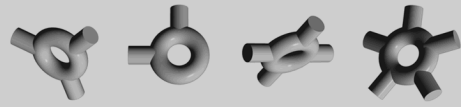


elements



## Reference Data Package

TARMS Inc.

September 07, 2000

Copyright ©2000 TARMS Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this model and associated documentation files (the “Model”), to deal in the Model without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Model, and to permit persons to whom the Model is furnished to do so, subject to the following conditions:

1. The origin of this model must not be misrepresented; you must not claim that you wrote the original model. If you use this Model in a product, an acknowledgment in the product documentation would be appreciated but is not required. Similarly notification of this Model’s use in a product would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice, including the above copyright notice shall be included in all copies or substantial portions of the Model.

THE MODEL IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE MODEL OR THE USE OR OTHER DEALINGS IN THE MODEL.

Typeset in L<sup>A</sup>T<sub>E</sub>X.

## Contents

<b>1</b>	<b>Interfaces</b>	<b>3</b>
1.1	ExternalReference . . . . .	3
1.1.1	Relationships . . . . .	3
1.1.2	Operations . . . . .	3
1.2	ExternalSource . . . . .	4
1.2.1	Relationships . . . . .	5
1.2.2	Operations . . . . .	5
1.3	ReferenceData . . . . .	6
1.3.1	Relationships . . . . .	6
1.3.2	Operations . . . . .	6
1.4	Visibility . . . . .	8
1.4.1	Relationships . . . . .	9
1.4.2	Operations . . . . .	9
<b>2</b>	<b>Classes</b>	<b>9</b>
2.1	ExternalReferenceModel . . . . .	9
2.1.1	Relationships . . . . .	9
2.1.2	Attributes . . . . .	9
2.1.3	Operations . . . . .	10
2.2	ExternalSourceModel . . . . .	10
2.2.1	Relationships . . . . .	10
2.2.2	Attributes . . . . .	10
2.2.3	Operations . . . . .	10
2.3	ReferenceDataModel . . . . .	11
2.3.1	Relationships . . . . .	11
2.3.2	Attributes . . . . .	11
2.3.3	Operations . . . . .	12
2.4	ExternalSourceReferenceDataModel . . . . .	13
2.4.1	Relationships . . . . .	13
2.5	VisibilityReferenceDataModel . . . . .	13
2.5.1	Relationships . . . . .	13
2.5.2	Operations . . . . .	13
2.6	VisibilityModel . . . . .	14
2.6.1	Relationships . . . . .	14
2.6.2	Attributes . . . . .	14
2.6.3	Operations . . . . .	14

<b>3</b>	<b>Associations</b>	<b>14</b>
3.1	model	15
3.2	visibility	15
3.3	maintenance	16
3.4	reference data	16
3.5	origin	16
3.6	sourcing	16
3.7	model	16
3.8	external reference	16
3.9	references	17
3.10	visibility	17

## List of Figures

1	Class Diagram— Reference Data	18
2	Class Diagram— External References	19
3	Class Diagram— Visibility	20

## List of Tables

1	Reference Data— Associations	14
1	... continued	15

## Package Description

Reference data represents information that is in some way external to the object model, usually something supplied by an external source or standard. Examples of reference data include such things as currency definitions or organization definitions; in both cases, the nature of this data is decided by entities outside those using the object model. Reference data is essentially static, although the dynamic update of this data is supported.

The scope and locality of reference data is important. Certain items, such as currencies have world-wide validity and can be maintained as part of a global reference data model. Other items of reference data, such as user definitions are essentially local to the business unit using the data and can be maintained locally.

Reference data is both versioned and likely to contain complex links between items of reference data, other pieces of reference data and other business objects. Reference data is often used as a base for constructing views and partitions of

other data — for example, a position hierarchy based on the book structure of the organization. Such views and partitions form an implicit cache of data, based on the *current* reference data. Since reference data is also dynamic data, care needs to be taken to ensure that systems can refresh this cached data when the underlying reference data changes.

