|  |
| --- |
|  |
| |  |  |  | | --- | --- | --- | | 75 | Date: | October 2014 | |
| Archetype Modeling Language (AML)  Version: 0.5  **OMG Document Number: formal/2014-10-17**  **Standard document URL: http://www.omg.org/spec/AML/1.0**  Original File: N/A |

USE OF SPECIFICATION - TERMS, CONDITIONS & NOTICES

The material in this document details an Object Management Group specification in accordance with the terms, conditions and notices set forth below. This document does not represent a commitment to implement any portion of this specification in any company&apos;s products. The information contained in this document is subject to change without notice.

LICENSES

The companies listed above have granted to the Object Management Group, Inc. (OMG) a nonexclusive, royalty-free, paid up, worldwide license to copy and distribute this document and to modify this document and distribute copies of the modified version. Each of the copyright holders listed above has agreed that no person shall be deemed to have infringed the copyright in the included material of any such copyright holder by reason of having used the specification set forth herein or having conformed any computer software to the specification.

Subject to all of the terms and conditions below, the owners of the copyright in this specification hereby grant you a fully-paid up, non-exclusive, nontransferable, perpetual, worldwide license (without the right to sublicense), to use this specification to create and distribute software and special purpose specifications that are based upon this specification, and to use, copy, and distribute this specification as provided under the Copyright Act; provided that: (1) both the copyright notice identified above and this permission notice appear on any copies of this specification; (2) the use of the specifications is for informational purposes and will not be copied or posted on any network computer or broadcast in any media and will not be otherwise resold or transferred for commercial purposes; and (3) no modifications are made to this specification. This limited permission automatically terminates without notice if you breach any of these terms or conditions. Upon termination, you will destroy immediately any copies of the specifications in your possession or control.

PATENTS

The attention of adopters is directed to the possibility that compliance with or adoption of OMG specifications may require use of an invention covered by patent rights. OMG shall not be responsible for identifying patents for which a license may be required by any OMG specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. OMG specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

GENERAL USE RESTRICTIONS

Any unauthorized use of this specification may violate copyright laws, trademark laws, and communications regulations and statutes. This document contains information which is protected by copyright. All Rights Reserved. No part of this work covered by copyright herein may be reproduced or used in any form or by any means--graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems--without permission of the copyright owner.

DISCLAIMER OF WARRANTY

WHILE THIS PUBLICATION IS BELIEVED TO BE ACCURATE, IT IS PROVIDED "AS IS" AND MAY CONTAIN ERRORS OR MISPRINTS. THE OBJECT MANAGEMENT GROUP AND THE COMPANIES LISTED ABOVE MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS PUBLICATION, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE OR USE.   
  
IN NO EVENT SHALL THE OBJECT MANAGEMENT GROUP OR ANY OF THE COMPANIES LISTED ABOVE BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, RELIANCE OR COVER DAMAGES, INCLUDING LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY ANY USER OR ANY THIRD PARTY IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.  
  
The entire risk as to the quality and performance of software developed using this specification is borne by you. This disclaimer of warranty constitutes an essential part of the license granted to you to use this specification.

RESTRICTED RIGHTS LEGEND

Use, duplication or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c) (1) (ii) of The Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 or in subparagraph (c)(1) and (2) of the Commercial Computer Software - Restricted Rights clauses at 48 C.F.R. 52.227-19 or as specified in 48 C.F.R. 227-7202-2 of the DoD F.A.R. Supplement and its successors, or as specified in 48 C.F.R. 12.212 of the Federal Acquisition Regulations and its successors, as applicable. The specification copyright owners are as indicated above and may be contacted through the Object Management Group, 140 Kendrick Street, Needham, MA 02494, U.S.A.

TRADEMARKS

MDA®, Model Driven Architecture®, UML®, UML Cube logo®, OMG Logo®, CORBA® and XMI® are registered trademarks of the Object Management Group, Inc., and Object Management Group©, OMG© , Unified Modeling Language©, Model Driven Architecture Logo©, Model Driven Architecture Diagram©, CORBA logo©, XMI Logo©, CWM©, CWM Logo©, IIOP© , MOF© , OMG Interface Definition Language (IDL)© , and OMG SysML© are trademarks of the Object Management Group. All other products or company names mentioned are used for identification purposes only, and may be trademarks of their respective owners.

COMPLIANCE

The copyright holders listed above acknowledge that the Object Management Group (acting itself or through its designees) is and shall at all times be the sole entity that may authorize developers, suppliers and sellers of computer software to use certification marks, trademarks or other special designations to indicate compliance with these materials.  
  
Software developed under the terms of this license may claim compliance or conformance with this specification if and only if the software compliance is of a nature fully matching the applicable compliance points as stated in the specification. Software developed only partially matching the applicable compliance points may claim only that the software was based on this specification, but may not claim compliance or conformance with this specification. In the event that testing suites are implemented or approved by Object Management Group, Inc., software developed using this specification may claim compliance or conformance with the specification only if the software satisfactorily completes the testing suites.

**OMG's Issue Reporting Procedure**

All OMG specifications are subject to continuous review and improvement. As part of this process we encourage readers to report any ambiguities, inconsistencies, or inaccuracies they may find by completing the Issue Reporting Form listed on the main web page http://www.omg.org, under Documents, Report a Bug/Issue (http://www.omg.org/technology/agreement.)

**Acknowledgements**

The following individuals submitted parts of this specification and/or have assisted the AML team in the development of the specification:

Table of Contents

1 5

2 6

2.1 6

2.2 6

2.2.1 6

2.2.2 6

**Preface**

**OMG**

Founded in 1989, the Object Management Group, Inc. (OMG) is an open membership, not-for-profit computer industry standards consortium that produces and maintains computer industry specifications for interoperable, portable, and reusable enterprise applications in distributed, heterogeneous environments. Membership includes Information Technology vendors, end users, government agencies, and academia.  
  
OMG member companies write, adopt, and maintain its specifications following a mature, open process. OMG™s specifications implement the Model Driven Architecture (MDA®), maximizing ROI through a full-lifecycle approach to enterprise integration that covers multiple operating systems, programming languages, middleware and networking infrastructures, and software development environments. OMG™s specifications include: UML® (Unified Modeling Language); CORBA® (Common Object Request Broker Architecture); CWM (Common Warehouse Metamodel); and industry-specific standards for dozens of vertical markets.  
  
More information on the OMG is available at http://www.omg.org/.

**OMG Specifications**

As noted, OMG specifications address middleware, modeling and vertical domain frameworks. A Specifications Catalog is available from the OMG website at:

*http://www.omg.org/technology/documents/spec\_catalog.htm*

Specifications within the Catalog are organized by the following categories:

**OMG Modeling Specifications**

• UML  
• MOF  
• XMI  
• CWM  
• Profile specifications

**OMG Middleware Specifications**

• CORBA/IIOP  
• IDL/Language Mappings  
• Specialized CORBA specifications  
• CORBA Component Model (CCM)

**Platform Specific Model and Interface Specifications**

• CORBAservices  
• CORBAfacilities  
• OMG Domain specifications  
• OMG Embedded Intelligence specifications  
• OMG Security specifications

OMG Headquarters   
 109 Highland Ave,   
 Needham, MA 02494 USA  
 USA   
   
 Tel: +1-781-444-0404   
 Fax: +1-781-444-0320   
 Email: pubs@omg.org  
   
Certain OMG specifications are also available as ISO standards. Please consult http://www.iso.org

**Typographical Conventions**

The type styles shown below are used in this document to distinguish programming statements from ordinary English. However, these conventions are not used in tables or section headings where no distinction is necessary.

Times/Times New Roman - 10 pt.: Standard body text

**Helvetica/Arial - 10 pt. Bold: OMG Interface Definition Language (OMG IDL) and syntax elements.**

Courier - 10 pt. Bold: Programming language elements.

Helvetica/Arial - 10 pt : Exceptions

NOTE: Terms that appear in italics are defined in the glossary. Italic text also represents the name of a document, specification, or other publication.

# Scope

# Conformance

# Normative References

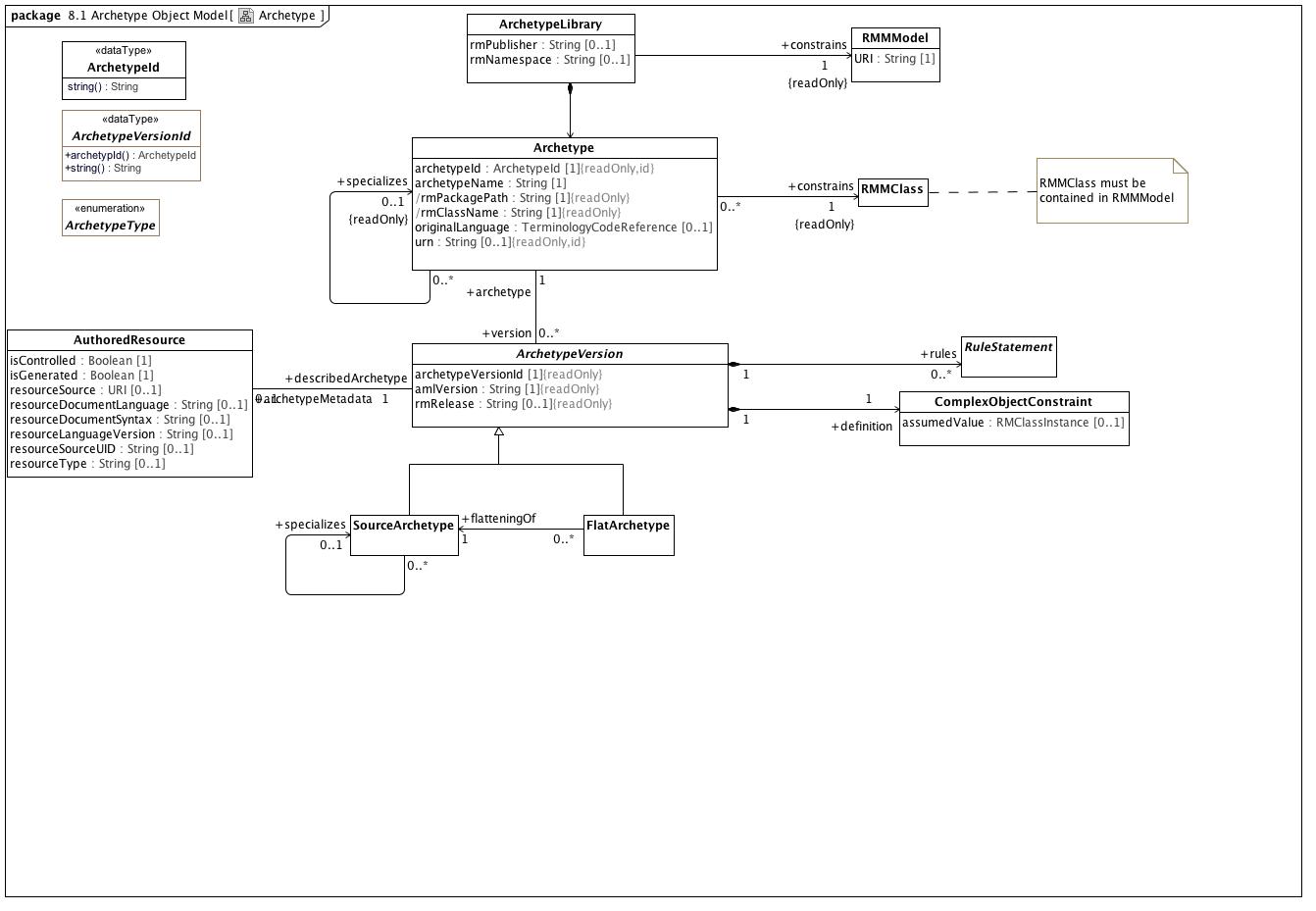
# Terms and Definitions

# Symbols

# Additional Information

# AML Object Model

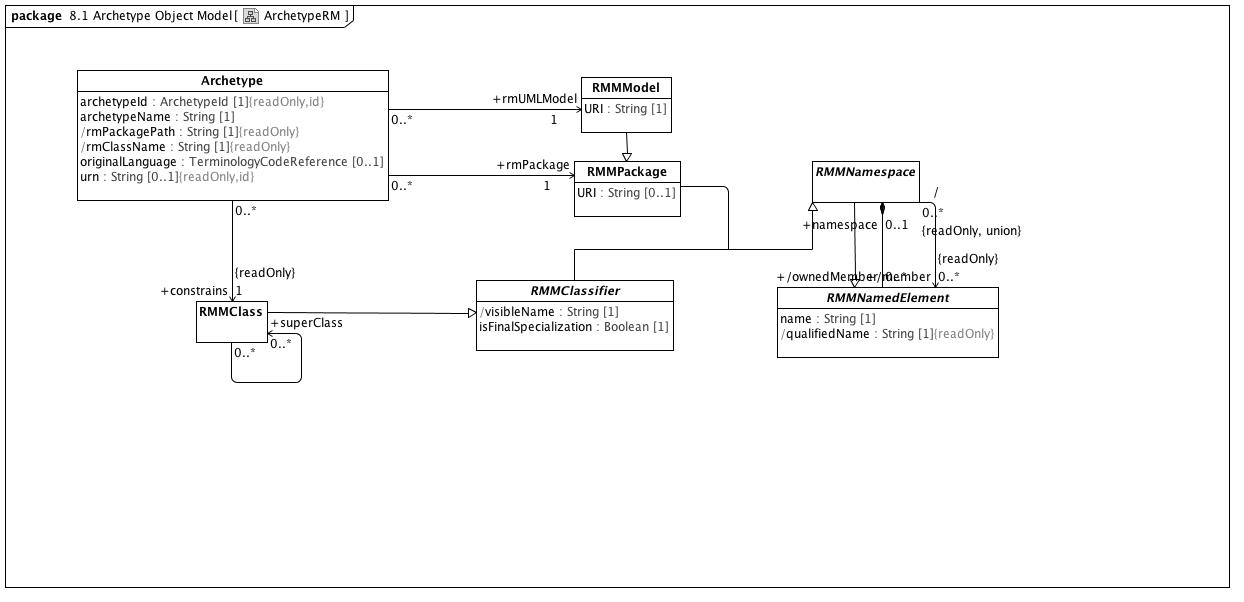
## Archetype Object Model



**Archetype**

The Archetype package, showing the compositions rules, definition, archetypeId, and terminology; the concrete kinds of archetypes; and lineage by the recursive parent relationship.

[Interfaces = 0, Classes = 10, Enumerations = 1, DataTypes=2, StereoTypes=0]



**ArchetypeRM**

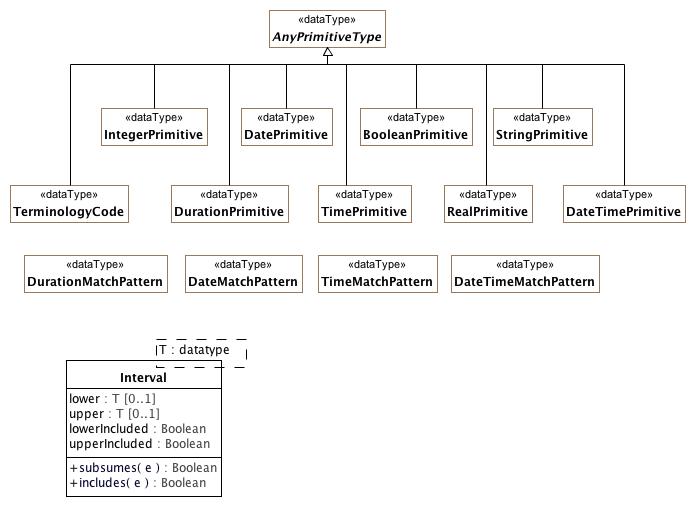
An Archetype references (or constrains) a single class in a UML Reference Model. The rmURI identifies the particular model, the rmPackage names the path to the (a) package that has the constrained class as a member, and the rmClassName identifies the particular class being constrained.

