|  |
| --- |
|  |
| |  |  |  | | --- | --- | --- | | 75 | Date: | April 2015 | |
| Archetype Modeling Language (AML)    Version: 1.0  **OMG Document Number: health/2014-10-01**  **Standard document URL: http://www.omg.org/spec/AML/1.0**  Original File: N/A |

Copyright © 2014, Mayo Clinic  
Copyright © 2014, Object Management Group, Inc.  
Copyright © 2014, Visumpoint, LLC

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'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 logos™, XMI Logo™, CWM™, CWM Logo™, IIOP™ , IMM™ , 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>.)

Table of Contents

1. Scope 6

1.1. Archetype Modeling Language (AML) Background 6

1.2. AML Intended Users 7

1.3. AML Profiles 7

2. Conformance 8

2.1. Conformance Points 8

2.2. AML Reference Model Profile 8

2.3. AML Terminology Binding Profile 8

2.4. AML Constraint Model Profile 8

3. Normative References 8

4. Terms and Definitions 9

5. Symbols 13

5.1. Graphical Symbols 13

5.2. Abbreviations 13

6. Additional Information 14

6.1. Changes to Adopted OMG Specifications 14

6.2. Acknowledgements 14

7. The AOM and the AML Metamodel 14

8. specification 14

8.1. Introduction 14

8.2. Scope 15

8.3. Profiles 15

8.3.1. Dependencies 15

8.3.2. ReferenceModelProfile [Profile] 15

8.3.3. Terminology Binding Profile 15

8.3.3.1. Terminology Binding Profile 15

8.3.4. ConstraintProfile [Profile] 31

Appendix A: AML MetaModel 32

**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

## Archetype Modeling Language (AML) Background

This specification defines the Archetype Modeling Language (AML). The AML defines a standard means for modeling Archetype Models (AMs) to support the representation of Clinical Information Modeling Initiative (CIMI) artifacts using modeling profiles as defined in the UML. Archetype Models are Platform Independent Models (PIMs) and are developed as a set of constraints on a specific Reference Model (RM).

The CIMI RM is the underlying RM on which CIMI’s clinical information models are defined. The reference model defines a rigorous and stable set of modeling patterns that include a set of structural patterns, complex data types, and demographic classes. All CIMI clinical models will be defined by constraining the CIMI reference model. Each instance of a CIMI Clinical Model will be a constrained instance of the CIMI reference model conforming to the constraints defined by the associated clinical model.

The motivation for including a reference model in the CIMI clinical modeling architecture is to provide a consistent computational framework upon which model authoring and translation tools can be based. The reference model is the ‘common language’ used to describe all clinical models. It provides a single information model that can be used to represent instances of all clinical models and upon which further constraints can be applied to represent the specific information requirements of all clinical model. This information model represents the core artifact implemented in software; it provides the physical structure of the clinical models and its example instances. Existing implementation experience has shown this increases the computational capabilities of the resulting modeling and translation tools.

Development of the AML specification was guided by:

1. The need for a means to accurately and usefully represent AMs in accordance with the openEHR Foundation’s Archetype Definition Language (ADL) and Archetype Object Model (AOM) version 2.0 specifications;
2. Compatibility with the Object Management Group (OMG) *Common Terminology Service 2 (CTS2)* specification; and
3. Where possible, being informed by and faithful to the *ISO/IEC 11179, Information Technology, -- Metadata registries*, specification.

In the AML RFP, the version of the openEHR Foundation’s ADL and AOM specifications cited for coverage by the OMG AML specification was version 1.5. In the process of producing the AML specification, however, a number of inconsistencies were discovered in the openEHR specifications, as well as opportunities for improvements. These were reported to the openEHR Foundation. In response, the openEHR Foundation revised the specifications. This resulted in a set of changes to the specifications that were not backward compatible with version 1.5. As a consequence, the revised specifications were released as version 2.0, subsuming the requirements found in version 1.5, now made consistent in version 2.0, and forming the updated requirements basis for AML coverage.

## AML Intended Users

The AML is primarily intended to support two clinical modeling communities of users:

* Those having subject matter expertise regarding clinical model domains and currently using ADL-based tools to develop such models, and
* Those familiar with modeling using the UML, though not necessarily familiar with clinical modeling domains or current methods employed to represent them.

Clause 7 of this specification, *AML Meta Model*, provides an informational meta model of the openEHR AOM as an aid to bridging between these communities.

While the AML specification targets CIMI clinical modeling practitioners, the modeling approach defined in the profiles is intended to be generalizable for use with other reference models and application in other domain areas.

## AML Profiles

The AML is specified by three UML profilescollectively meeting the requirements of archetype modeling. These are the:

* *Reference Model Profile (RMP)*: Enables the specification of reference models upon which archetypes can be based;
* *Constraint Model Profile (CMP)*: Supports the specification of constraints on a given reference model to enable the development of archetypes including Clinical Information Models (CIMs); and
* *Terminology Binding Profile (TBP)*: Supports the binding of information models to terminology. Terminology bindings include:
  1. *Value Bindings*: Support linking the data model to value domains that restrict the valid value of an attribute to a set of values corresponding to a set of meanings recorded in an external terminology;
  2. *Semantic Bindings:* Define the meaning of model elements using concepts in an external terminology; and
  3. *Constraint Bindings:* Specify constraints on the information model using concepts and relationships defined in an external terminology.