## 1 Interfaces

### 1.1 ExternalReference

Data in this system may need to be imported from or exported to an external system of some sort. For example, a settled deal may need to be posted to a general ledger system. As another example, a feed of deals from some external brokerage system may be required. When importing and exporting data, the identifiers used for reference data within this system may not match the identifiers used in the external system. The ExternalReference interface provides a protocol for matching the reference data managed by this system to sets of possible external names.

It is possible that, when mapping reference data from this system onto an external system there are several possible matching pieces of external reference data. This overlap is likely wherever the external system draws finer distinctions between pieces of data than the local system. An external reference can be *preferred*, indicating that this external reference should be used in preference to any others when exporting to this external system.

#### 1.1.1 Relationships

	Class	Description	Notes
↑	Comparable		
↓	ExternalReferenceModel §2.1		
↔	ReferenceDataModel §2.3	external reference 1..1	
↔	ExternalSourceModel §2.2	references 1..1	

↑:Inherits ↓:Realized by ↔:Association →:Navigable ◇:Aggregate ◆:Composite

#### 1.1.2 Operations

##### ReferenceData referenceData()

referenceData

Associated reference data. Return the piece of reference data associated with this external reference.

<b>String key()</b>	key
External reference. Return the string which the external system uses to identify this piece of reference data.	
<b>ExternalSource source()</b>	source
External source. Return the source (external system) that produces or consumes this piece of external reference data.	
<b>Boolean isPreferred()</b>	isPreferred
Return true if this external reference for this source is the preferred representation for this source.	
<b>Boolean equals(Comparable arg)</b>	equals
<b>arg: Comparable</b>	
Equality relationship. Two external references are equal if they refer to the same external source and have the same key.	

## 1.2 ExternalSource

Objects which implement the ExternalSource interface can be used to indicate the origin of a piece of data. External sources have an identifier which is used internally, and by interfaces, to identify the ultimate source or destination of a piece of reference data.

Since every piece of reference data is associated with a source, this system is also associated with a source for consistency. The piece of source data associated with this system is known as the *local source*.

## 1.2.1 Relationships

	Class	Description	Notes
↑↑	Identifiable		
↓	ExternalSourceModel §2.2		
↓	ExternalSourceReferenceData-Model §2.4		
↔	ExternalReferenceModel §2.1	origin 0..n	
↔	ReferenceDataModel §2.3	sourcing 0..n	
↔	ExternalSourceReferenceData-Model §2.4	model 0..1	

↑:Inherits ↓:Realized by ↔:Association →:Navigable ◇:Aggregate ◆:Composite

## 1.2.2 Operations

### String identifier()

identifier

The source identifier. Return a string providing a unique name for the external system (eg. 'D2000' or 'Devon').

### Set<ReferenceData> referenceData(String reference)

referenceData

**reference:** String The key which we wish to match.

Match reference data. Return the set of reference data items which have reference as an external identifier and this source. All possible matches over all reference data classes will be returned. If no items match, return the empty set.

### Set<ReferenceData> referenceData(String reference, Set<Class> classes)

referenceData

**reference:** String The external key for this reference data.

**classes:** Set<Class> The acceptable reference data classes for returned reference data.

Match reference data. Return the set of reference data items from the set of classes which have reference as an external identifier and this source. If no items match, return the empty set.

### «Static Method» ExternalSource localSource()

localSource

Local source. Return the object which represents the local source of data — ie. this system.

### 1.3 ReferenceData

ReferenceData provides the basic interface for reference data, data that is not expected to change very rapidly.

Each piece of reference data in a class provides a user-readable identifier that forms the basis of pick-lists and reports. A longer description is also provided, to give a more complete description of what the reference data is for. For example, Australia may have the identifier “AU” and the description “Australia” or “Commonwealth of Australia”.