[Interfaces = 0, Classes = 7, Enumerations = 0, DataTypes=0, StereoTypes=0]

## Reference Object Model

### Primitive Data Types

In the AML/ADL context, the term "primitive data type" is used to indicate "leaf nodes" -- data elements that are treated as being atomic and are only constrained in terms of their possible value ranges. AML data types, like UML data types, are "model Types whose instances are distinguished only by their value" but, unlike the UML definition of "Primitive Type", AML primitive types can embody the notion of substructure. When a AML profile is applied to a UML Reference Model, it may be necessary to map one or more of the AML types to corresponding types in the target model.

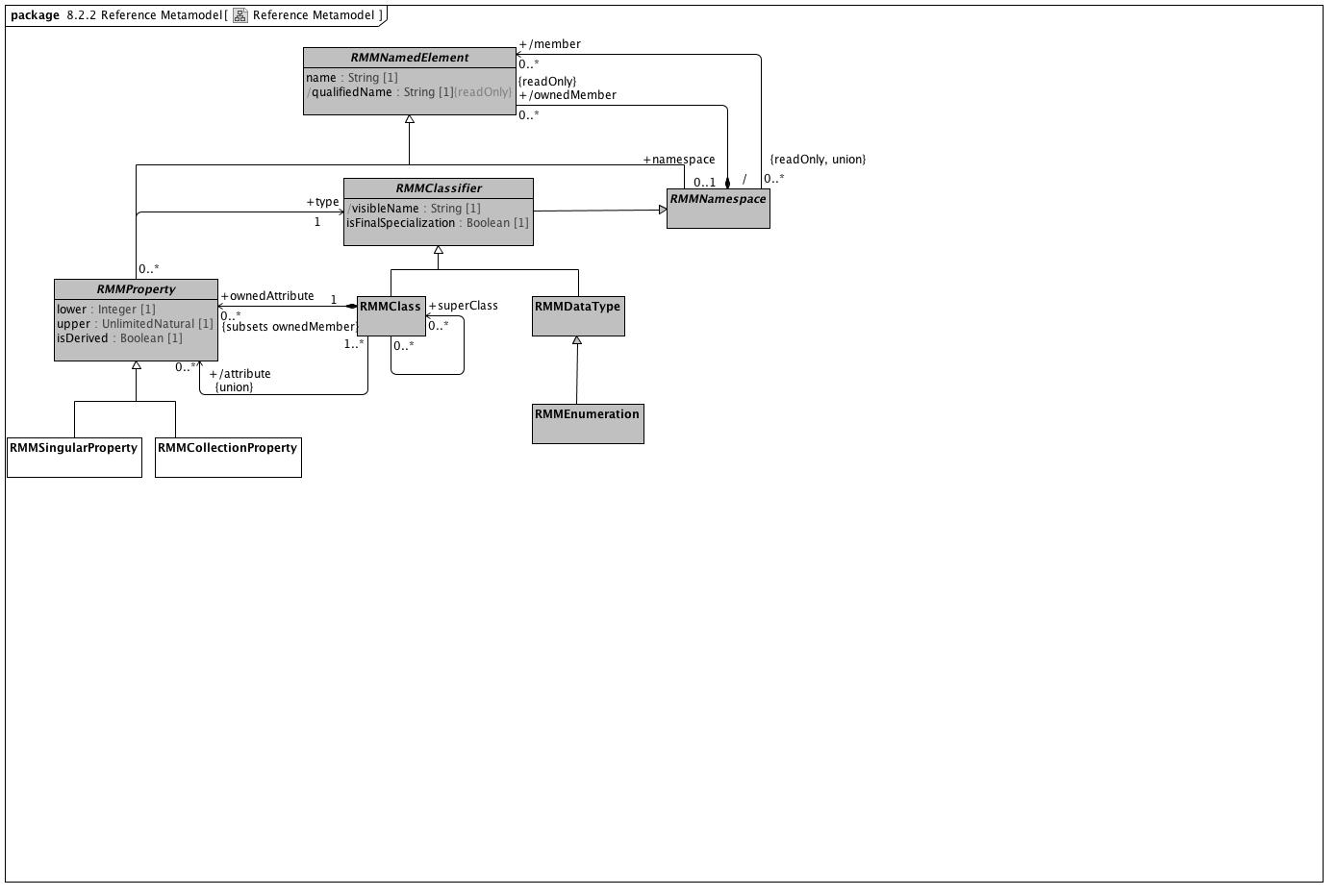


**PrimitiveDataTypes**

The set of primitive data types that can appear on a reference model and can be constrained using AML primitive constraints.

[Interfaces = 0, Classes = 1, Enumerations = 0, DataTypes=14, StereoTypes=0]

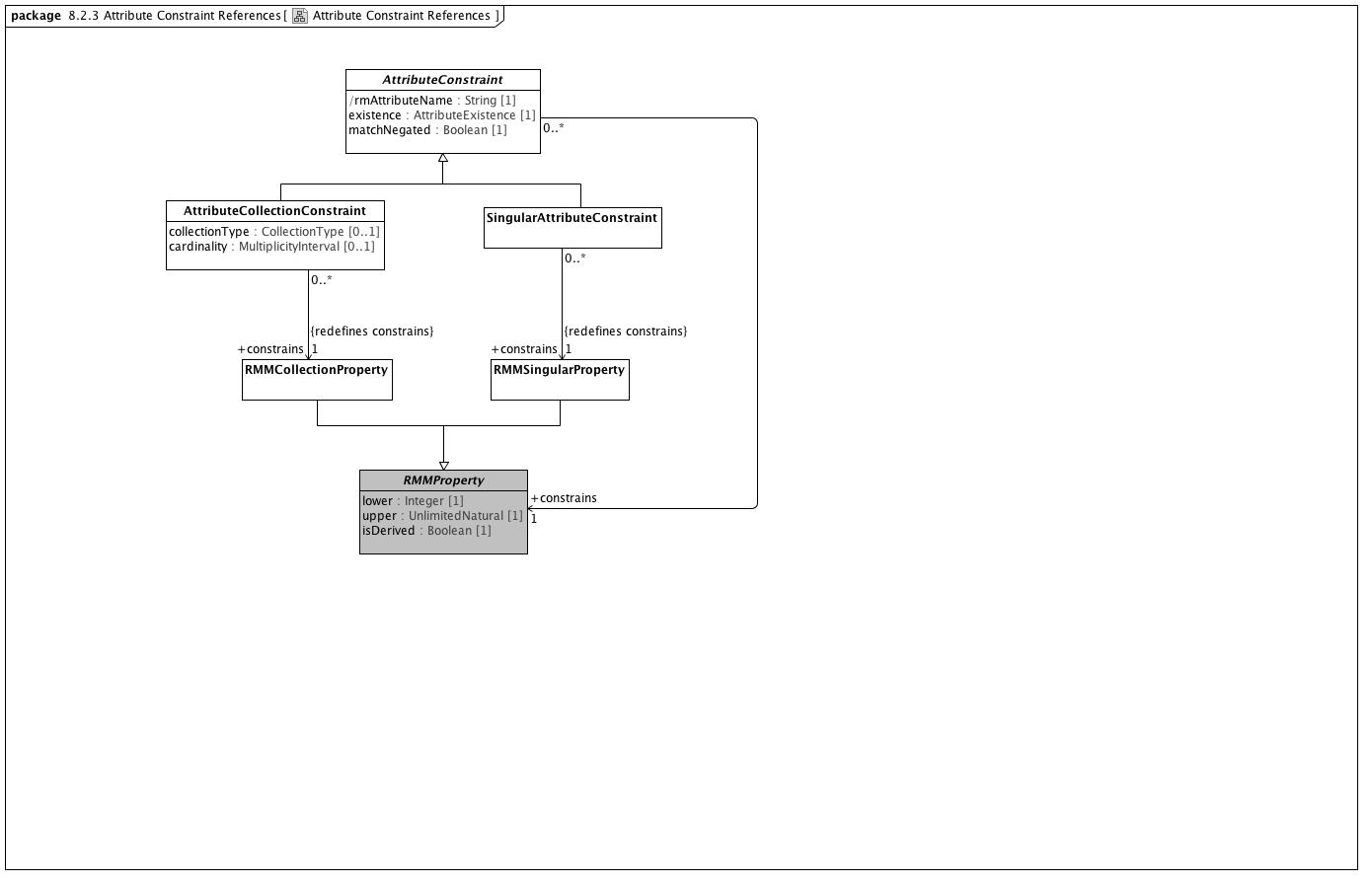
### Reference Metamodel



**Reference Metamodel**

[Interfaces = 0, Classes = 9, Enumerations = 0, DataTypes=0, StereoTypes=0]

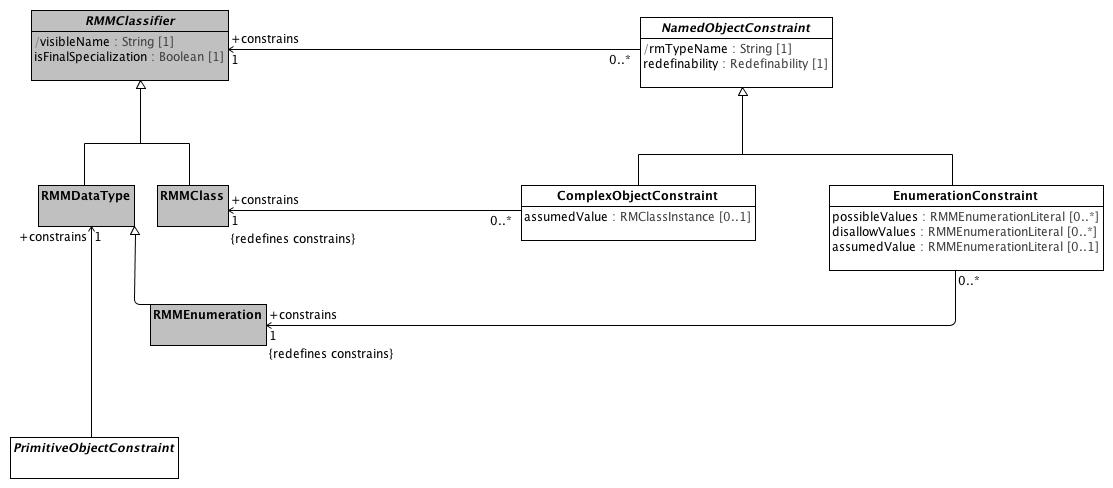
### Attribute Constraint References



**Attribute Constraint References**

[Interfaces = 0, Classes = 6, Enumerations = 0, DataTypes=0, StereoTypes=0]

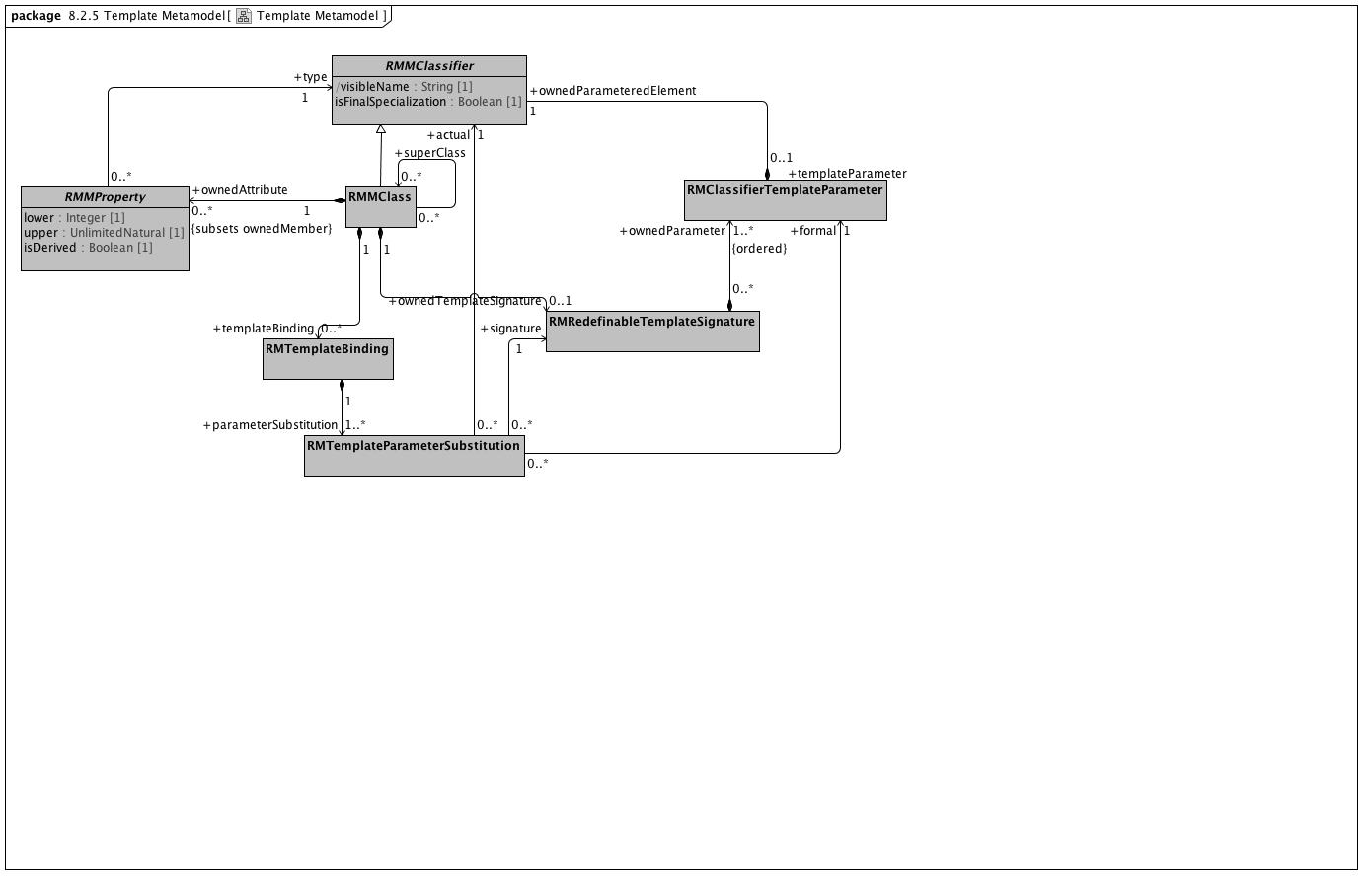
### Object Constraint References



**Object Constraint References**

[Interfaces = 0, Classes = 8, Enumerations = 0, DataTypes=0, StereoTypes=0]