This set of UML profiles enables the specification of CIMI clinical model content (using the CIMI Reference Model) and the generation of CIMI clinical model artifacts, such as ones represented by the openEHR Foundation’s ADL. (The ADL is a serialization of the openEHR Foundation’s AOM.) While the transformation of AML models to an instance of the AOM was an optional requirement for the AML specification, the AML profile supports the representation of sufficient information in an AM to enable such a transformation.

# Conformance

## Conformance Points

This specification defines the following conformance points (also referred to as conformance targets):

* AML Reference Model Profile
* AML Terminology Binding Profile
* AML Constraint Model Profile

## AML Reference Model Profile

Sub clause 8.1 of this specification defines the AML Reference Model Profile.

## AML Terminology Binding Profile

Sub clause 8.2 of this specification defines the AML Terminology Binding Profile. The Terminology Binding Profile imports the Reference Model Profile.

## AML Constraint Model Profile

Sub clause 8.3 of this specification defines the AML Constraint Model Profile. The Constraint Model Profile imports both the Reference Model Profile and Terminology Binding Profile.

# Normative References

The following normative documents contain provisions that, through reference in this text, constitute provisions of this specification. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply.

[ADL] openEHR *Archetype Definition Language: ADL2*, <http://www.openehr.org/releases/trunk/architecture/am/adl2.pdf>

[AOM] *openEHR Archetype Object Model* (AOM), <http://www.openehr.org/releases/trunk/architecture/am/aom2.pdf>

[AOMT] openEHR *openEHR Templates* (supersedes *openEHR Archetype Templates*), <http://www.openehr.org/releases/trunk/architecture/am/tom.pdf>

[ARCH] *openEHR Archetypes: Constraint-based Domain Models for Future-proof Information Systems*, <http://www.openehr.org/publications/archetypes/archetypes_beale_oopsla_2002.pdf>

[CIMI] CIMI Reference Model Requirements, <http://informatics.mayo.edu/CIMI/index.php/CIMI_Reference_Model_Requirements>

[CTS2] OMG *Common Terminology Service 2 (CTS2)*, [http://www.omg.org/spec/CTS2/1.1/](http://www.omg.org/spec/CTS2/1.1/" \o "http://www.omg.org/spec/CTS2/1.1/)

[HLV7v3] *HL7 Version 3 Standard: Core Principles and Properties of Version 3 Models*, <http://www.hl7.org/implement/standards/product_brief.cfm?product_id=58>

[MDMI] OMG *Model Driven Message Interoperability (MDMI), Version 1.0*, <http://www.omg.org/spec/MDMI/1.0/>

[MDR] *ISO/IEC 11179, Information Technology, -- Metadata registries*, [http://metadata-standards.org/11179/](http://metadata-standards.org/11179/" \o "http://metadata-standards.org/11179/)

[NIEM] OMG *UML Profile for NIEM Version 1.0*, [http://www.omg.org/spec/NIEM-UML/1.0/](http://www.omg.org/spec/NIEM-UML/1.0/" \o "http://www.omg.org/spec/NIEM-UML/1.0/)

[OCL] OMG *Object Constraint Language (OCL), Version 2.4*, <http://www.omg.org/spec/OCL/2.4/>

[ODM] OMG *Ontology Definition Metamodel (ODM) Version 1.1*, <http://www.omg.org/spec/ODM/1.1/>

[QVT] OMG *Meta Object Facility (MOF) 2.0 Query/View/Transformation, V1.2 (Beta)*, <http://www.omg.org/spec/QVT/1.2/Beta/>

[UML] OMG *Unified Modeling Language (UML) Version 2.5 – Beta 2*, <http://www.omg.org/spec/UML/2.5/Beta2/>

# Terms and Definitions

For the purposes of this specification, the following terms and definitions apply.

Archetype

An archetype is a re-usable formal definition of domain level information defined in terms of constraints on an information model. The key feature of the archetype approach to computing is a complete separation of information models (such as object models of software or models of database schemas) from domain models.

Archetype Definition Language (ADL)

ADL is a formal language for expressing archetypes. It provides a formal, textual syntax for describing constraints on any domain entity whose data is described by an information model (also known as the 'underlying reference model'). The ADL syntax is semantically equivalent to the AOM and represents one possible serialization of the AOM. The current version of ADL is known as 'ADL 2'.

Archetype Instance

An archetype instance is a single instantiation of data conforming to a specific archetype. In the context of CIMI this data will typically be clinical.

Archetype Model (AM)

An AM is a re-usable, formal model of an archetype expressed as a computable set of constraint statements on an underlying reference model (URM). Concepts that can be modeled using archetypes include weight measurement, blood pressure, microbiology results, discharge referral, prescription, or diagnosis. CIMI archetypes will be represented as an instance of the ‘Archetype Object Model’.

Archetype Object Model (AOM)

The AOM is the definitive expression of archetype semantics and is independent of any particular syntax. It is defined as an object model using a UML class diagram. It is a generic model, meaning it can be used to express archetypes for any reference model in a standard way. Version 1.4 of the AOM was standardized in ISO-13606:2. The current version is known as 'AOM 2'.

