbol is defined, then the number of decimals is greater than 0 and the smallSymbolPlacement is defined. The smallSymbol must not contain a number.

4 Enumerations

4.1 CommodityClassificationEnum

An enumeration describing the types of commodity. These types refer to the kind of control exerted over the commodity and, for abstract commodities, the kind of backing that they receive. The classification is taken from the usual classification of currencies.

4.1.1 Relationships

	Class	Description	Notes
⇑	Enum		

⇑:Inherits

4.1.2 Operations

«Static Method» CommodityClassificationEnum commodity() commodity

Returns an instance of this class with identifier “commodity”.

Commodities have some intrinsic value. Examples are wheat, gold or silver, or currencies minted from gold or silver, such as the Roman denarius or the English sovereign.

«Static Method» CommodityClassificationEnum fiduciary() fiduciary

Returns an instance of this class with identifier “fiduciary”.

Fiduciary commodities are essentially paper exchangeable for some commodity of intrinsic value. Examples of fiduciary currencies are the bank notes in circulation during the 18th and 19th century in Europe or US greenbacks prior to the US Civil War.

«Static Method» CommodityClassificationEnum fiat() fiat

Returns an instance of this class with identifier “fiat”.

Fiat commodities have no intrinsic value, but are backed as legal tender by some government or organization. Most modern currencies, such as francs, dollars or yen are fiat commodities.

«Static Method» Collection<Enum> elements() elements

The elements of this enumeration. Returns a collection containing the results of the commodity(), fiduciary() and fiat() operations.

4.2 CommodityRoundingEnum

An enumeration of rounding conventions for use by commodities.

4.2.1 Relationships

	Class	Description	Notes
⇑	Enum		

⇑:Inherits

4.2.2 Operations

«Static Method» CommodityRoundingEnum round() round

Returns an instance of this class with identifier “round”.

Assume that we have an amount of commodity x, a smallest amount of s and that the rounded amount is denoted by x'. $x' = \lfloor x/s + 0.5 \rfloor \times s$.

«Static Method» CommodityRoundingEnum up() up

Returns an instance of this class with identifier “up”.

Assume that we have an amount of commodity x, a smallest amount of s and that the rounded amount is denoted by x'. $x' = \lceil x/s \rceil \times s$.

«Static Method» CommodityRoundingEnum down() down

Returns an instance of this class with identifier ‘down”.

Assume that we have an amount of commodity x, a smallest amount of s and that the rounded amount is denoted by x'. $x' = \lfloor x/s \rfloor \times s$.

«Static Method» Collection<Enum> elements() elements

The elements of this enumeration. Returns a collection containing the results of the round(), up() and down() operations.

4.3 CurrencySymbolPlacementEnum

An enumeration giving whether a currency symbol is placed before or after the currency amount.

4.3.1 Relationships

	Class	Description	Notes
⇑	Enum		

⇑:Inherits

4.3.2 Operations

«Static Method» CurrencyPlacementEnum before() before

Returns an instance of this class with an identifier of “before”.

A placement type of before indicates that the symbol is placed before the amount of that currency.

«Static Method» CurrencyPlacementEnum after() after

Returns an instance of this class with an identifier of “after”.

A placement type of after indicates that the symbol is placed after the amount of that currency.

«Static Method» Collection<Enum> elements() elements

The elements of this enumeration. Returns a collection containing the results of the before() and after() operations.

5 Associations

Table 1: Basic Commodities— Associations

Association Role	Class	Card.	Notes
currency			
location	LocationModel	0..n	
currency	CurrencyModel §3.3	0..1	→
model			
model	Currency §2.3		→
wrapper	CurrencyReferenceDataModel §3.1		
authority			
authority	Organization	0..1	→
commodity	PrimitiveCommodityModel §3.2	0..n	
location			
location	Location	0..1	→
commodity	PrimitiveCommodityModel §3.2	0..n	

→:Navigable ◊:Aggregate ⧫:Composite

5.1 currency

Role: location LocationModel, 0..n.
Role: currency *Navigable* CurrencyModel, 0..1.
The local currency for this location.

5.2 model

Role: model *Navigable* Currency.
Role: wrapper CurrencyReferenceDataModel.
The model that provides the reference data's behavior.

5.3 authority

Role: authority *Navigable* Organization, 0..1.
Role: commodity PrimitiveCommodityModel, 0..n.
The organization that regulates this commodity. No associated organization indicates a commodity that is either unregulated, or regulated by a disparate group of organizations.

5.4 location

Role: location *Navigable* Location, 0..1.
Role: commodity PrimitiveCommodityModel, 0..n.

The location within which this commodity is normally traded. If this commodity is location-independent, then there is no association.