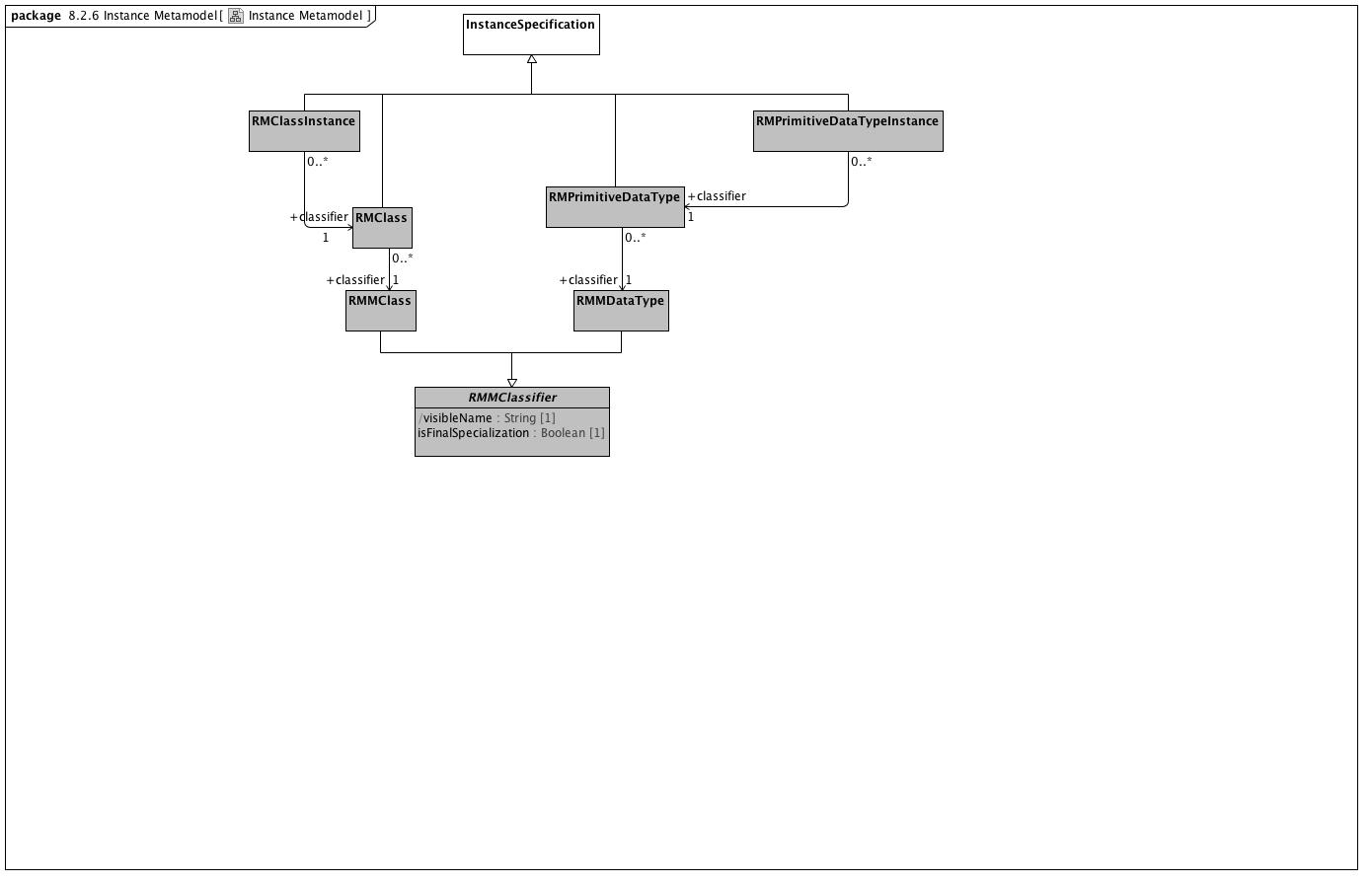
### Template Metamodel



**Template Metamodel**

[Interfaces = 0, Classes = 7, Enumerations = 0, DataTypes=0, StereoTypes=0]

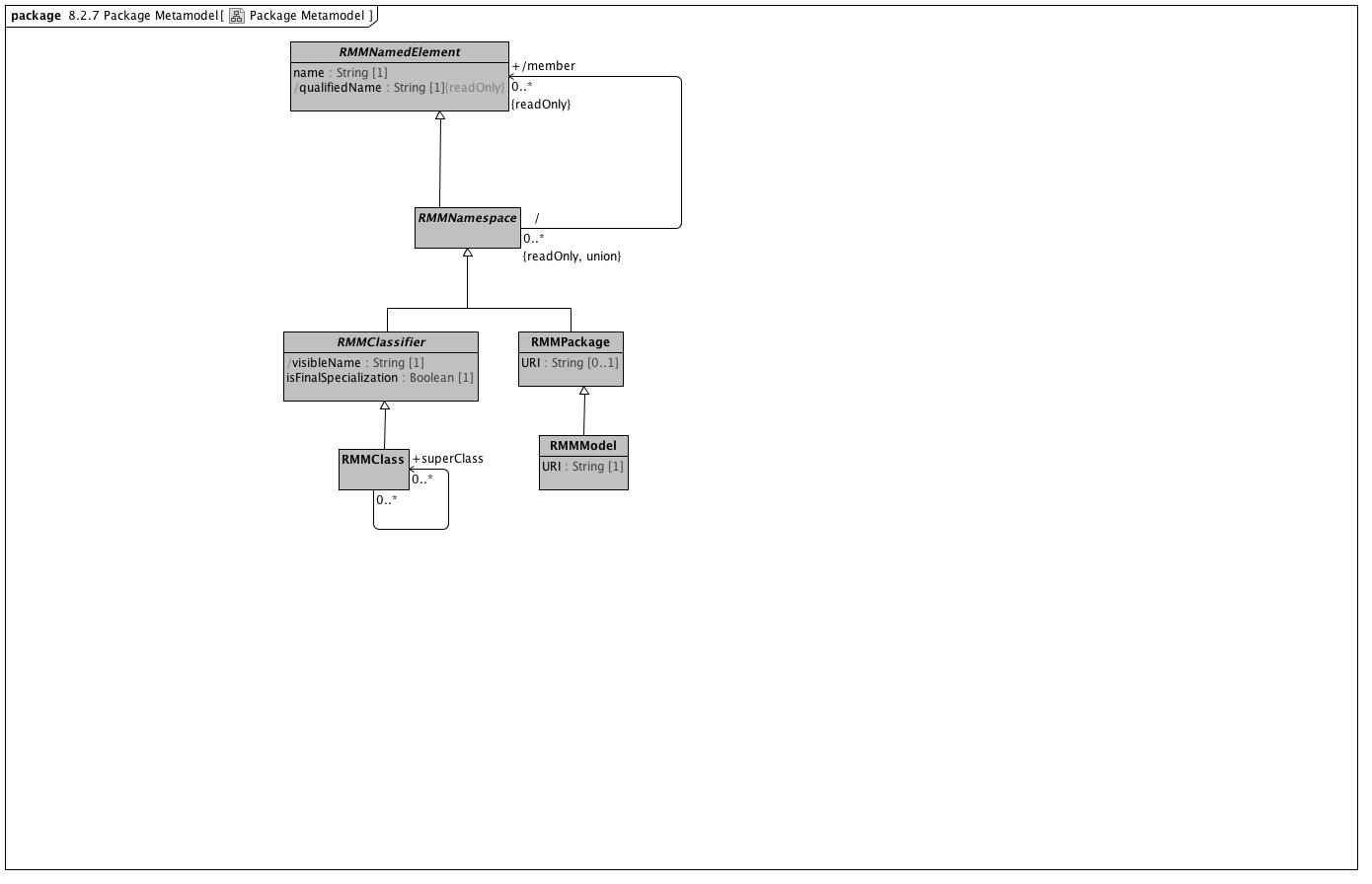
### Instance Metamodel



**Instance Metamodel**

[Interfaces = 0, Classes = 8, Enumerations = 0, DataTypes=0, StereoTypes=0]

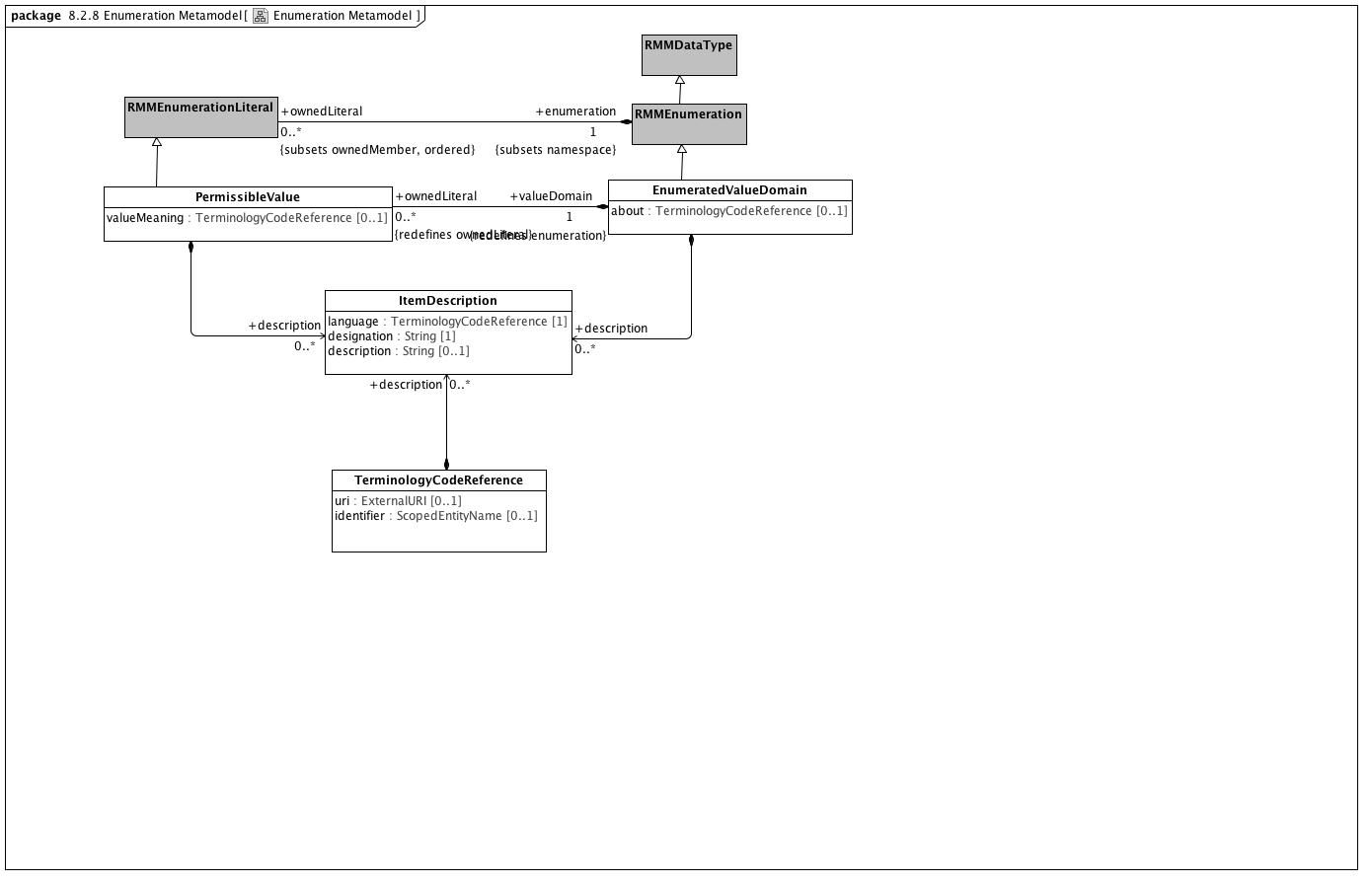
### Package Metamodel



**Package Metamodel**

[Interfaces = 0, Classes = 6, Enumerations = 0, DataTypes=0, StereoTypes=0]

### Enumeration Metamodel



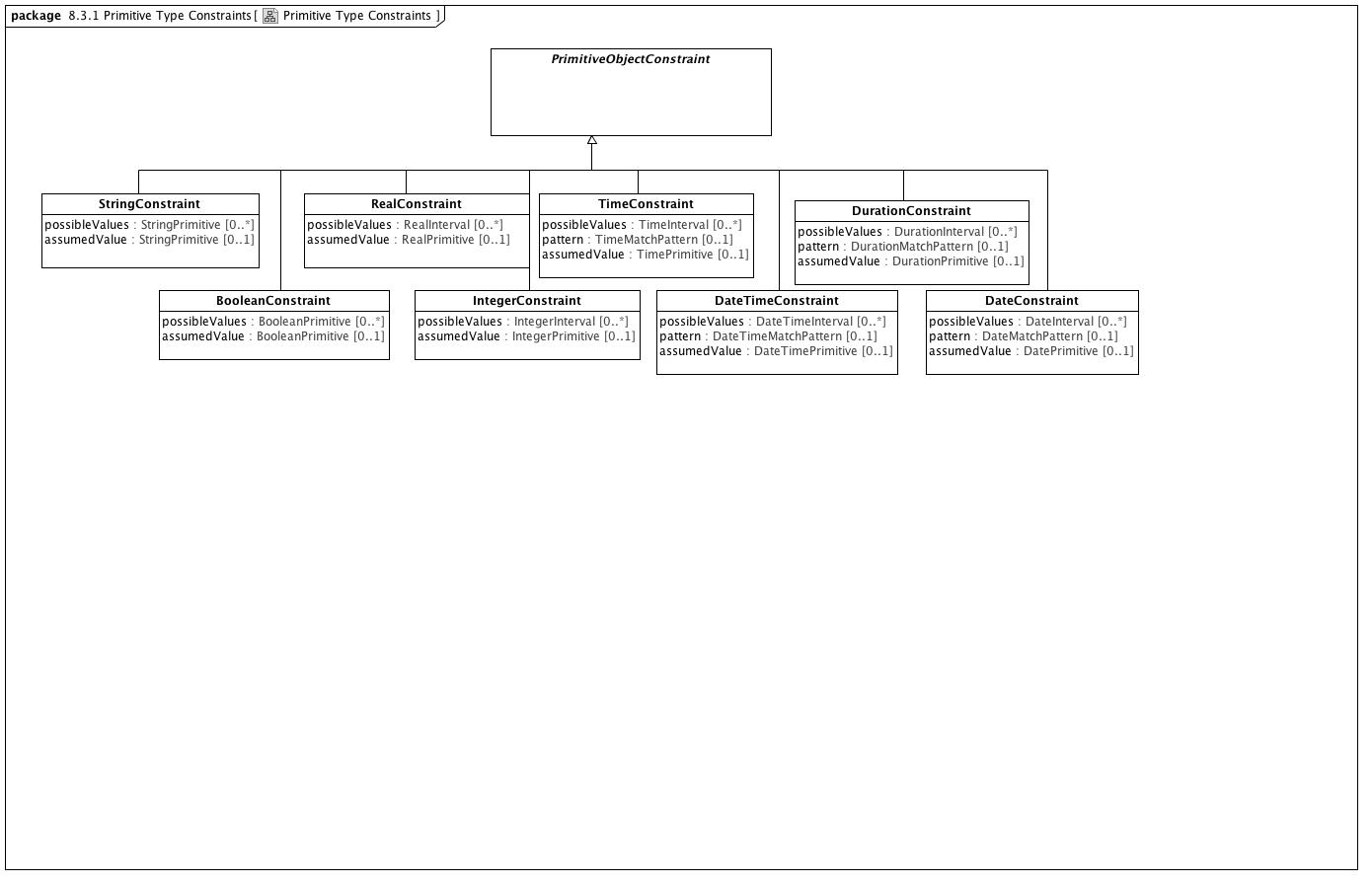
**Enumeration Metamodel**

[Interfaces = 0, Classes = 7, Enumerations = 0, DataTypes=0, StereoTypes=0]

## Constraint Object Model

Constraint Model Package Overview The constraint model is the core of the archetype design. It illustrates how constraints are defined, showing the object-attribute-object pattern characteristic of object constraints. ComplexObjectConstraint. Because objects are composed of properties (attributes and relationships), and properties consist of objects, the archetype definition consists of alternate layers of ArchetypeRootConstraint, but rather than a single archetype, it defines a set of archetypes. It can be thought of like a keyhole, into which few or many keys might fit, depending on how specific its shape is. Logically it has the same semantics as a ComplexObjectConstraint, except that the constraints are expressed in another archetype, not the current one. ComplexObjectConstraints PrimitiveObjectConstraints AttributeConstraints SingularAttributeConstraint class. Where a SingularAttributeConstraint is associated with more than one ObjectConstraint, the ObjectConstraints are alternatives. AttributeCollectionConstraint, which differentiates between unique and repeatable and between ordered and unordered collections. In addition, while the AttributeConstraint determines whether a property may exist, the quantity of a repeating element is defined in the AttributeCollectionConstraint’s cardinality property. AttributeCollectionConstraint, with its defined cardinality, there may be different sets of sibling members with different constraints, and the number of each of these subsets is specified as the AttributeCollectionMember’s occurrences property. In an organization, for instance, the cardinality for the member property may be “two or more,” but within that set of members, we may have two constraints. One type of member, the leader (indicated by an ObjectConstraint on the person type or role), may be required to occur exactly once, whereas other types may have multiple occurrences. AttributeTupleConstraints AttributeTuple would be defined for each pair of the two values (unit code and numeric ceiling), and these tuples would be grouped into an AttributeTupleConstraint, defining an array of acceptable sets of values.