Archetype Query Language (AQL)

The AQL is a declarative query language developed specifically for expressing queries used for searching and retrieving the clinical data found in archetype-based EHRs. AQL expresses queries at the archetype level, i.e. semantic level, and not at the data instance level. This is key to achieving shared queries across system or enterprise boundaries.

Clinical Data Repository (CDR)

A CDR is a data store holding and managing clinical data collected from service encounters at the point-of-service locations such as hospitals, clinics, etc.

Clinical Document Architecture (CDA)

A CDA is an HL7 XML-based markup standard intended to specify the encoding, structure, and semantics of clinical documents for exchange.

Clinical Information Model (CIM)

A CIM is a representation of the structured clinical information (including relationships, constraints and terminology) describing a specific clinical concept - e.g. a blood pressure observation, a Discharge Summary, or a Medication Order.

Clinical Information Modeling Initiative (CIMI)

CIMI is an initiative established to “improve the interoperability of healthcare information systems through shared implementable clinical information models.”

Clinical Information Modeling Initiative (CIMI) Reference Model (RM)

The CIMI RM is the underlying Reference Model on which CIMI's clinical models (i.e. archetypes) are defined. This reference model defines a rigorous and stable set of modeling patterns, including a set of complex data types, information patterns (e.g. data, qualifier, state), and structural patterns (e.g. composition, entry, tree). All CIMI clinical models (i.e. archetypes) will be defined by constraining the CIMI RM. The RM is intended to be instantiated with patient data which conforms to the constraints defined by the associated clinical model.

Clinical Model Governance

Clinical Model Governance is a set of policies and processes through which the high clinical quality of all clinical artifacts (including clinical models and-or archetypes) is maintained during creation, storage, verification, maintenance, and distribution, by, for, and on behalf of CIMI.

Clinical Model Repository

The Clinical Model Repository is a data store holding clinical information models and associated artifacts in an agreed sharable format.

Clinical Model Verification

Clinical Model Verification is the act of reviewing, inspecting, or testing in order to establish a clinical model specification meets appropriate clinical safety and quality standards.

Clinical Modeling Language

A Clinical Modeling Language is a modeling language defining clinical information models.

Clinical Requirement

Clinical Requirements are requirements articulating clinical needs including clinical practices, standards, guidelines, principles, and other clinical concepts.

Code System

A Code System is a managed collection of uniquely identifiable concepts with associated representations. A code system may also form an ontological system for representing a set of concepts, e.g. SNOMED-CT, LOINC, ICD-10, etc.

Common Terminology Services 2 (CTS2)

CTS2 is an OMG specification providing a standard interface to disparate terminology sources. The Information Model specifies the structural definition, attributes, and associations of resources common to structured terminologies such as Code Systems, Binding Domains, and Value Sets. The Computational Model specifies the service descriptions and interfaces needed to access and maintain structured terminologies.

Concept

In information modeling, a concept represents an “idea” as a word or phrase in order to support human understanding, but may also be represented with a concept identifier in order to bind it to a controlled terminology or ontology.

Concept Domain

A Concept Domain is a named category of like concepts bound to one or more coded elements in an information model. Concept Domains exist to constrain the intent of the coded element and are independent of any specific vocabulary, code system, or Realm. A Concept Domain provides a high level grouping for all things possible in a given domain from which value sets will be constructed.

Concept Domain Binding

A Concept Domain Binding is the association of a value set with a concept domain in a given context.

Conceptual Information Model

A Conceptual Information Model is a representation of real-world objects and their relationships and constraints as understood by domain experts. A conceptual model should include no implementation-specific details.

Conformance

Conformance is the requirement that those who participate in CIMI by contributing data components or creating and sharing ADL artifacts are following the agreed-upon procedures for doing so and that all documentation meets minimum criteria and the CIMI Naming and Design Rules where applicable.

Constraint Model

A Constraint Model is a formal specification used for describing constraints on an Underlying Reference Model. The Constraint Model is used to express clinical information models (i.e. archetypes), not to be confused with the clinical information models that are instances of the constraint model.

Detailed Clinical Model

A Detailed Clinical Model is a relatively small standalone information model designed to express a precise clinical concept in a standardized and reusable manner.

Fully Defined Concept

A Fully Defined Concept is a concept uniquely defined by a set of defining relationships.

Information Model

An Information Model is a structured representation of the information requirements of a domain including the classes of information required and their attributes, relationships, and constraints.

Node

A Node is a named part of an information model.

Ontology

An Ontology is a formal representation of knowledge as a set of concept identifiers, terms describing the concepts so identified, and the relationships among them.

Reference Model

A Reference Model is an information model defining a set of modeling patterns upon which clinical models are defined.

Reference Terminology

A Reference Terminology is a terminology designed to provide common semantics for diverse implementations.

Semantic Binding

Semantic Binding is the association of a node in an information model with a concept from a controlled terminology representing its meaning.

Terminology

A Terminology is a vocabulary of technical terms used in a particular field, subject, science, or art.