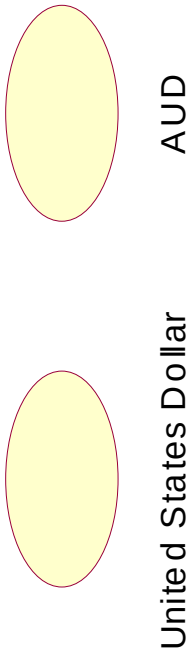


Figure 1: Class Diagram— Examples

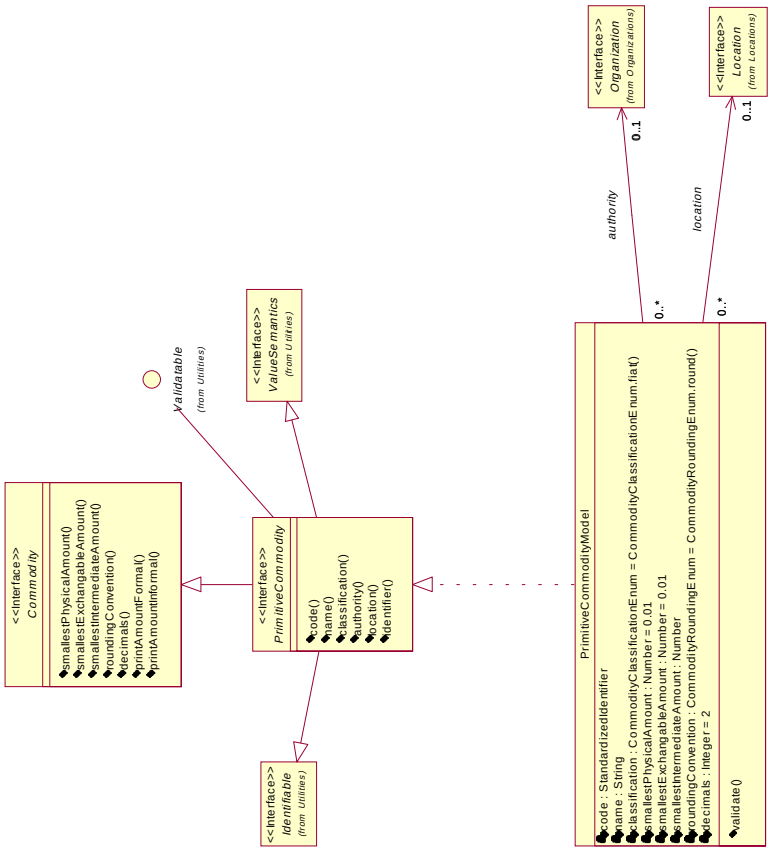


Figure 2: Class Diagram— Commodities

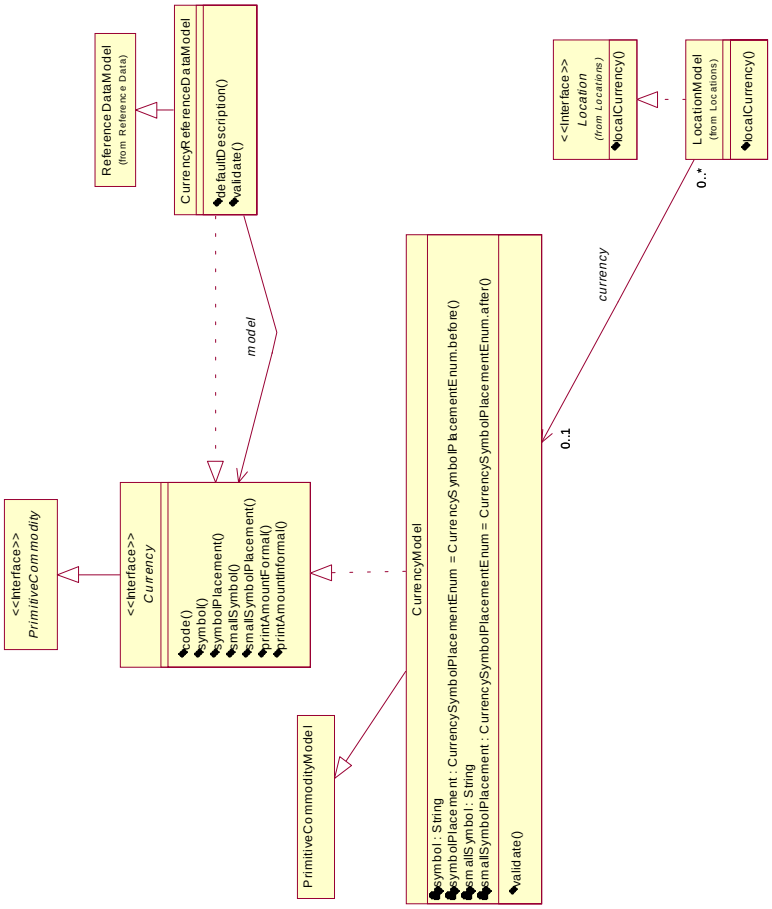


Figure 3: Class Diagram— Currencies

6 Extensions to the Locations Package

6.1 Location

6.1.1 Relationships

	Class	Description	Notes
↓	LocationModel		
↔	PrimitiveCommodityModel §3.2	location 0..n	

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

6.1.2 Operations

Currency localCurrency() localCurrency

The location's currency.

Returns the currency that this location commonly uses.

6.2 LocationModel

6.2.1 Relationships

	Class	Description	Notes
↑	Location		
↔	CurrencyModel §3.3	currency 0..1	→

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

6.2.2 Operations

Currency localCurrency() localCurrency

The location's currency.

Returns the associated currency, if there is one. Otherwise, it returns the currency of the parent location, if there is one. Otherwise, it returns nil.

References

[1] International Organization for Standardization (ISO). *Codes for the Representation of Currencies and Funds*, number ISO 4217, 1995. http://www.bsi.org.uk/bsi/news/tablea1.htm.