Reference data also provides the basic mechanism for identifying who is responsible for the maintenance of the data. Only that party (or parties) can change the data or re-assign it to another maintainer.

Reference data has a visibility, giving the domain of applicability of the reference data. See the Visibility §1.4 interface.

Each non-abstract class which implements ReferenceData must have two special instances: the *none* instance and the *anonymous* instance. The none instance is used wherever a “not applicable” or “not known” instance is required. The anonymous instance is used as a replacement for reference data that needs to be anonymized.

Reference data tends to come from sources external to the external system. See the ExternalSource §1.2 interface.

When importing or exporting data from other systems, the identifiers used internally need to be mapped from or to the identifiers that the external system uses. See the ExternalReference §1.1 interface.

#### 1.3.1 Relationships

	Class	Description	Notes
↑	DynamicData		
↑	Identifiable		
↓	ReferenceDataModel §2.3		
↔	ExternalReferenceModel §2.1	reference data	0..n

↑:Inherits ↓:Realized by ↔:Association →:Navigable ◇:Aggregate ◆:Composite

#### 1.3.2 Operations

##### String identifier()

identifier

Unique identifier. Returns the unique identifier for this piece of reference data.



<b>print(OutputStream stream)</b> <b>stream: OutputStream</b> The stream to print to. Print description. The identifier is added to the output stream.	print
<b>String description()</b> Long description. Returns a long description of what the reference data is or is for.	description
<b>String defaultDescription()</b> Default description of the reference data. In some cases, a default description of the reference data can be built from the underlying data. Return this built description, or an empty string if there is no default description.	defaultDescription
<b>Visibility visibility()</b> Instance visibility. Returns the visibility of this piece of reference data.	visibility
<b>Responsible maintainer()</b> Data maintainer. Returns the party responsible for maintenance of this piece reference data.	maintainer
<b>assignMaintainer(Responsible maintainer maintainer: Responsible)</b> The new maintainer. Reassign maintainer. Reassigns the party responsible for maintenance of this piece of reference data. Ownership of the reference data is moved to the domain of the new maintainer.	assignMaintainer
<b>Boolean isNone()</b> Return true if this is the special <i>none</i> instance.	isNone
<b>Boolean isAnonymous()</b> Returns true if this is the special <i>anonymous</i> instance.	isAnonymous
<b>ReferenceData asAnonymous()</b> Anonymize. Returns the anonymous instance that should be used to anonymize this piece of reference data.	asAnonymous
<b>Set&lt;ExternalReference&gt; externalReferences()</b>	externalReferences

External references for all sources. Return a set of all external references to this object for all possible sources of data.

**Set<ExternalReference> externalReferences(ExternalSource source)** externalReferences

**source: ExternalSource** The source for external references.

All external references for a particular source. Return a set of those external references associated with the given source and this piece of reference data.

**ExternalSource source()** source

Reference data source. Returns the source of this piece of reference data.

**«Static Method» ReferenceData none()** none

*None* instance. Return the special *none* instance for this class of reference data.

**«Static Method» ReferenceData anonymous()** anonymous

*Anonymous* instance. Return the special *anonymous* instance of this class of reference data.

**«Static Method» Visibility defaultVisibility()** defaultVisibility

Standard visibility. Returns the default visibility to be used when constructing this piece of reference data. The default visibility is likely to change with reference data class and user.

**«Static Method» Responsible defaultMaintainer()** defaultMaintainer

Standard maintaining entity. Returns the default entity responsible for maintaining this piece of reference data. Generally, this will be the current user.

## 1.4 Visibility

Reference data objects are associated with a visibility; the scope within the business of the data. The scope of visibility is expressed as a domain. This domain ranges from the top level domain, indicating truly global data (eg. currencies) to the local domain, for individual users.

At certain times, reference data needs to be exported beyond its intended domain. To ensure confidentiality and legislative requirements, the reference data may be *anonymized*, replaced with an anonymous piece of reference data intended to disguise the reference data.