Terminology Binding

Terminology Binding is the assertion of a relationship between an information model and a terminology.

Value Binding

Value Binding is the association of a given node in a clinical model with the set of valid concepts that may populate it.

Value Set

A Value Set is a set of concept identifiers deemed valid for use in a specific context, especially to define the domain of a data element.

# Symbols

## Graphical Symbols

No AML-specific graphical symbols are defined in this specification.

## Abbreviations

ADL Archetype Definition Language

AM Archetype Model

AML Archetype Modeling Language

AOM Archetype Object Model

AQL Archetype Query Language

CDA Clinical Document Architecture

CDL Clinical Document Language

CDR Clinical Data Repository

CIM Clinical Information Model

CIMI Clinical Information Modeling Initiative

CMP Constraint Model Profile

CRM Clinical Reference Model

CTS2 Common Terminology Services 2

EHR Electronic Health Record

HL7 Health Level Seven

ICD-10 International Statistical Classification of Diseases and Related Health Problems, 10th Edition

LOINC Logical Observation Identifiers Names and Codes

MDA Model Driven Architecture

OCL Object Constraint Language

OMG Object Management Group

OpenEHR Open Electronic Health Record

PIM Platform Independent Model

PSM Platform Specific Model

RM Reference Model

RMP Reference Model Profile

SNOMED CT Systematized Nomenclature of Medicine – Clinical Terms

TBP Terminology Binding Profile

UML Unified Modeling Language

URI Uniform Resource Identifier

URM Underlying Reference Model

# Additional Information

## Changes to Adopted OMG Specifications

No changes to adopted OMG specifications are required to adopt this specification.

## Acknowledgements

The following companies submitted this specification:

1. Mayo Clinic
2. Visumpoint, LLC

The following companies supported this specification:

1. Escape Velocity, LLC

The following individuals aided the development of this specification:

|  |  |
| --- | --- |
| 1. Thomas Beale | Ocean Informatics |
| 1. Dave Carlson | Intermountain Healthcare (Consultant) |
| 1. Angelique Cortez | Accenture |
| 1. Bob Daniel | Escape Velocity LLC |
| 1. Tom Digre | Visumpoint LLC |
| 1. Stanley M. Huff, MD | Intermountain Healthcare |
| 1. Patrick Langford | Intermountain Healthcare (Consultant) |
| 1. Robert Lario | Visumpoint LLC |
| 1. Jay Lyle | Ockham Information Services LLC |
| 1. Virginia Riehl | Independent Consultant |
| 1. Deepak Kumar Sharma, M.S. | Mayo Clinic |
| 1. Harold Solbrig | Mayo Clinic |
| 1. Mason Tran | Visumpoint LLC |
| 1. Michael van der Zel | University Medical Center Groningen  Results4Care |

# The AOM and the AML Metamodel

This section describes the purpose behind the AML Metamodel and how it relates to the AOM. The actual AML Metamodel can be found in Appendix A

# specification

## Introduction

## Scope

## Profiles

### Dependencies

### ReferenceModelProfile [Profile]

### Terminology Binding Profile

The Terminology Binding Profile supports the binding of information models terminology, with optional support for binding to CTS2. Profile bindings include:

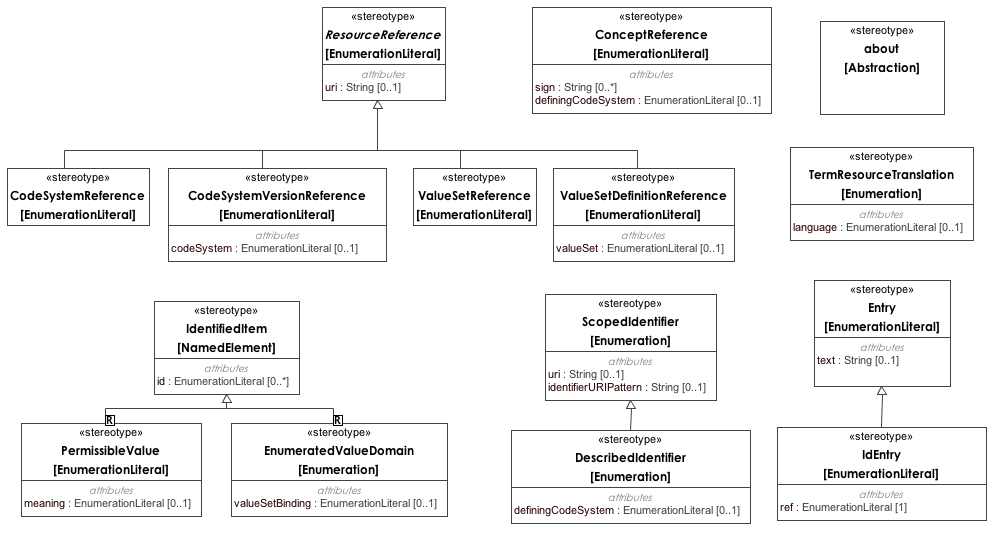
1. *Value Bindings:* Linkage of the data model to value domains, which restrict the valid value(s) of an attribute to a set of values that correspond to a set of meanings recorded in an external terminology;
2. *Semantic Bindings*: Definition of the meaning of model elements using concepts in an external terminology;
3. *Constraint Bindings:* Specifying constraints on the information model, using concepts and relationships defined in an external terminology.