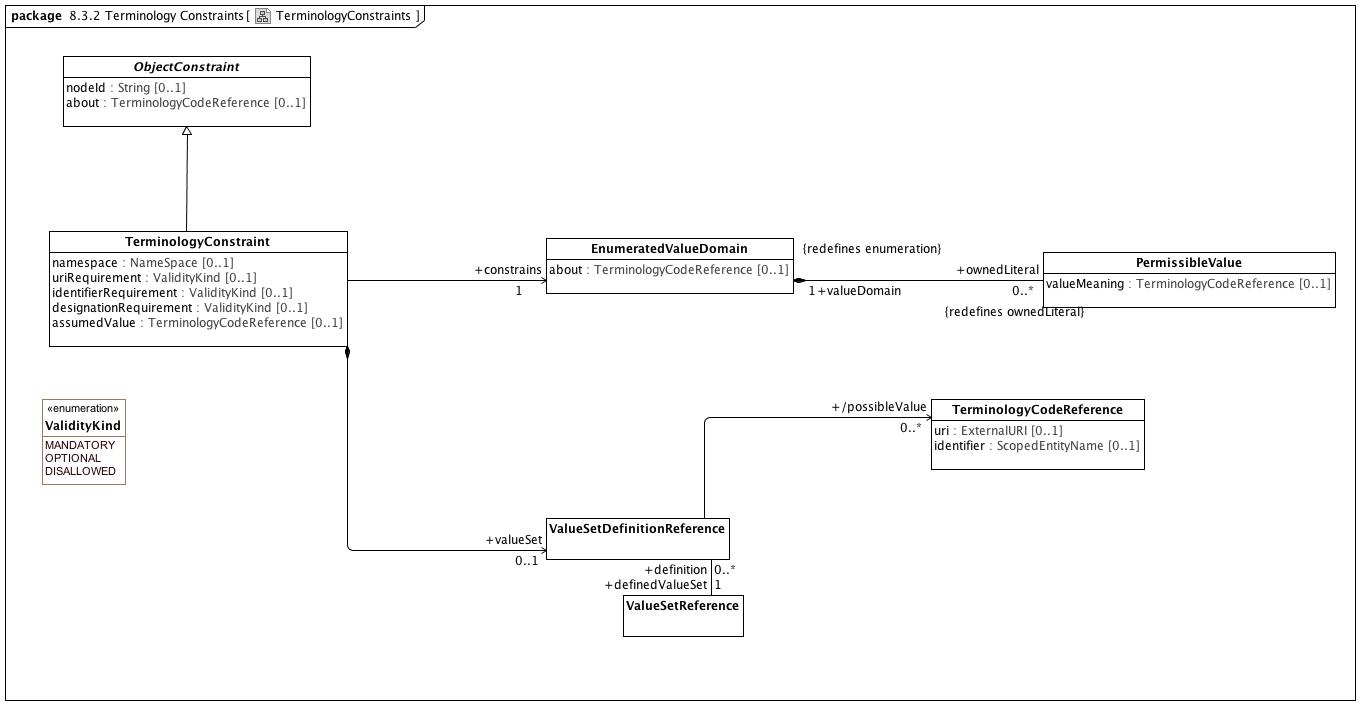
### Primitive Type Constraints



**Primitive Type Constraints**

[Interfaces = 0, Classes = 9, Enumerations = 0, DataTypes=0, StereoTypes=0]

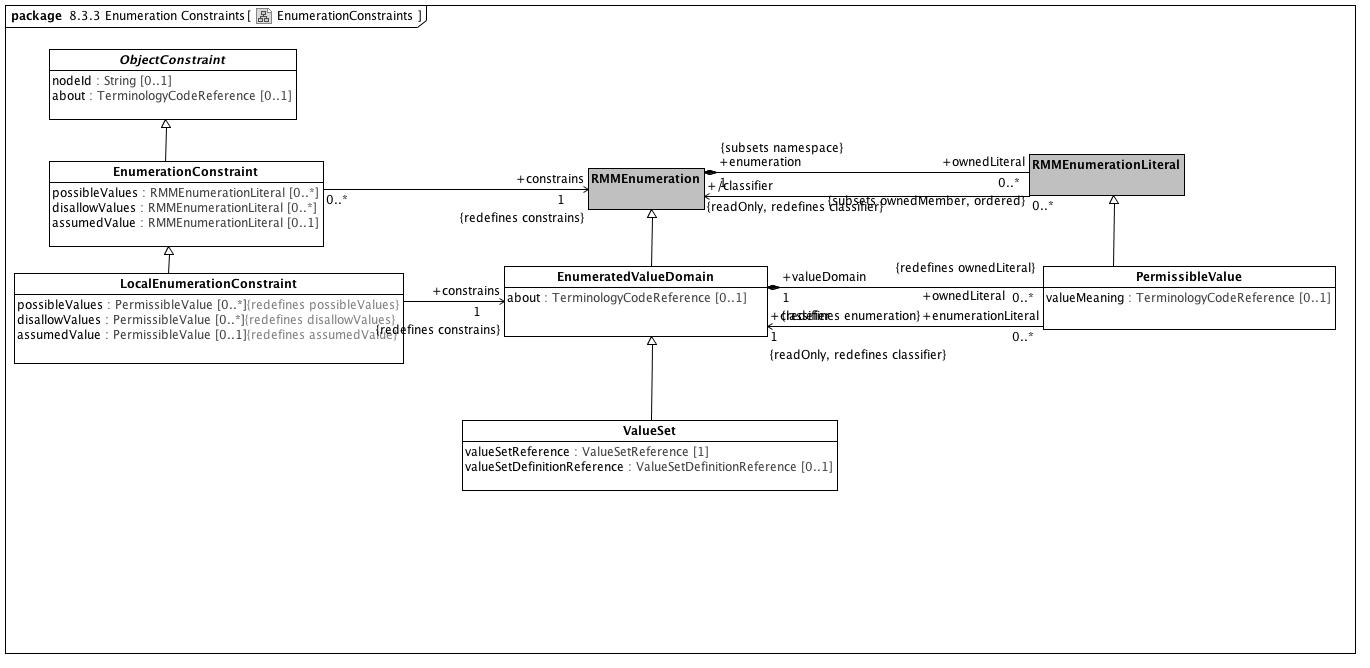
### Terminology Constraints



**TerminologyConstraints**

[Interfaces = 0, Classes = 7, Enumerations = 1, DataTypes=0, StereoTypes=0]

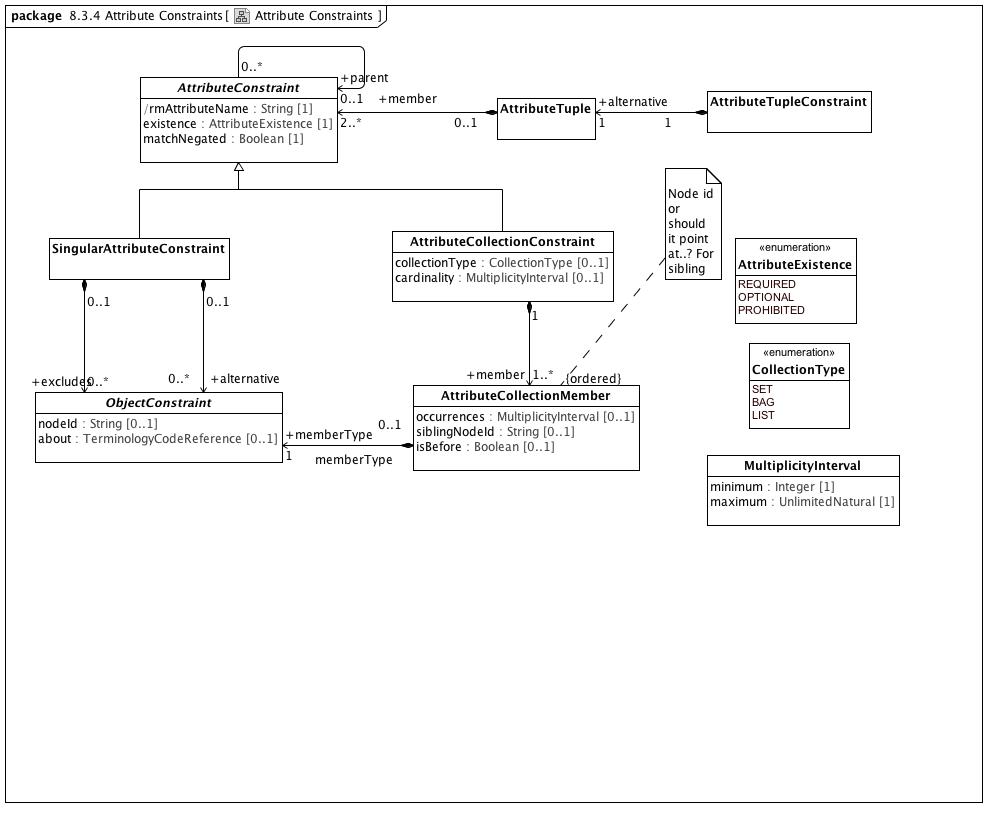
### Enumeration Constraints



**EnumerationConstraints**

[Interfaces = 0, Classes = 8, Enumerations = 0, DataTypes=0, StereoTypes=0]

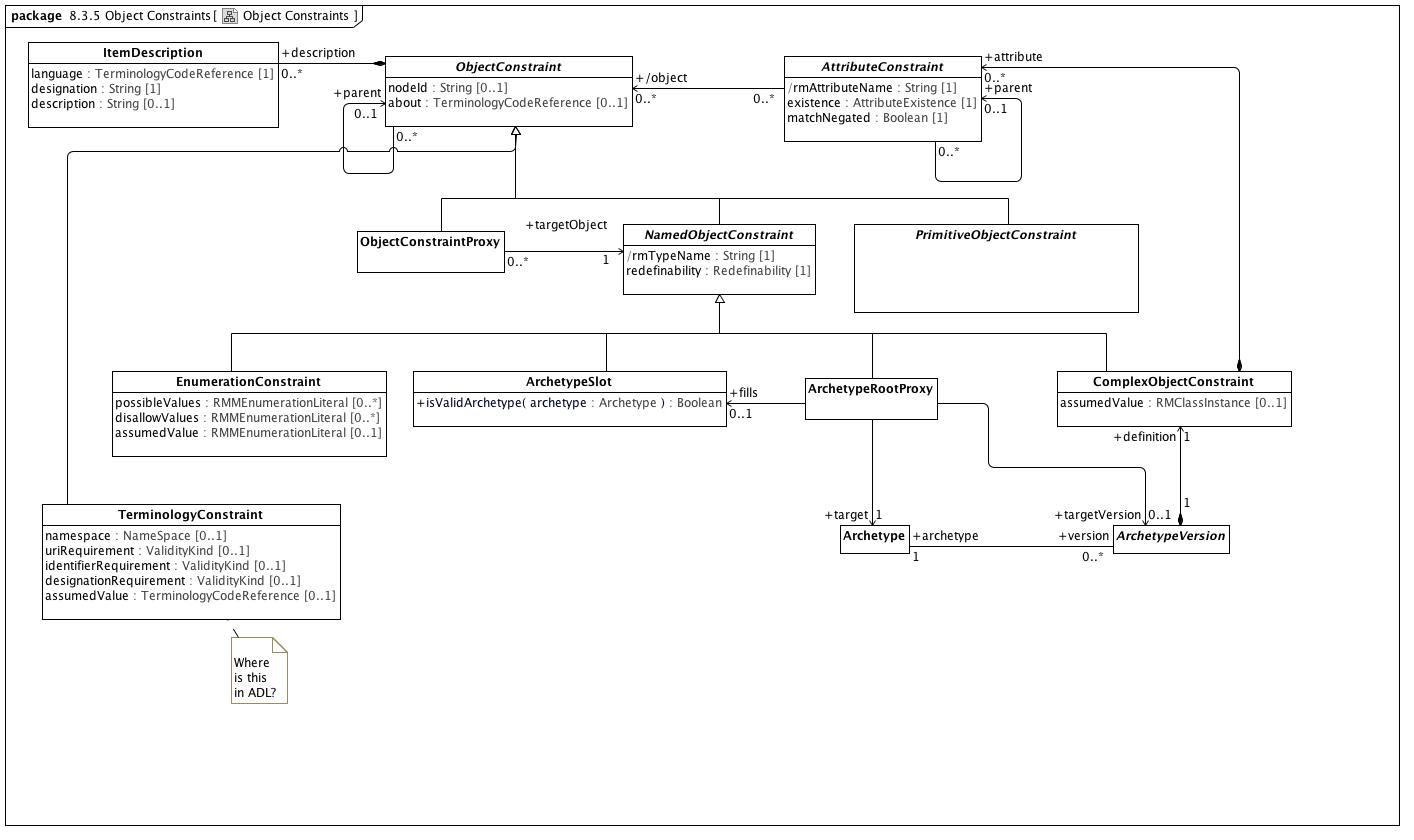
### Attribute Constraints



**Attribute Constraints**

[Interfaces = 0, Classes = 8, Enumerations = 2, DataTypes=0, StereoTypes=0]