### 1.4.1 Relationships

	Class	Description	Notes
↑	Comparable		
↑	Identifiable		
↓	VisibilityModel §2.6		
↓	VisibilityReferenceDataModel §2.5		
↔	ReferenceDataModel §2.3	visibility 0..n	

↑:Inherits ↓:Realized by ↔:Association →:Navigable ◇:Aggregate ◆:Composite

### 1.4.2 Operations

#### Domain domain()

domain

Domain of validity. Returns the domain within which this piece of data is valid.

#### Boolean anonymizeOutside()

anonymizeOut-  
side

Returns true if the referring piece of data should be anonymized when passing out of the domain within which it is valid.

## 2 Classes

### 2.1 ExternalReferenceModel

The ExternalReferenceModel provides an implementation of the ExternalReference interface.

#### 2.1.1 Relationships

	Class	Description	Notes
↑	ExternalReference §1.1		
↔	ReferenceData §1.3	reference data 1..1	→
↔	ExternalSource §1.2	origin 1..1	→

↑:Realizes ↔:Association →:Navigable ◇:Aggregate ◆:Composite

#### 2.1.2 Attributes

**key: String** The external reference key.

**isPreferred: Boolean** Preferred external reference representation

### 2.1.3 Operations

**ReferenceData referenceData()** referenceData

Associated reference data. Return the associated reference data.

**ExternalSource source()** source

External system source. Return the associated external source that produces/consumes this reference data.

## 2.2 ExternalSourceModel

The ExternalSourceModel implements the ExternalSource interface.

The implementation of the referenceData operation can be simple. However, a caching dictionary implementation is likely to be needed for any large implementation.

### 2.2.1 Relationships

	Class	Description	Notes
↑	ExternalSource §1.2		
↔	ExternalReference §1.1	references 0..n	→
↑:Realizes ↔:Association →:Navigable ◇:Aggregate ◆:Composite			

### 2.2.2 Attributes

**identifier: String** The source identifier

### 2.2.3 Operations

**Set<ReferenceData> referenceData(String reference)** referenceData

**reference: String** The key which we wish to match.

Match reference data. Return the set of reference data entries computed from the set of ExternalReference §1.1 objects associated with this source which match the supplied reference.

**Set<ReferenceData> referenceData(String reference, Set<Class> classes)** referenceData

**reference: String** The external key for this reference data.

**classes: Set<Class>** The acceptable reference data classes for returned reference data.

Match reference data. Return the set of reference data entries computed from the set of ExternalReference §1.1 objects associated with this source which match the supplied reference and the supplied classes.

## 2.3 ReferenceDataModel

The ReferenceDataModel class forms an abstract superclass for all reference data and implements the ReferenceData interface. Behavior for common information, such as visibility, maintainer, etc. is implemented in this class.

In most cases, a subclass of ReferenceDataModel is associated with an actual *model* class. The model class provides the actual functionality for some interface that the ReferenceDataModel subclass implements. The model class and reference data class can then be used interchangeably, allowing both static reference data management and on-the-fly models.

### 2.3.1 Relationships

	Class	Description	Notes
↑	DynamicDataModel		
↑	ReferenceData §1.3		
↑	Validatable		
↓	VisibilityReferenceDataModel §2.5		
↓	ExternalSourceReferenceData-Model §2.4		
↔	Visibility §1.4	visibility 1..1	→
↔	Responsible	maintenance 1..1	→
↔	ExternalSource §1.2	sourcing 1..1	→
↔	ExternalReference §1.1	external reference 0..n	→

↑:Inherits ↓:Inherited by ↑:Realizes ↔:Association →:Navigable ◇:Aggregate ◆:Composite

### 2.3.2 Attributes

**description: String** The long description of this piece of reference data. This attribute may be nil, indicating that the defaultDescription is used.

### 2.3.3 Operations

#### **Reportable validate()**

validate

Validate this piece of data.