The Terminology Binding Profile includes the UML equivalent of the ADL 2.0 terminology section, including:

* Identifiers -- The IdentifiedItem stereotype allows "id", "at" and "ac" identifiers to be assigned to Class constraints, Enumeration Literals and Enumerations respectively.
* Term definitions -- The ResourceTranslation, Entry and IdEntry stereotypes allow language specific text/ description tuples to be assigned directly to model elements (Entry) or indirectly to identified elements
* Term bindings
  + Model elements may be associated with a concept reference in an external terminology using the *about* association, which includes term bindings for ADL 2.0 "id" codes
  + Enumerations may be associated with a value set and optional definition that identifies the list of possible "meanings" that can be associated with the owned enumeration literals, which includes term bindings for ADL "ac" codes
  + Enumeration literals may be associated concept references in an external terminology that define the intended meaning of the enumeration literal in the context of the containing enumeration which includes term bindings for ADL "at" codes

The Terminology Binding profile draws on the ISO 11179-3 model for the identification, designation, definition and value / meaning binding aspects and on the OMG Common Terminology Services 2 (CTS2) specification for the model of Concept, Code System, Code System Version, Value Set and Value Set Definition references.

#### Terminology Binding Profile



**Terminology Binding Profile**

##### ArchetypeType [Enumeration]

**Description**

An Enumeration which represents the type of an Archetype. The ArchetypeType is the UML representation for Archetype structural variants described by AOM 2.0 in terms of the AOM Archetype attributes is\_template and is\_overlay.

**Diagrams**

,

**Enumeration Literals**

* **archetype**

The Archetype is structured as a source Archetype.

Source archetypes can be specialised, in which case their definition structure is a partial overlay on the flat parent, or ‘top-level’, in which case the definition structure is complete. C\_ARCHETYPE\_ROOT instances may only occur representing direct references to other archetypes - ‘external references’.

An "archetype" corresponds to an AOM ARCHETYPE in which both is\_template and is\_overlay are false.

* **template**

The Archetype is structured as a Template.

A source template is an archetype containing C\_ARCHETYPE\_ROOT objects representing slot fillers - each referring to an external archetype or template, or potentially an overlay archetype.

Corresponds to an AOM Archetype with is\_template=True.

* **template\_overlay**

The archetype is structured as a template overlay.

These are purely local components of templates, and include only the *definition* and *terminology*. The definition structure is always a specialised overlay on something else, and may not contain any slot fillers or external references, i.e. no C\_ARCHETYPE\_ROOT objects. No identifier, *adl\_version*, *languages* or *description* are required, as they are considered to be propagated from the owning root template.

Accordingly, template overlays act like a simplified specialised archetype. Template overlays

can be thought of as being similar to ‘anonymous’ or ‘inner’ classes in some object-oriented programming languages.

A template\_overlay corresponds to an AOM ARCHETYPE with both is\_template and is\_overlay set to true.

##### about [Stereotype]

**Description**

The about stereotype extends the Abstraction association and associates an AML model element with its intended ontological meaning. The about stereotype models the ISO 11179-3 "meaning" association between a Data Element and a Data Element Concept and a Value Domain and a Conceptual Domain.

**Diagrams**

[Terminology Binding Profile](#_b3fed7a67275752ffa266629e157b6c0)

**Meta-classes**

UML Standard Profile::UML2 Metamodel::Abstraction

**Constraints**

* **isConceptReference**

The supplier (target) of the about stereotype must be stereotyped by ConceptReference

[OCL]

self.base\_Abstraction.supplier->select(c|c.stereotypedBy('ConceptReference'))->size()=1

##### ArchetypeTerm [Stereotype]

**Description**

Representation of any coded entity (term or constraint) in the archetype terminology.

The ADL id corresponds to the ArchetypTerm name.

An <<ARCHETYPE\_TERM>> corresponding to an AOM ac-code may define a value set by populating the tag value\_set\_members with the sibling <<ARCHETYPE\_TERM>>s representing AOM at-codes.

An <<ARCHETYPE\_TERM>> corresponding to an AOM at-code may be associated with a term\_binding. A term\_binding is modeled by populating the tag term\_bindings with <<ConceptReference>> EnumerationLiterals corresponding to each technology binding associated with that at-code.

**Diagrams**

[Terminology Binding Profile](#_b3fed7a67275752ffa266629e157b6c0),

**Direct Superclasses (Generalization)**

[TerminologyProfile::PermissibleValue](#_5bb7ce8128b60ee5eb2ca275444e9692)

**Meta-classes**

UML Standard Profile::UML2 Metamodel::EnumerationLiteral

**Attributes**

**•** value\_set\_members : UML Standard Profile::UML2 Metamodel::EnumerationLiteral [0..\*]

When the <<ARCHETYPE\_TERM>> corresponds to an AOM ac-code, value\_set\_members represents value-set relationships. This <<ARCHETYPE\_TERM>> represents an ac-code, the value\_set\_members represent at-codes. All of the at-codes must be in the same Enumeration as the ac-code.