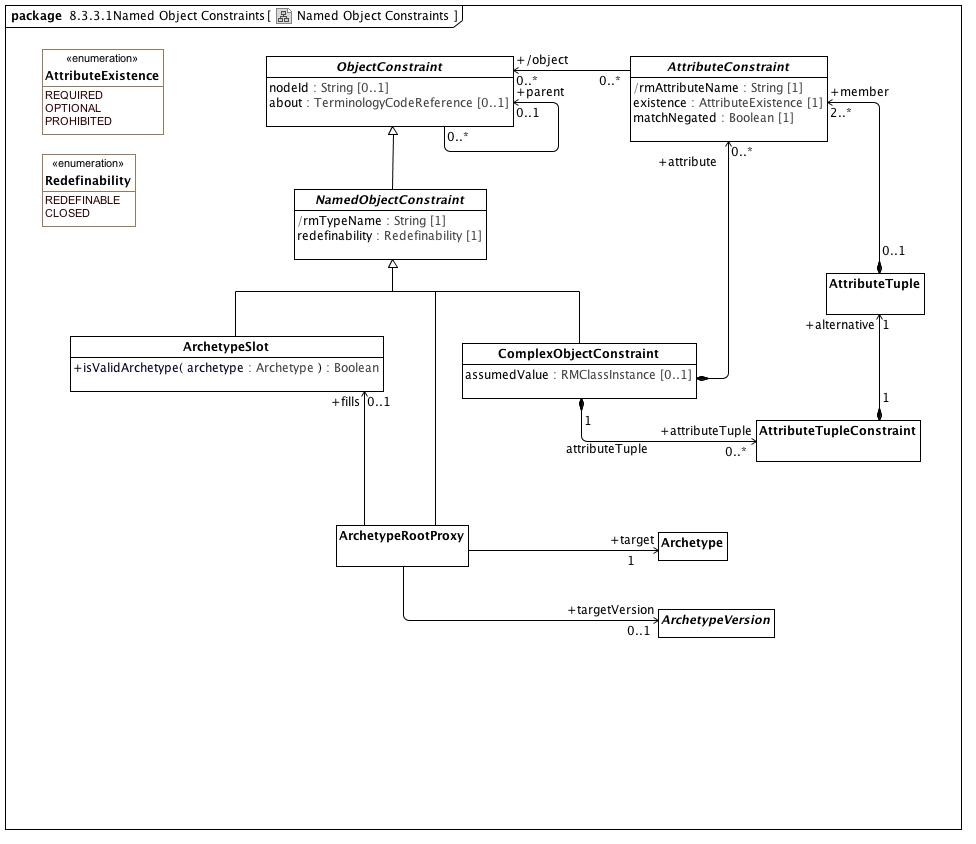
### Object Constraints



**Object Constraints**

[Interfaces = 0, Classes = 13, Enumerations = 0, DataTypes=0, StereoTypes=0]

#### Named Object Constraints

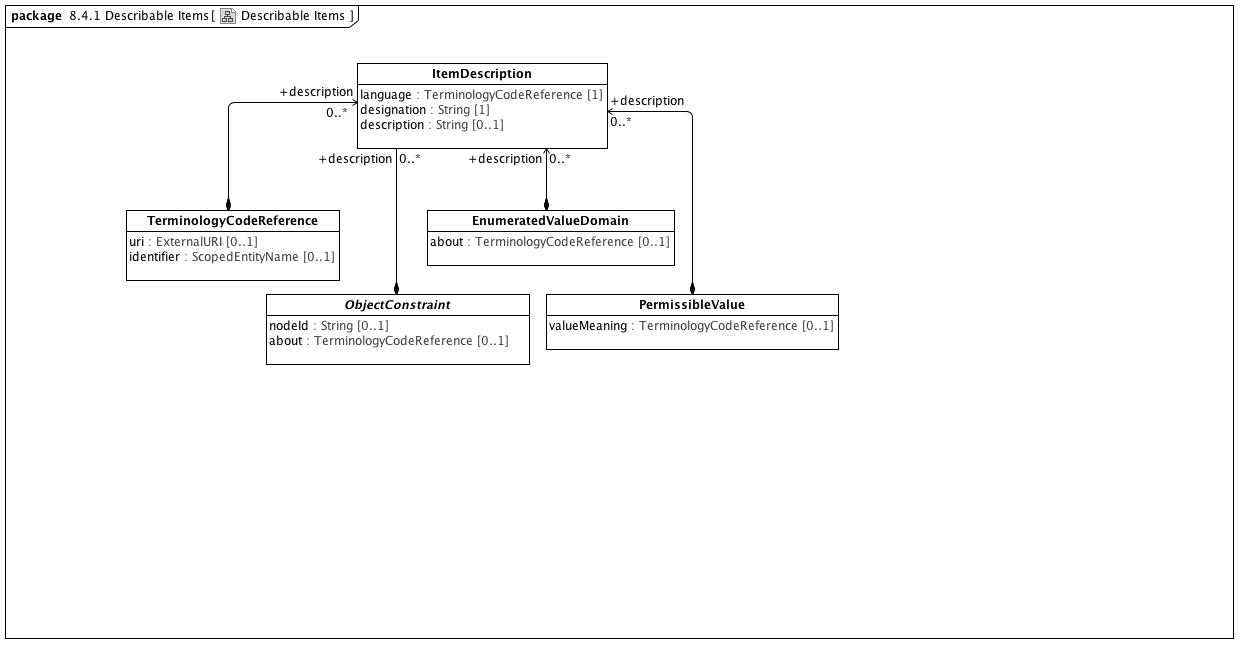


**Named Object Constraints**

[Interfaces = 0, Classes = 10, Enumerations = 2, DataTypes=0, StereoTypes=0]

## Terminology Object Model

### Describable Items



**Describable Items**

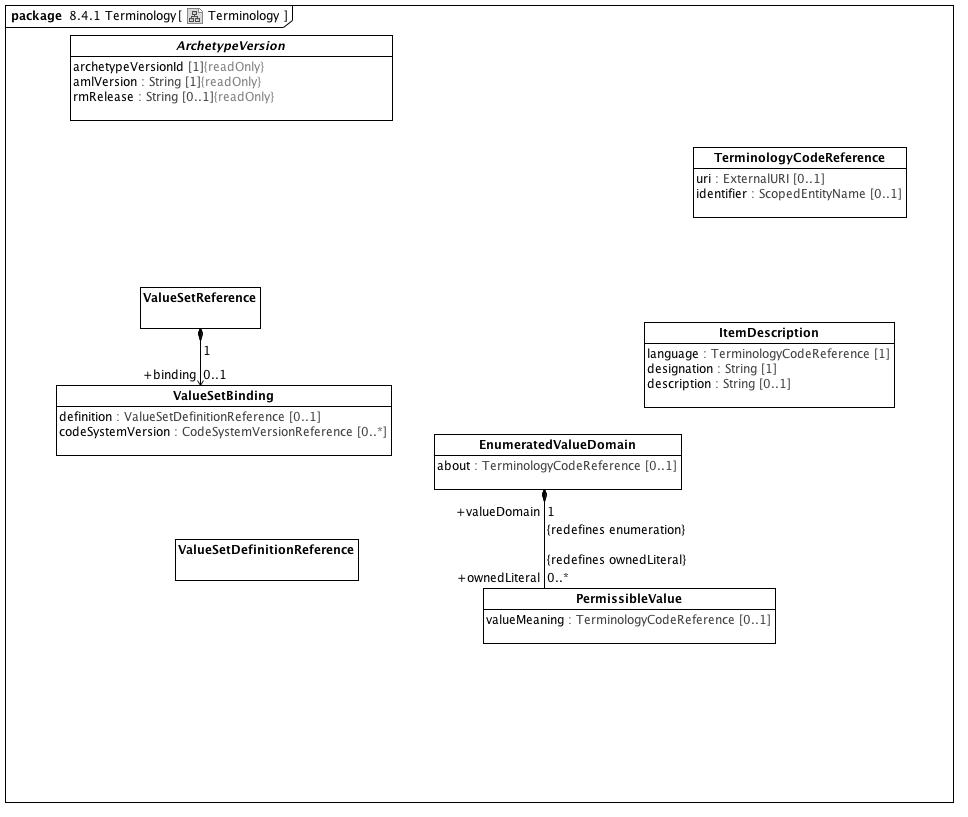
This diagram shows the classes that may be associated with one or more Item Descriptions. Each of these entries corresponds to an entry in the ADL 1.5 terminology section:

* ObjectConstraint - this represents the "id" codes in ADL -- the "term definitions" of object constraints
* EnumeratedValueDomain - this represents the "ac" codes in ADL -- the "term definitions" of internal and external value sets
* PermissibleValue - this represents the "at" codes in ADL -- the "term definition" of individual codes

In addition, terminology code references may be accompanied by one or more Item Descriptions that represent the intent of the code reference on a strictly informative basis.

[Interfaces = 0, Classes = 5, Enumerations = 0, DataTypes=0, StereoTypes=0]

### Terminology

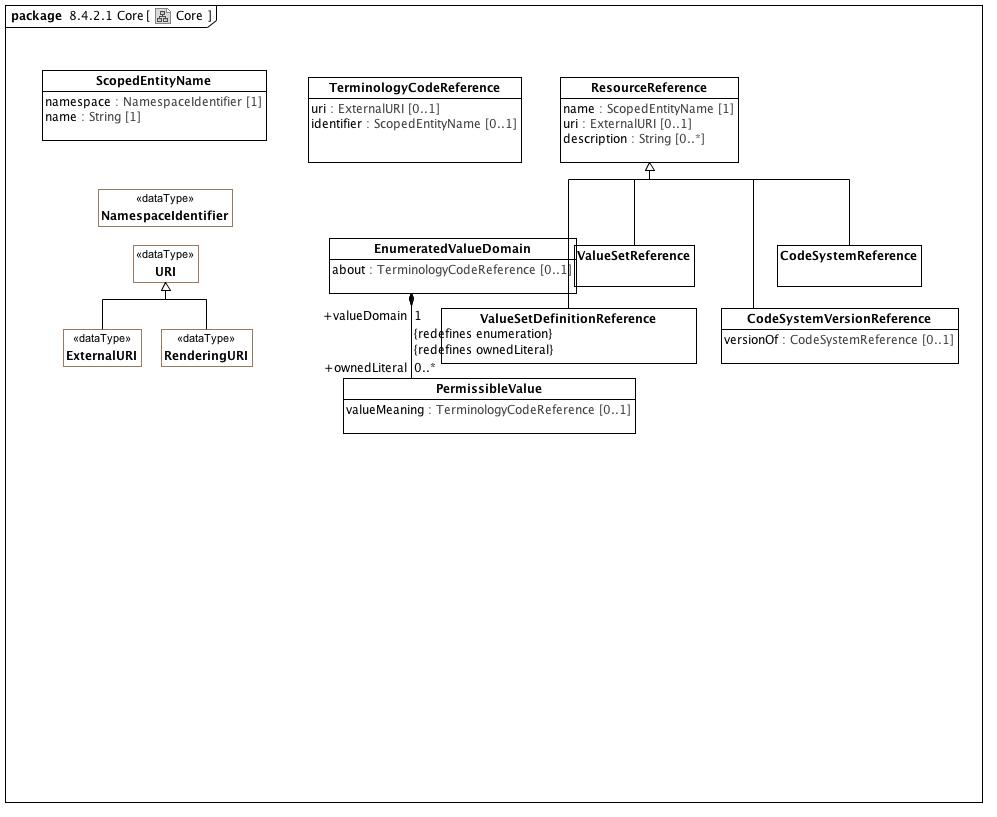


**Terminology**

[Interfaces = 0, Classes = 8, Enumerations = 0, DataTypes=0, StereoTypes=0]

### Terminology Services

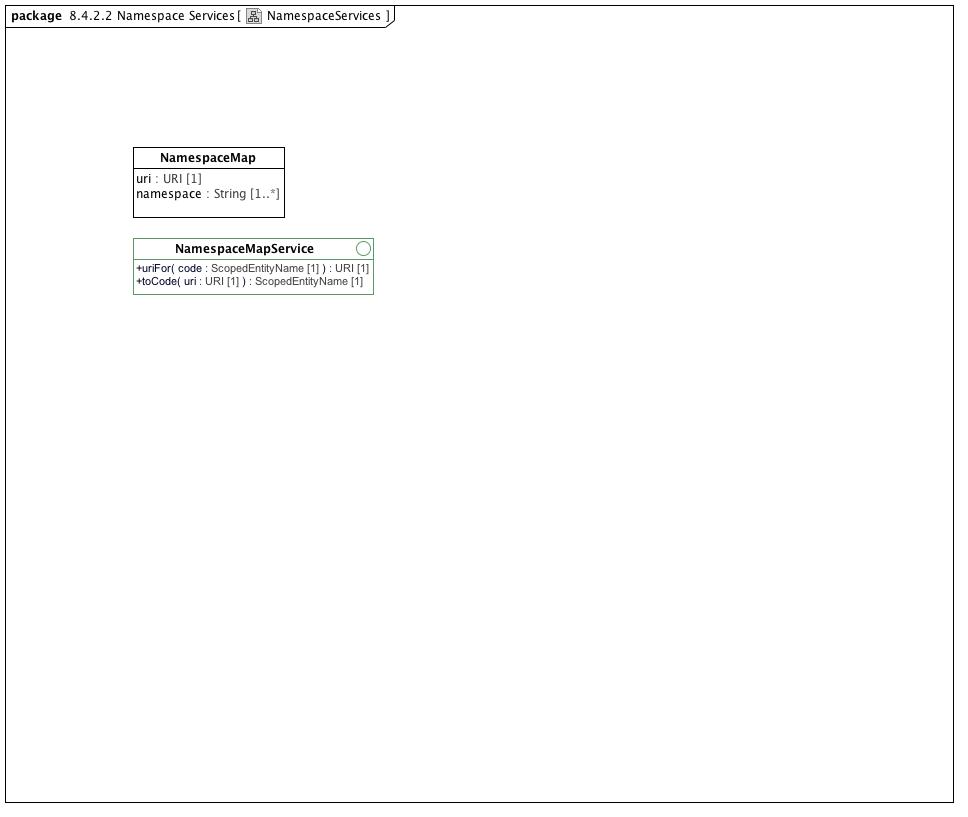
#### Core



**Core**

[Interfaces = 0, Classes = 9, Enumerations = 0, DataTypes=4, StereoTypes=0]

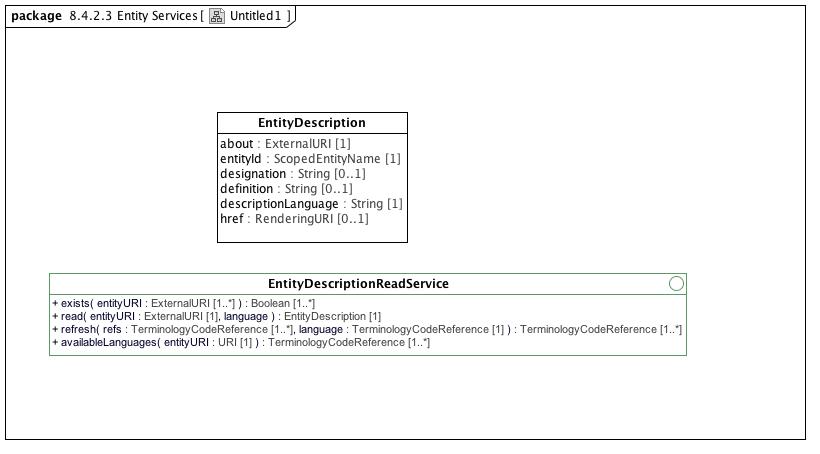
#### Namespace Services



**NamespaceServices**

[Interfaces = 1, Classes = 1, Enumerations = 0, DataTypes=0, StereoTypes=0]

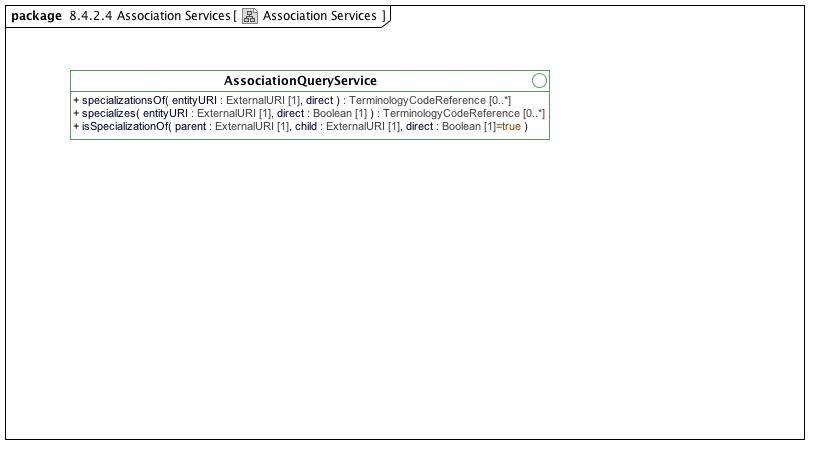
#### Entity Services



**Untitled1**

[Interfaces = 1, Classes = 1, Enumerations = 0, DataTypes=0, StereoTypes=0]