- No two pieces of reference data in the same class may have the same identifiers; this test is provided by the container of reference data, rather than by an item of reference data.
- At most one of the external references from each source may be preferred.

#### **String identifier()**

identifier

Unique identifier. Abstract. Implemented by any subclass of this class. Generally, the associated model will be Identifiable and have an identifier that can be used as the identifier for this piece of reference data.

#### **String description()**

description

Long description. Return the description attribute, if it is not nil. If the description attribute is nil, return the result of the defaultDescription() operation.

#### **Visibility visibility()**

visibility

Domain of visibility. Return the associated visibility.

#### **Responsible maintainer()**

maintainer

Maintainer. Return the associated maintainer.

#### **ExternalSource source()**

source

The source of this reference data. Return the associated source.

#### **«Static Method» Visibility defaultVisibility()**

defaultVisibility

Standard visibility. Return visibility indicating the local Domain. This operation may be overridden by more sophisticated subclasses.

#### **«Static Method» Responsible defaultMaintainer()**

defaultMaintainer

Standard maintaining entity. If the system has a notion of current user, then return the current user. Otherwise return the party indicating reference data maintenance responsibility.

## 2.4 ExternalSourceReferenceDataModel

The ExternalSourceReferenceDataModel class implements the ExternalSource §1.2 interface as a piece of reference data by holding an ExternalSource object. Implementation of ExternalSource consists of delegating to the held model.

### 2.4.1 Relationships

	Class	Description	Notes
↑	ReferenceDataModel §2.3		
↑	ExternalSource §1.2		
↔	ExternalSource §1.2	model 1..1	→

↑:Inherits ↑:Realizes ↔:Association →:Navigable ◇:Aggregate ◆:Composite

## 2.5 VisibilityReferenceDataModel

This class acts as a holder for VisibilityModel §2.6 instances, allowing the models to be managed as part of static data. The Visibility §1.4 interface is implemented by passing queries on to the held VisibilityModel.

### 2.5.1 Relationships

	Class	Description	Notes
↑	ReferenceDataModel §2.3		
↑	Visibility §1.4		
↑	Validatable		
↔	VisibilityModel §2.6	model 1..1	→

↑:Inherits ↑:Realizes ↔:Association →:Navigable ◇:Aggregate ◆:Composite

### 2.5.2 Operations

#### String description()

description

Long description. If the description attribute is not nil, then return the description attribute.

If the description attribute is nil, then construct a description by printing the domain of visibility, followed by 'Anonymize' if anonymizeOutside() returns true.

#### Reportable validate()

validate

Validate this visibility. A valid visibility must have a valid domain. Multiple versions with the same domain and anonymizeOutside values are permitted.

## 2.6 VisibilityModel

The VisibilityModel class provides a concrete implementation of the Visibility interface.

### 2.6.1 Relationships

	Class	Description	Notes
↑	Visibility §1.4		
↔	VisibilityReferenceDataModel §2.5	model 0..1	
↔	Domain	visibility 1..1	→
↑:Realizes ↔:Association →:Navigable ◇:Aggregate ◆:Composite			

### 2.6.2 Attributes

**anonymizeOutside: Boolean = false** Indicates whether data moving outside this domain should be anonymized.

**identifier: String** The identifier for this visibility.

### 2.6.3 Operations

**Domain domain()** domain  
Domain of visibility. Return the associated domain of visibility.

## 3 Associations

Table 1: Reference Data— Associations

Association	Role	Class	Card.	Notes
model	model	VisibilityModel §2.6	1..1	→
	reference data	VisibilityReferenceDataModel §2.5	0..1	
visibility	visibility	Visibility §1.4	1..1	→
	theReferenceDataModel	ReferenceDataModel §2.3	0..n	