This tag encapsulates the AOM concept of value\_sets within the archetype terminology. The specification of value\_set\_members should only occur in the ARCHETYPE's terminology\_original\_language.

**•** term\_bindings : UML Standard Profile::UML2 Metamodel::EnumerationLiteral [0..\*]

This tag is used to specify term bindings for the <<ARCHETYPE\_TERM>>. A term binding may be used to specify the "meaning" of the <<ARCHETYPE\_TERM>>, either in the ISO 11179 sense or as a reference to a terminology definition provided by an external service. In either case, the term binding is a <<ConceptReference>> within an Enumeration representing the AOM concept of term\_binding.

**Constraints**

* **VTLC- language consistency**

VTLC- language consistency. Languages consistent: all term codes and constraint codes exist in all languages.

[OCL]

self.base\_EnumerationLiteral.namespace.oclAsType(Enumeration).general->notEmpty() and self.base\_EnumerationLiteral.namespace.oclAsType(Enumeration).general->forAll(language|language.namespace.oclAsType(Package).ownedType ->select(t|(t<>language)and t.stereotypedBy('ResourceTranslation')).oclAsType(Enumeration).\_generalizationOfGeneral.specific.oclAsType(Enumeration) ->forAll(sibling|sibling.ownedLiteral->exists(ol|ol.name=self.base\_EnumerationLiteral.name)) )

##### AssumedValue [Stereotype]

**Description**

AssumedValue associates a ConceptReferenceConstraint with the ConceptReference that is the "assumed value" for the constrained object. An "assumed value" is the default value for an element if it is absent in a data record. Note that "assumed value" is not the same as "default value", which, in this context, is the value that is provided as the default when a new data record is being created. Default values are considered to be implementation details and are not represented in the AML specification.

**Diagrams**

[Terminology Binding Profile](#_b3fed7a67275752ffa266629e157b6c0),

**Meta-classes**

UML Standard Profile::UML2 Metamodel::Abstraction

##### CodeSystemReference [Stereotype]

**Description**

A reference to a code system (aka. "terminology", "classification scheme", "ontology").

**Diagrams**

[Terminology Binding Profile](#_b3fed7a67275752ffa266629e157b6c0)

**Direct Superclasses (Generalization)**

[TerminologyProfile::ResourceReference](#_1b2eec63ad4ef6c72d57b9985e0346ff)

**Meta-classes**

UML Standard Profile::UML2 Metamodel::EnumerationLiteral

**Constraints**

* **version**

There must be at most 1 imported > Package.

[Binary]

self.base\_Package.packageImport.importedPackage->select(p|p.stereotypedBy('CodeSystemVersionReference'))->size()<=1

##### CodeSystemVersionReference [Stereotype]

**Description**

"A reference to a specific version of a code system and, if known, the code system which it is a version of." [cts2]

**Diagrams**

[Terminology Binding Profile](#_b3fed7a67275752ffa266629e157b6c0)

**Direct Superclasses (Generalization)**

[TerminologyProfile::ResourceReference](#_1b2eec63ad4ef6c72d57b9985e0346ff)

**Meta-classes**

UML Standard Profile::UML2 Metamodel::EnumerationLiteral

**Attributes**

**•** codeSystem : UML Standard Profile::UML2 Metamodel::EnumerationLiteral [0..1]

A reference to the code system that this reference is a version if. The codeSystem tag is used in situations where the code system itself has a well-known URI but there referenced version does not. In this case the version URI can be omitted and the reference used in its place.

**Constraints**

* **referencesCodeSystem**

codeSystem must reference an EnumerationLiteral stereotyped by CodeSystem

[OCL]

not(self.codeSystem.oclIsUndefined()) implies self.codeSystem.stereotypedBy('CodeSystemReference')

##### ConceptReference [Stereotype]

**Description**

ConceptReference is the scoped identifier of a concept. The name of the base EnumerationLiteral is the code (aka. "id", "label" and in the case of CTS2, "name") of the target concept. The scoping namespace is supplied by the owning ScopedIdentifier.

**Diagrams**

[Terminology Binding Profile](#_b3fed7a67275752ffa266629e157b6c0)

**Meta-classes**

UML Standard Profile::UML2 Metamodel::EnumerationLiteral

**Attributes**

**•** sign : UML Standard Profile::UML2 Metamodel::PrimitiveTypes::String [0..\*]

A contextually appropriate name or signifier for the referenced concept.

**•** definingCodeSystem : UML Standard Profile::UML2 Metamodel::EnumerationLiteral [0..1]

A reference to the code system or code system version that contains the (preferred) description of this concept reference. If present, definingCodeSystem overrides the definingCodeSystem tag for the owning Enumeration stereotyped by the DescribedIdentifier profile.

**Constraints**

* **describingCodeSystem**

The CodeSystemReference or CodeSystemVersionReference from which the complete description of the referenced concept was derived, if any. If describingCodeSystem is a CodeSystemReference, the description can derived from whatever is considered "Current" (in the CTS2 [cts2] sense) in a given implementation environment.