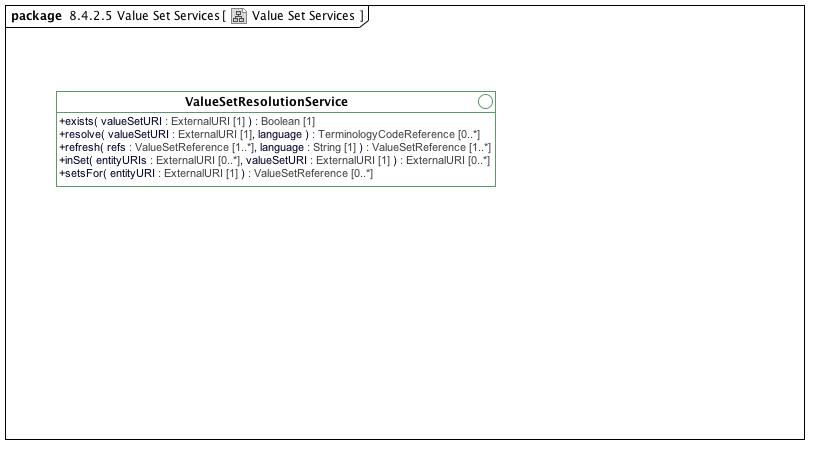
#### Association Services



**Association Services**

[Interfaces = 1, Classes = 0, Enumerations = 0, DataTypes=0, StereoTypes=0]

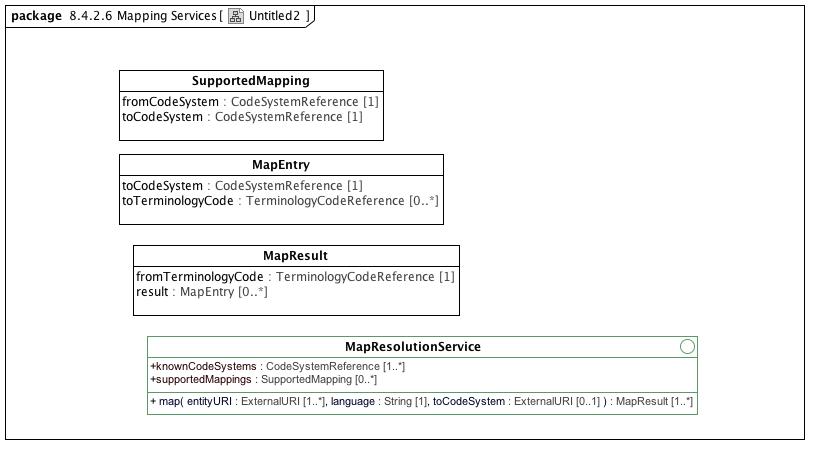
#### Value Set Services



**Value Set Services**

[Interfaces = 1, Classes = 0, Enumerations = 0, DataTypes=0, StereoTypes=0]

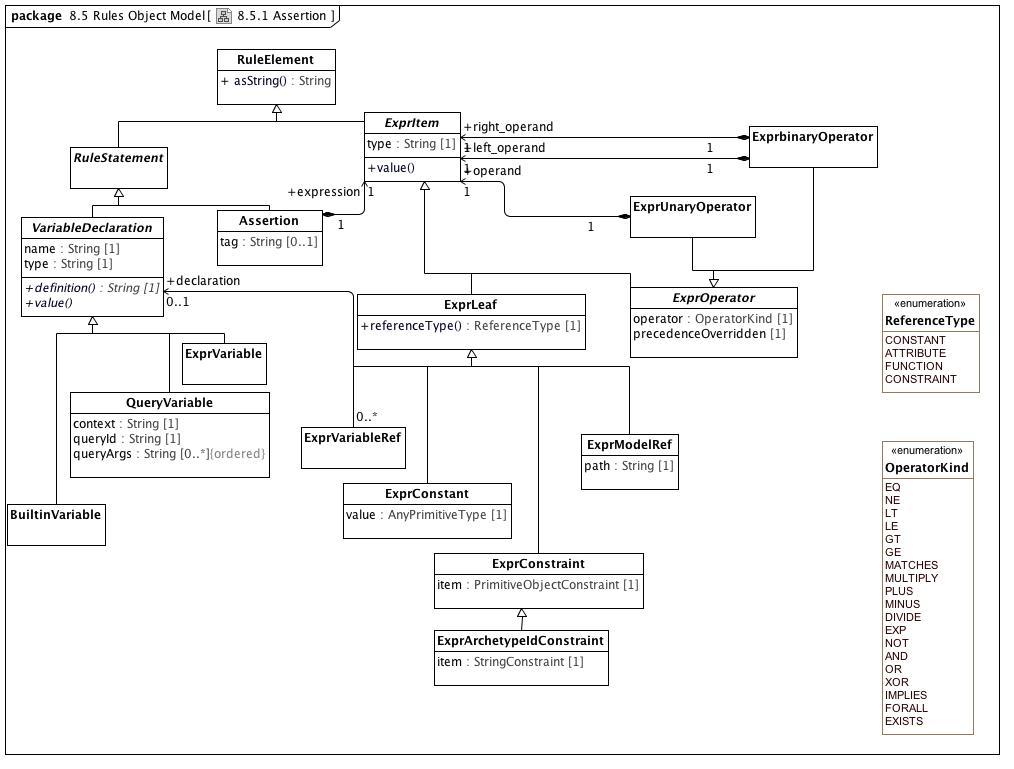
#### Mapping Services



**Untitled2**

[Interfaces = 1, Classes = 3, Enumerations = 0, DataTypes=0, StereoTypes=0]

## Rules Object Model



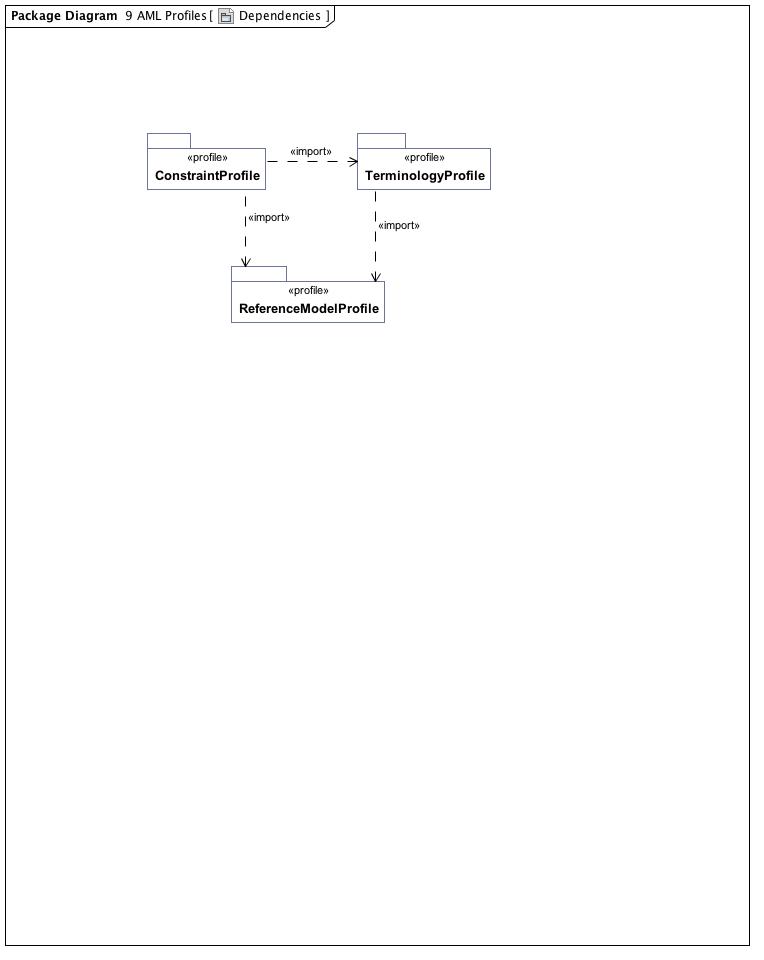
**8.5.1 Assertion**

[Interfaces = 0, Classes = 17, Enumerations = 2, DataTypes=0, StereoTypes=0]

## Metadata Object Model

# AML Profiles

Introduction to the AML Profiles section



**Dependencies**

[Interfaces = 0, Classes = 0, Enumerations = 0, DataTypes=0, StereoTypes=0]

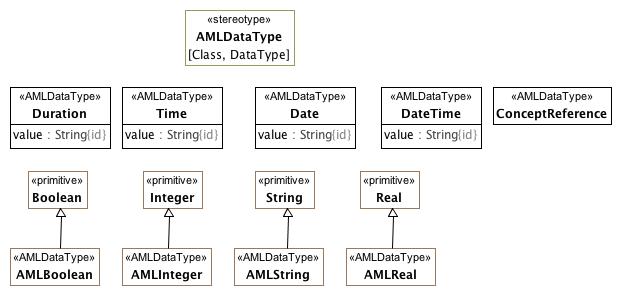
## Reference Model Profile

Introduction to the reference model profile section



**9.1.1 ConcreteInstance Specification**

[Interfaces = 0, Classes = 0, Enumerations = 0, DataTypes=0, StereoTypes=1]



**9.1.2 Implementable Data Types**

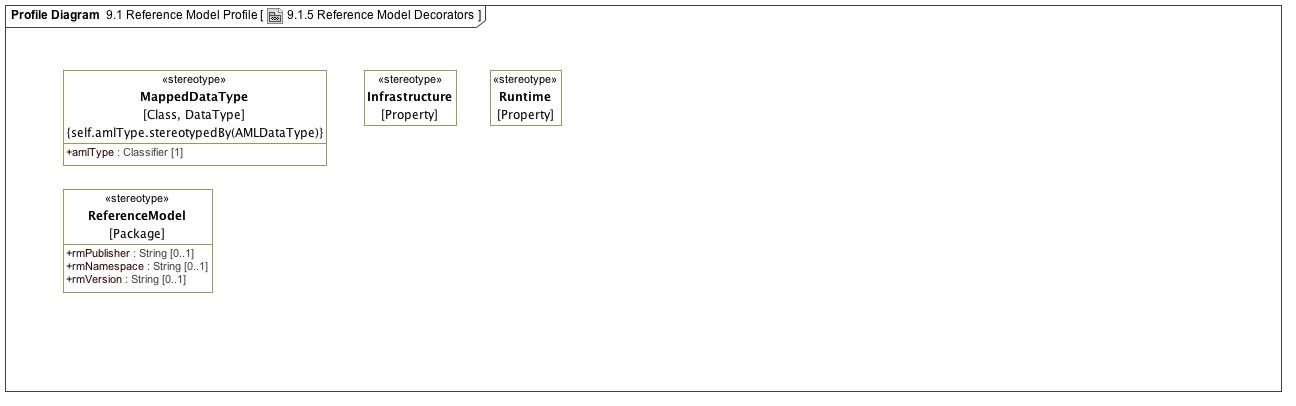
The AMLDataType stereotype serves two purposes:

1. To identify the set of atomic types whose possible values can be constrained in the AML Constraint Profile
2. To identify the set of types whose value will be treated as "data types" from the AML perspective.

The AMLDataType stereotype can extend both Class and DataType elements. The target reference model may choose to represent some or all of the AML DataTypes in a different fashion. A reference model may define its own String DataType rather than using the UML Native String type directly. Similarly, it may choose to represent a Date as a complex object consisting of year, month, day, granularity, zone, etc.

One of the tasks for a reference model implementor is to create maps from the appropriate AML Data Types and the corresponding reference model types.

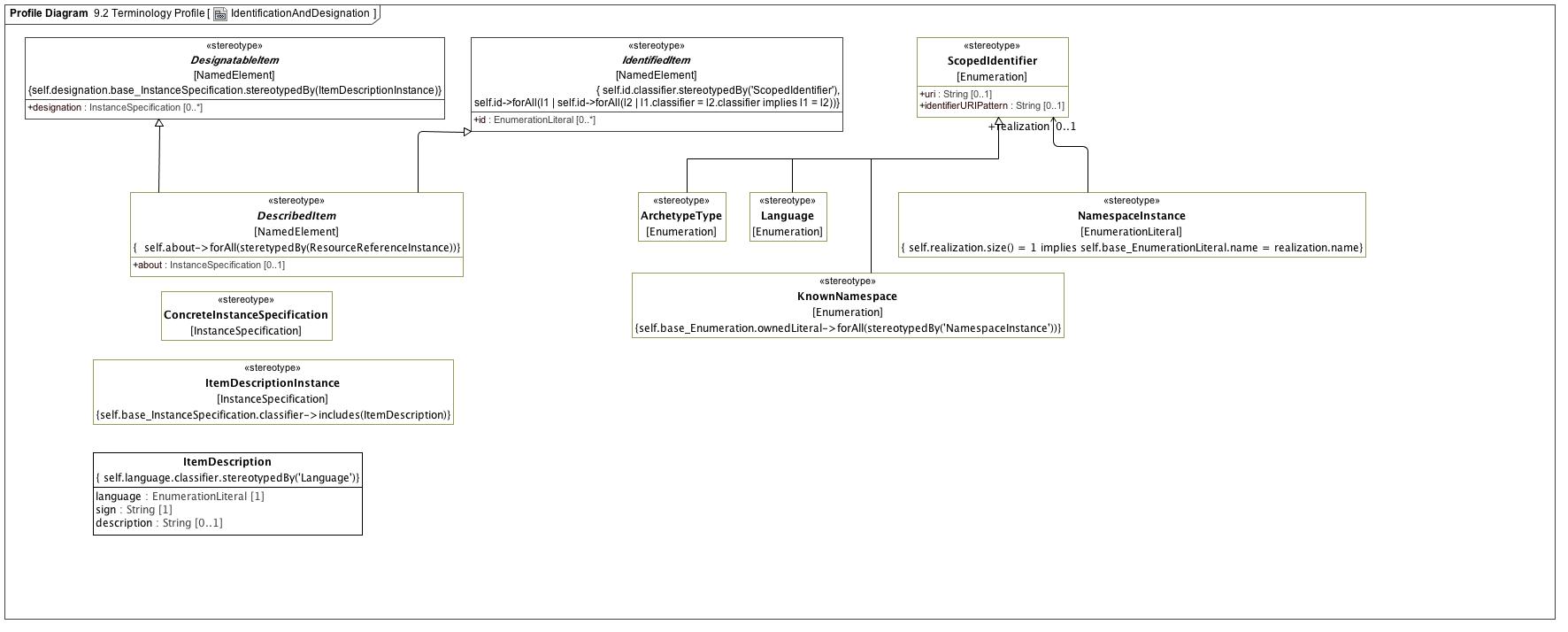
[Interfaces = 0, Classes = 5, Enumerations = 0, DataTypes=0, StereoTypes=1]