Table 1: ... continued

Association	Role	Class	Card.	Notes
maintenance				
	maintainer	Responsible	1..1	→
	reference data	ReferenceDataModel §2.3	0..n	
reference data				
	knownAs	ExternalReferenceModel §2.1	0..n	
	referencedBy	ReferenceData §1.3	1..1	→
origin				
	source	ExternalSource §1.2	1..1	→
	reference	ExternalReferenceModel §2.1	0..n	
sourcing				
	source	ExternalSource §1.2	1..1	→
	reference data	ReferenceDataModel §2.3	0..n	
model				
	model	ExternalSource §1.2	1..1	→
	reference model	ExternalSourceReferenceData-Model §2.4	0..1	
external reference				
	knownAs	ExternalReference §1.1	0..n	→
	referencedBy	ReferenceDataModel §2.3	1..1	
references				
	external reference	ExternalReference §1.1	0..n	→
	source	ExternalSourceModel §2.2	1..1	
visibility				
	theDomain	Domain	1..1	→
	visibility	VisibilityModel §2.6	0..n	

→:Navigable ◇:Aggregate ◆:Composite

### 3.1 model

**Role: model** *Navigable* VisibilityModel, 1..1.

**Role: reference data** VisibilityReferenceDataModel, 0..1.

The held model for the reference data model.

### 3.2 visibility

**Role: visibility** *Navigable* Visibility, 1..1.

**Role: theReferenceDataModel** ReferenceDataModel, 0..n.  
The visibility of this piece of reference data.

### 3.3 maintenance

**Role: maintainer** *Navigable* Responsible, 1..1.

**Role: reference data** ReferenceDataModel, 0..n.

The party responsible for the maintenance of this reference data.

### 3.4 reference data

**Role: knownAs** ExternalReferenceModel, 0..n.

**Role: referencedBy** *Navigable* ReferenceData, 1..1.

The piece of reference data that an external reference refers to.

### 3.5 origin

**Role: source** *Navigable* ExternalSource, 1..1.

**Role: reference** ExternalReferenceModel, 0..n.

The source of the external reference.

### 3.6 sourcing

**Role: source** *Navigable* ExternalSource, 1..1.

**Role: reference data** ReferenceDataModel, 0..n.

The system that is the source of this piece of reference data

### 3.7 model

**Role: model** *Navigable* ExternalSource, 1..1.

**Role: reference model** ExternalSourceReferenceDataModel, 0..1.

The wrapped external source.

### 3.8 external reference

**Role: knownAs** *Navigable* ExternalReference, 0..n.

**Role: referencedBy** ReferenceDataModel, 1..1.

The external names that this piece of reference data can be known by.

### 3.9 references

**Role: external reference** *Navigable* ExternalReference, 0..n.

**Role: source** ExternalSourceModel, 1..1.

The external references that come from this source.

### 3.10 visibility

**Role: theDomain** *Navigable* Domain, 1..1.

**Role: visibility** VisibilityModel, 0..n.

The domain in which a visibility applies.