[English]

describingCodeSystem:=self.base\_\*.clientDependency.supplier->select(p|p.stereotypedBy('CodeSystemReference'))->asSequence()->first()

* **mustBeScopedIdentifier**

The owning enumeration must be stereotyped with ScopedIdentifier

[OCL]

self.base\_EnumerationLiteral.namespace.stereotypedBy('ScopedIdentifier')

* **mustBeCodeSystemOrVersion**

If present, definingCodeSystem must be stereotyped by CodeSystemReference or CodeSystemVersionReference.

[English]

not(self.definingCodeSystem.oclIsUndefined()) implies( self.definingCodeSystem.stereotypedBy('CodeSystemReference') or self.definingCodeSystem.stereotypedBy('CodeSystemVersionReference') )

* **definingCodeSystem**

The EnumerationLiteral referenced by definingCodeSystem must be stereotyped by CodeSystemReference or CodeSystemVersionReference

[OCL]

not(self.definingCodeSystem.oclIsUndefined()) implies( self.definingCodeSystem.stereotypedBy('CodeSystemReference') or self.definingCodeSystem.stereotypedBy('CodeSystemVersionReference') )

##### DescribedIdentifier [Stereotype]

**Description**

The DescribedIdentifier stereotype is a ScopedIdentifier whose ownedLiterals are concept references. DescribedIdentifier includes an optional codeSystem tag that can reference the code system or code system version that describes the owned concept references. The codeSystem of an owned concept reference can be overridden on the concept reference level.

**Diagrams**

[Terminology Binding Profile](#_b3fed7a67275752ffa266629e157b6c0)

**Direct Superclasses (Generalization)**

[TerminologyProfile::ScopedIdentifier](#_59faf6918f4c546323d6df67392c366b)

**Meta-classes**

UML Standard Profile::UML2 Metamodel::Enumeration

**Attributes**

**•** definingCodeSystem : UML Standard Profile::UML2 Metamodel::EnumerationLiteral [0..1]

The default definingCodeSystem for the owned ConceptReferences.

**Constraints**

* **membersMustBeConceptReference**

All of the member EnumerationLiterals must be stereotyped by ConceptReference.

[English]

self.base\_Enumeration.ownedLiteral->forAll(ol|ol.stereotypedBy('ConceptReference'))

* **definingCodeSystem**

definingCodeSystem, if present, must reference an enumeration literal stereotyped by CodeSystemReference or CodeSystemVersionReference

[OCL]

not(self.definingCodeSystem.oclIsUndefined()) implies( self.definingCodeSystem.stereotypedBy('CodeSystemReference') or self.definingCodeSystem.stereotypedBy('CodeSystemVersionReference') )

##### Entry [Stereotype]

**Description**

A language specific name (text) and optional description of a NamedElement.

**Diagrams**

[Terminology Binding Profile](#_b3fed7a67275752ffa266629e157b6c0)

**Meta-classes**

UML Standard Profile::UML2 Metamodel::EnumerationLiteral

**Direct Subclasses (Specialization)**

[TerminologyProfile::IdEntry](#_639b489bb158a6e00785ad9ecb38b643)

**Attributes**

**•** text : UML Standard Profile::UML2 Metamodel::PrimitiveTypes::String [0..1]

If present, the actual text of the translation. If absent, the text of the translation is the name of the base EnumerationLiteral. The text tag exists to address the situation where two or more elements in the model have the same name in the same language. In this situation, both of the Entry elements cannot have the same name (text), so, instead they are given arbitrary unique names and the text tag carries the actual translation.

##### EnumeratedValueDomain [Stereotype]

**Description**

The EnumeratedValueDomain stereotype represents a discrete set of possible values (PermissibleValues) for a particular field or data element. Each PermissibleValue identifies a unique value and (optionally) its intended meaning.

An EnumeratedValueDomain may reference a value set or value set definition. Implementations may use this reference to validate the PermissibleValue meaning links, populate the permissible values in the EnumeratedValueDomain or provide selection lists for mapping existing mappings.

**Diagrams**

[Terminology Binding Profile](#_b3fed7a67275752ffa266629e157b6c0)

**Direct Superclasses (Generalization)**

[TerminologyProfile::IdentifiedItem](#_4b28f60cd7e8328f1d31dbcfa39d2ff3),

[TerminologyProfile::IdentifiedItem](#_4b28f60cd7e8328f1d31dbcfa39d2ff3)

**Meta-classes**

UML Standard Profile::UML2 Metamodel::Enumeration

**Attributes**

**•** valueSetBinding : UML Standard Profile::UML2 Metamodel::EnumerationLiteral [0..1]

The identifier of the value set or value set definition whose resolution defines the set of possible value meanings for this set of permissible values.

**Constraints**

* **meaningIsConceptReference**

There is a maximum of one > Abstractions.

[OCL]

self.base\_Enumeration.clientDependency->select(d|d.stereotypedBy('ValueSetReferenceInstance'))->size()<=1

* **permissibleValues**

All ownedLiterals must be stereotyped by PermissibleValue

[OCL]

self.base\_Enumeration.ownedLiteral->forAll(x:EnumerationLiteral|x.stereotypedBy('PermissibleValue') or x.stereotypedBy('ArchetypeTerm'))

* **VTVSID- value-set id defined**

VTVSID- value-set id defined. The identifying code of a value set must be defined in the term definitions of the terminology of the current archetype.