**9.1.5 Reference Model Decorators**

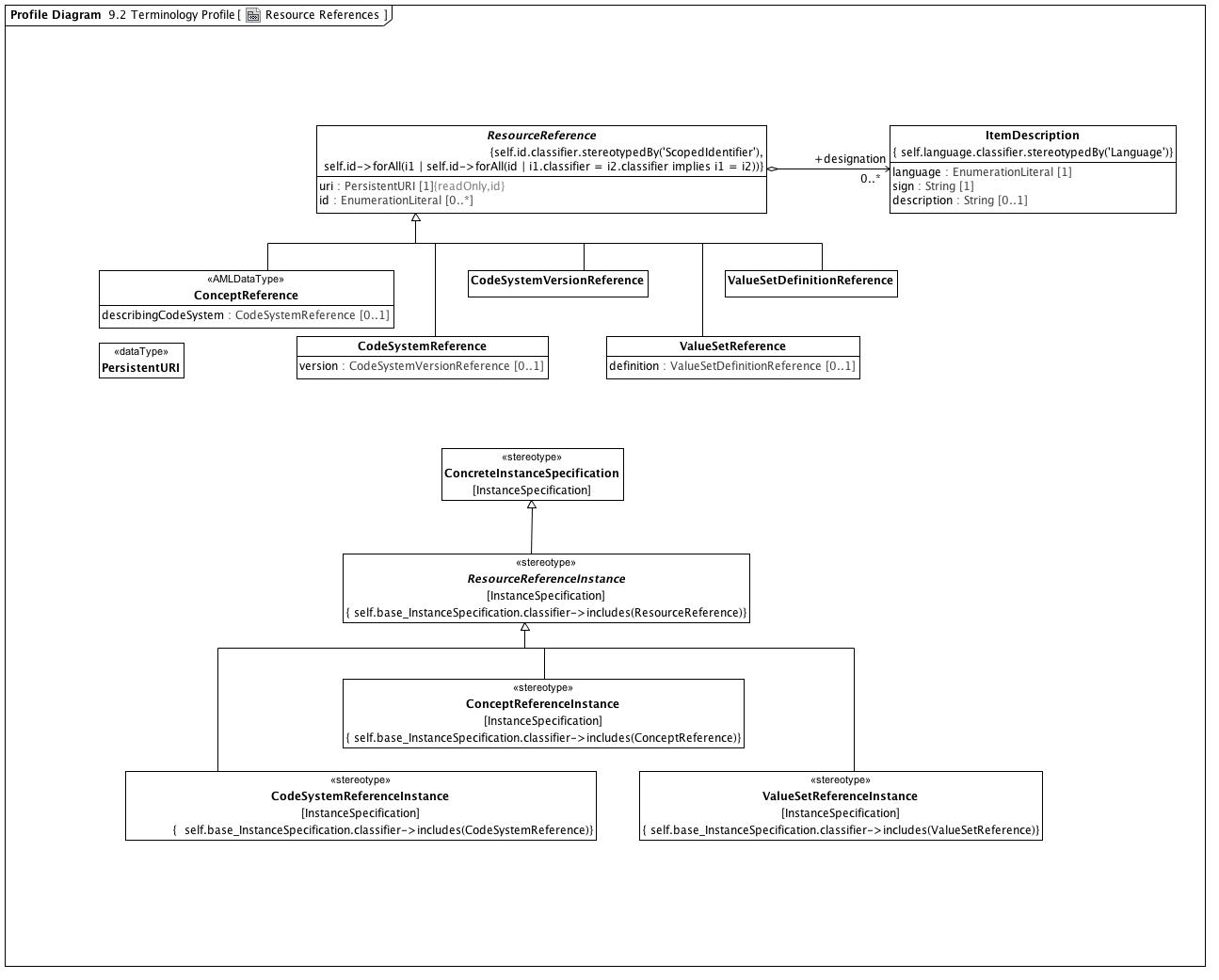
[Interfaces = 0, Classes = 0, Enumerations = 0, DataTypes=0, StereoTypes=4]

## Terminology Profile



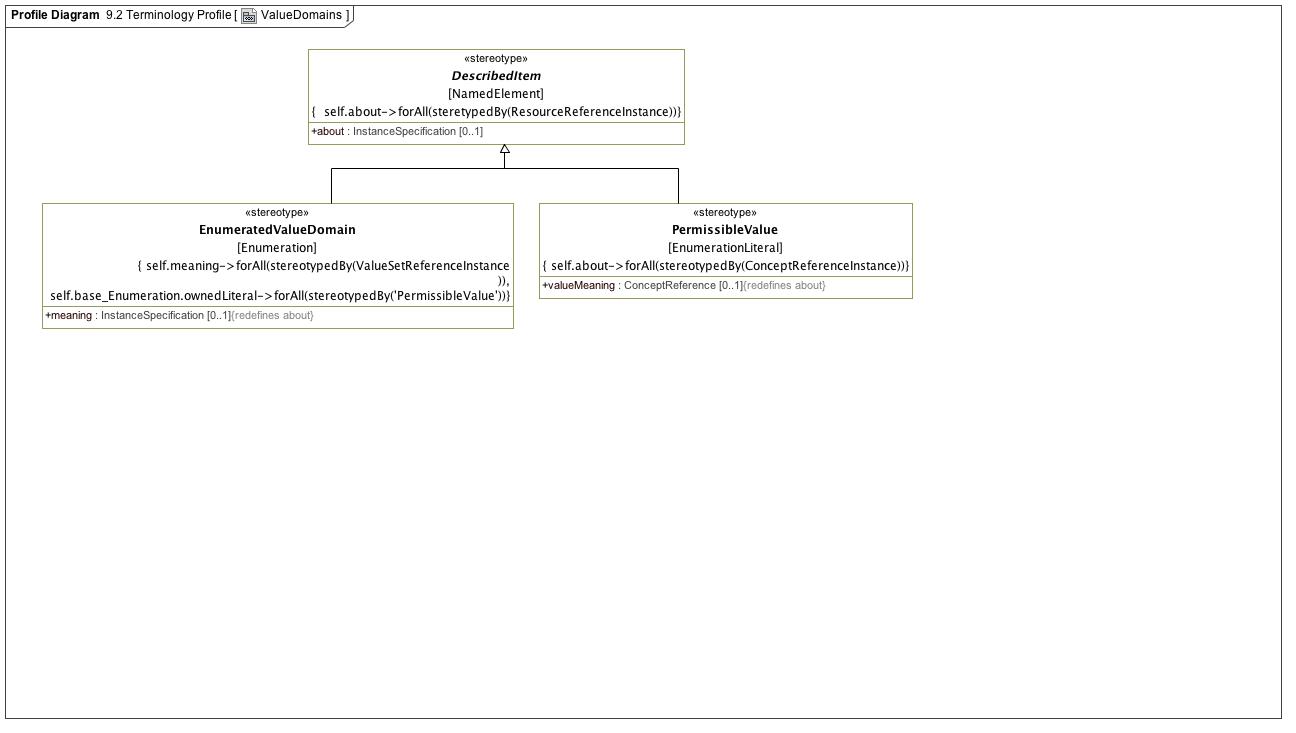
**IdentificationAndDesignation**

[Interfaces = 0, Classes = 1, Enumerations = 0, DataTypes=0, StereoTypes=10]



**Resource References**

[Interfaces = 0, Classes = 7, Enumerations = 0, DataTypes=1, StereoTypes=5]

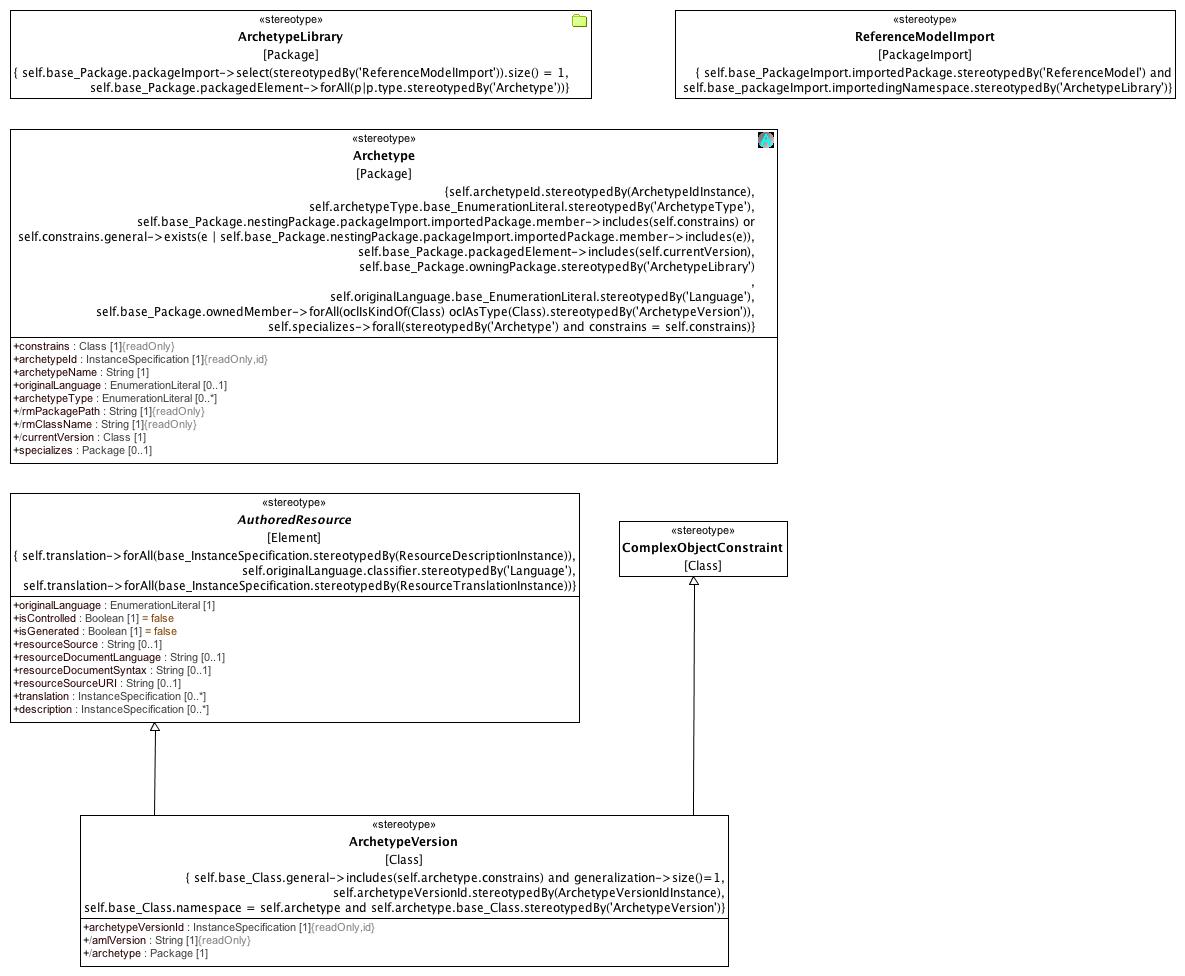


**ValueDomains**

[Interfaces = 0, Classes = 0, Enumerations = 0, DataTypes=0, StereoTypes=3]

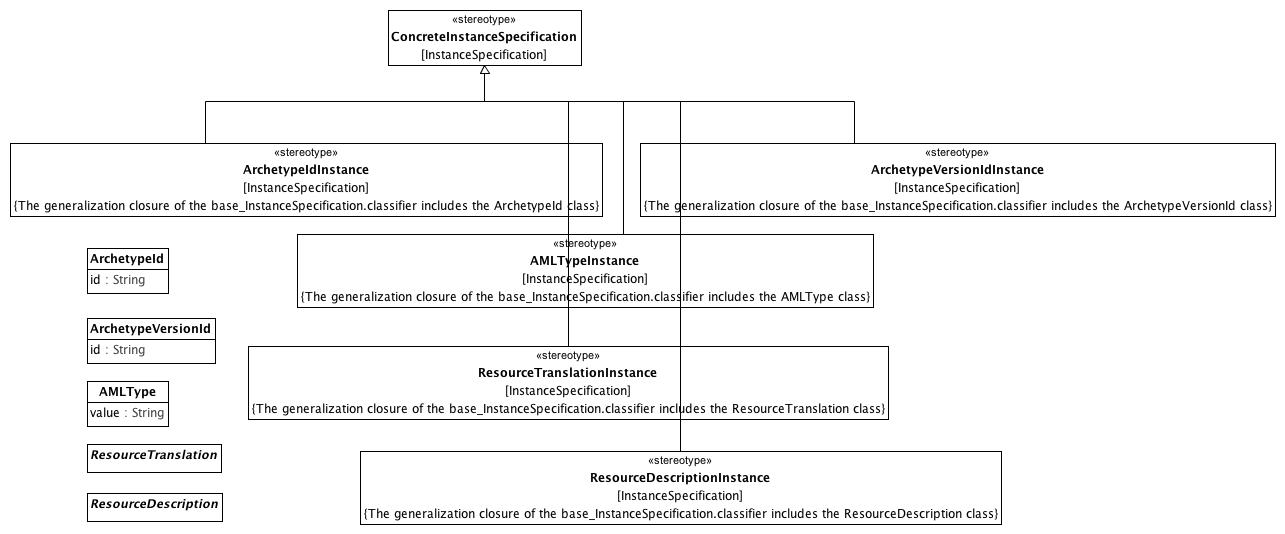
## Constraint Profile

### Archetypes



**9.3.1.1 Archetypes**

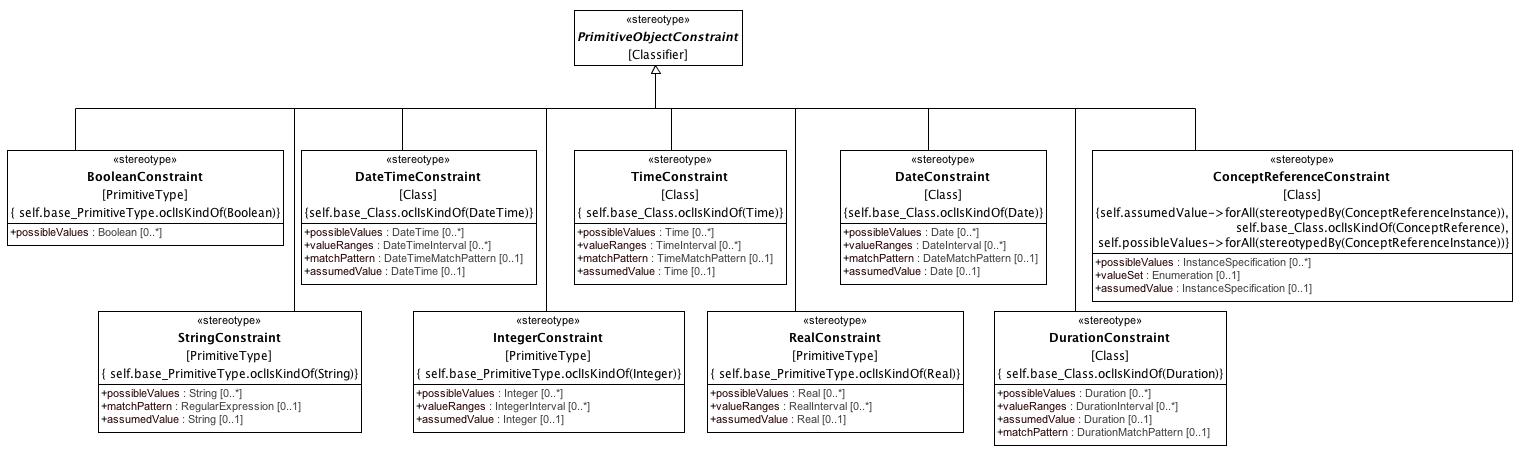
[Interfaces = 0, Classes = 0, Enumerations = 0, DataTypes=0, StereoTypes=6]