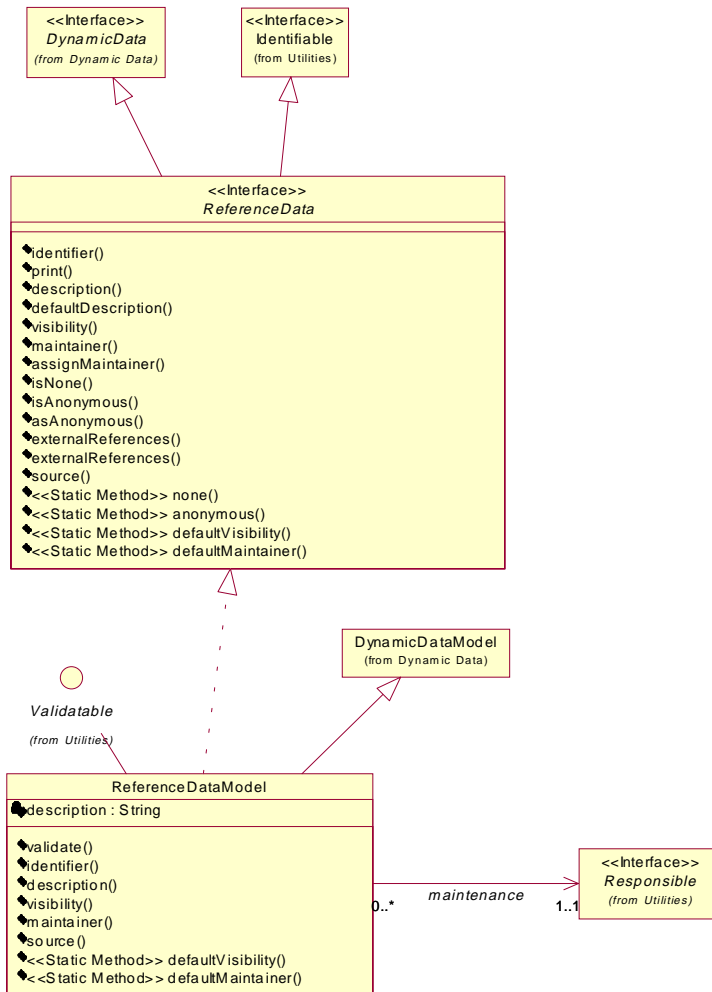


Figure 1: Class Diagram— Reference Data

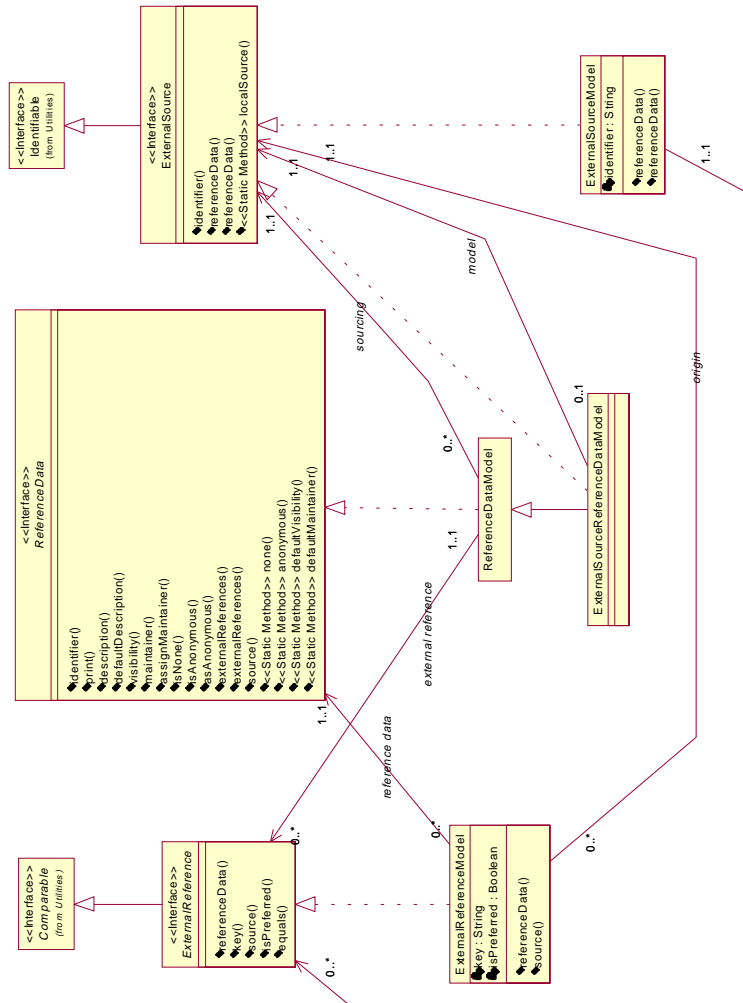


Figure 2: Class Diagram— External References



## References