**9.3.1.2 Archetype Metadata**

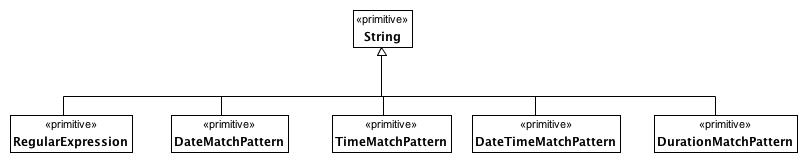
[Interfaces = 0, Classes = 5, Enumerations = 0, DataTypes=0, StereoTypes=6]

### Data Type Constraints



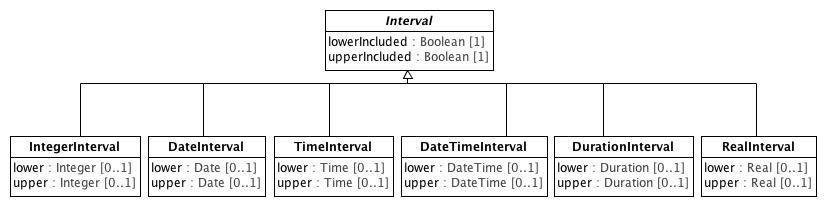
**9.3.2.1 Primitive Type Constraints**

[Interfaces = 0, Classes = 0, Enumerations = 0, DataTypes=0, StereoTypes=10]



**9.3.2.2 Date and Time Match Types**

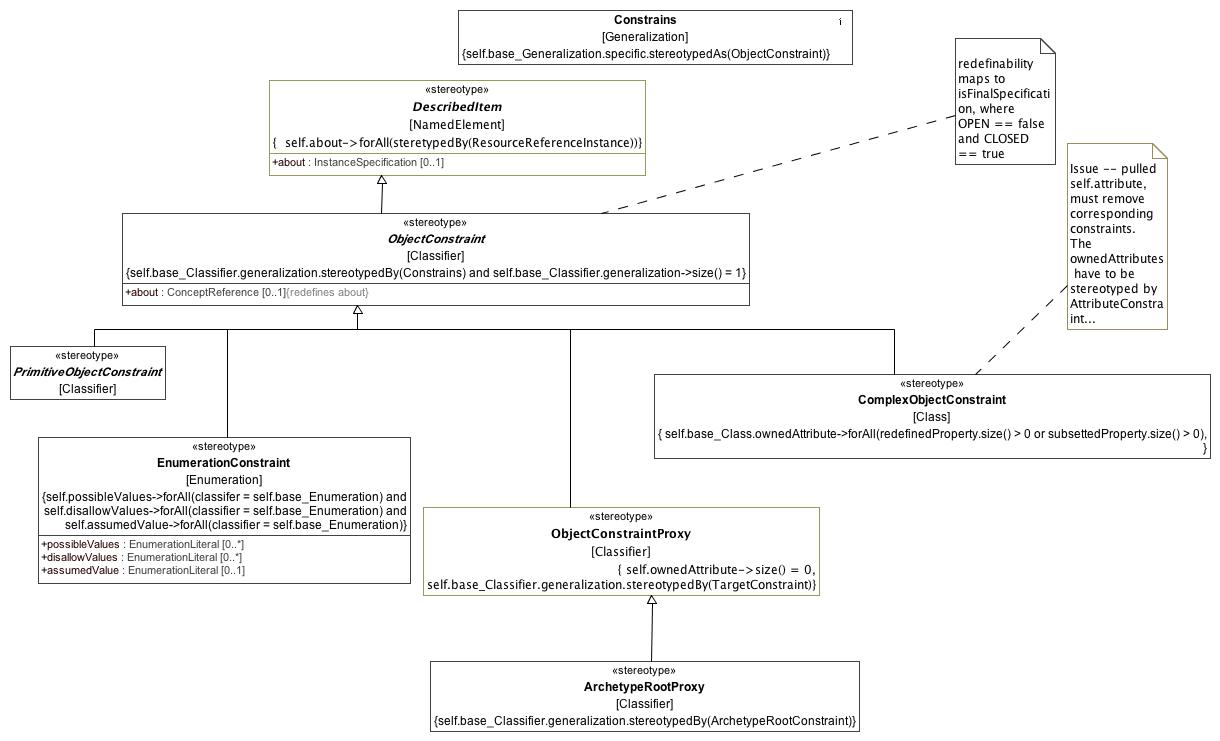
[Interfaces = 0, Classes = 0, Enumerations = 0, DataTypes=0, StereoTypes=0]



**9.3.2.3 Intervals**

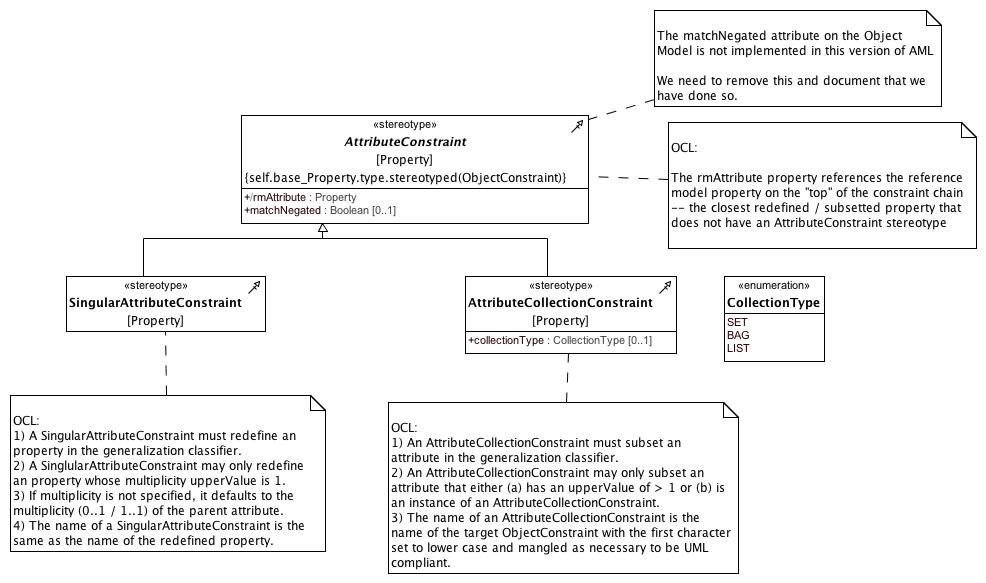
[Interfaces = 0, Classes = 7, Enumerations = 0, DataTypes=0, StereoTypes=0]

### Object and Property Constraints



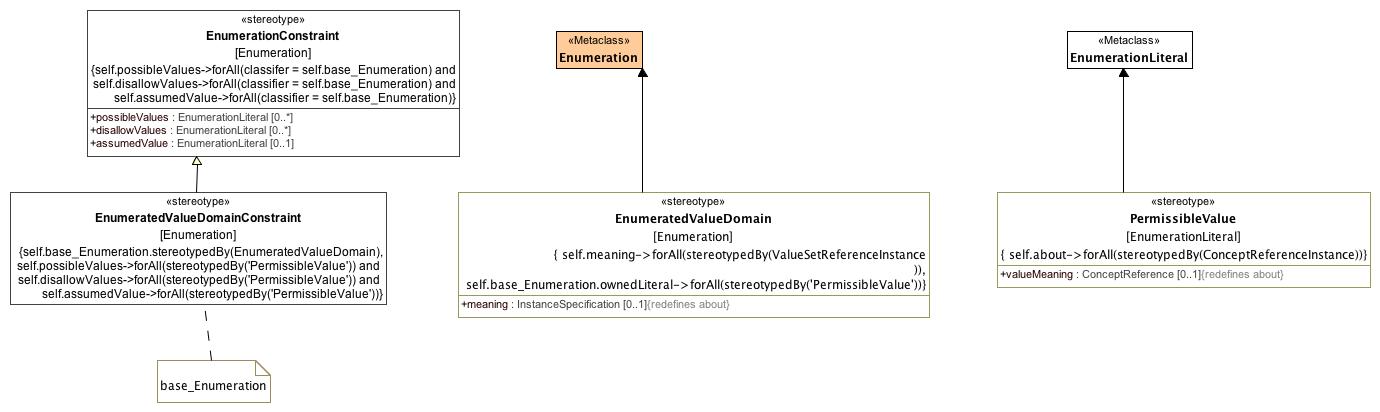
**9.3.3.1 Constraints**

[Interfaces = 0, Classes = 0, Enumerations = 0, DataTypes=0, StereoTypes=8]



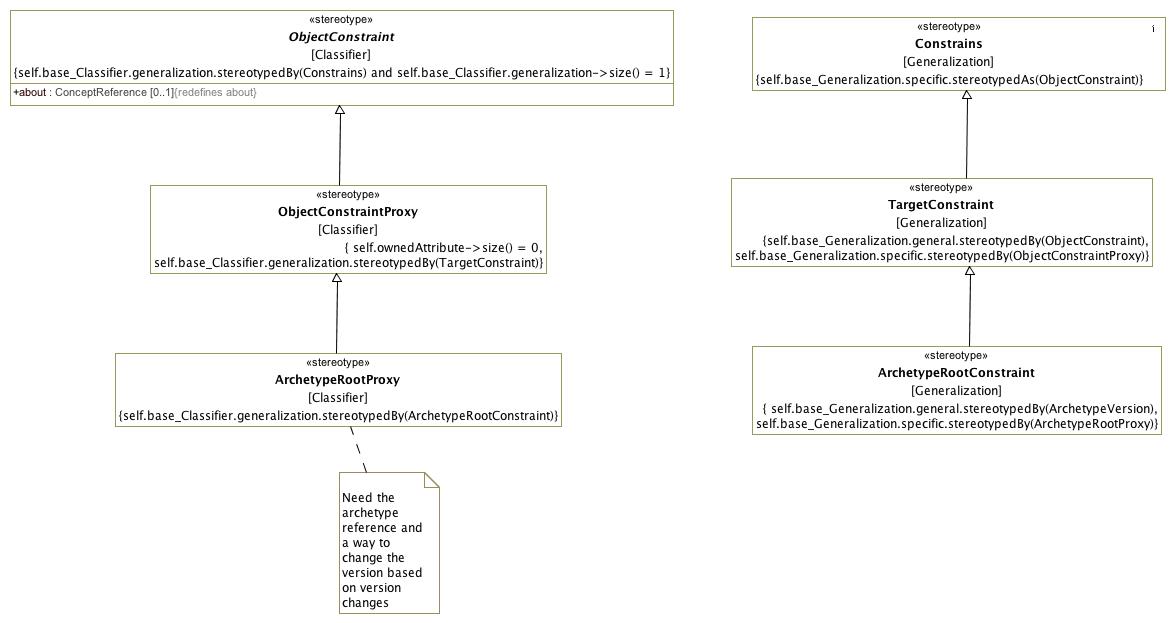
**9.3.3.2 Attribute Constraints**

[Interfaces = 0, Classes = 0, Enumerations = 1, DataTypes=0, StereoTypes=3]



**9.3.3.3 Enumeration Constraints**

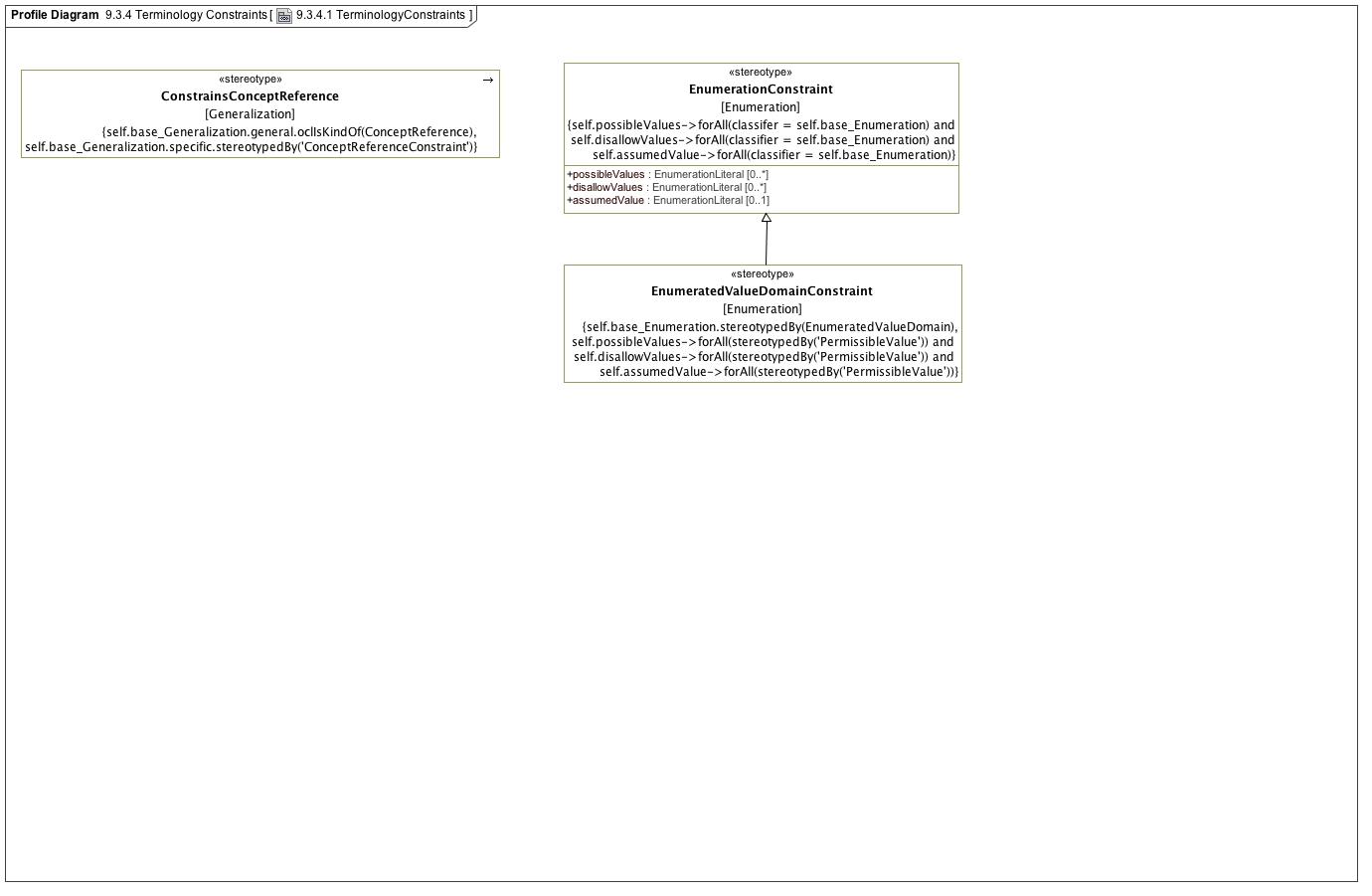
[Interfaces = 0, Classes = 2, Enumerations = 0, DataTypes=0, StereoTypes=4]



**9.3.3.4 Constraint Proxies**

[Interfaces = 0, Classes = 0, Enumerations = 0, DataTypes=0, StereoTypes=6]

### Terminology Constraints



**9.3.4.1 TerminologyConstraints**

[Interfaces = 0, Classes = 0, Enumerations = 0, DataTypes=0, StereoTypes=3]