[English]

self.base\_Enumeration.ownedLiteral->isEmpty() implies self.base\_Enumeration.clientDependency ->select(d|d.oclIsKindOf(Usage)).supplier ->select(s|s.stereotypedBy('ARCHETYPE\_TERM')).name ->exists(n|n=self.base\_Enumeration.name)

* **VTVSUQ- value-set members unique**

VTVSUQ- value-set members unique. The member codes of a value set must be unique within the value set.

[English]

self.base\_Enumeration.ownedLiteral->isEmpty() implies( self.base\_Enumeration.clientDependency.select->(d|d.oclIsKindOf(Usage)).supplier->select(s|s.stereotypedBy('ARCHETYPE\_TERM'))->asSet()->size() =self.base\_Enumeration.clientDependency.select->(d|d.oclIsKindOf(Usage)).supplier->select(s|s.stereotypedBy('ARCHETYPE\_TERM'))->asSequence()->size() )

* **bindingIsValueSetOrDefinition**

valueSetBinding, if present, must reference an enumeration that is stereotyped by "ValueSetReference" or "ValueSetDefinitionReference"

[OCL]

not(self.valueSetBinding.oclIsUndefined()) implies( self.valueSetBinding.stereotypedBy('ValueSetReference') or self.valueSetBinding.stereotypedBy('ValueSetDefinitionReference') )

##### IdentifiedItem [Stereotype]

**Description**

The IdentifiedItem stereotype allows one or more instances of a ScopedIdentifier to be assigned to a NamedElement.

**Diagrams**

[Terminology Binding Profile](#_b3fed7a67275752ffa266629e157b6c0)

**Meta-classes**

UML Standard Profile::UML2 Metamodel::NamedElement

**Direct Subclasses (Specialization)**

[TerminologyProfile::EnumeratedValueDomain](#_c7f411daaf64f83e013bec437cb8f30a),

[TerminologyProfile::PermissibleValue](#_5bb7ce8128b60ee5eb2ca275444e9692)

**Attributes**

**•** id : UML Standard Profile::UML2 Metamodel::EnumerationLiteral [0..\*]

An identifier for the NamedElement.

**Constraints**

* **scopedIdentifierLiteral**

Every id property is an instance of a ScopedIdentifier.

[English]

// this constraint does not seem to be useful anymore self.id.oclAsType(EnumerationLiteral)->forAll(x:EnumerationLiteral | x.namespace.stereotypedBy('ScopedIdentifier'))

* **uniqueScopes**

Every id must belong to a unique instance specification classifier. An identified Item cannot have two or more identifiers drawn from the same ScopedIdentifier enumeration.

[OCL]

self.id->forAll(l1 | self.id->forAll(l2 | l1.oclAsType(EnumerationLiteral).namespace = l2.oclAsType(EnumerationLiteral).namespace implies l1 = l2))

##### IdEntry [Stereotype]

**Description**

A term / description translation for a NamedElement with an IdentifiedItem stereotype.

**Diagrams**

[Terminology Binding Profile](#_b3fed7a67275752ffa266629e157b6c0)

**Direct Superclasses (Generalization)**

[TerminologyProfile::Entry](#_03adbea6748e4e63f69642986c8df42d)

**Meta-classes**

UML Standard Profile::UML2 Metamodel::EnumerationLiteral

**Attributes**

**•** ref : UML Standard Profile::UML2 Metamodel::EnumerationLiteral [1]

A reference to the id tag of a named element that stereotyped by IdentifiedItem.

##### Infrastructure [Stereotype]

**Description**

A stereotype indicating a base Property represents an aspect of an Archetype implementation such as a specific Archetype identifier or other element. Properties with the Infrastructure stereotype cannot be constrained in AML.

**Diagrams**

[Terminology Binding Profile](#_b3fed7a67275752ffa266629e157b6c0),

**Meta-classes**

UML Standard Profile::UML2 Metamodel::Property

##### MappedDataType [Stereotype]

**Description**

Used in a Reference Model to define the relationship between a RM Classifier and an AML primitive type. AML Primitive types are defined by the UML Type Library and/or the XML Primitive Type Library. The client of the Abstraction is a Reference Model Classifier. The supplier of the Abstraction is an AML Primitive type. The mapping of the Abstraction defines the transformations between the RM Classifier and its AML Primitive Type counterpart. Note that AML constraints are defined with respect to AML Primitive Types even when the Type being constrained is expressed as a Reference Model Type.

**Diagrams**

[Terminology Binding Profile](#_b3fed7a67275752ffa266629e157b6c0),

**Meta-classes**

UML Standard Profile::UML2 Metamodel::Abstraction

**Constraints**

* **isAMLDataType**

The supplier AML Primitive Type must be an AML Primitive Type defined in the UML Primitive Type or XML Primitive Type libraries.

[OCL]

self.base\_Abstraction.supplier->exists(s|s.oclIsKindOf(PrimitiveType) and ((s.namespace.name='XMLPrimitiveTypes') or (s.namespace.name='PrimitiveTypes')))

##### PermissibleValue [Stereotype]

**Description**

A possible value in a data record. A permissible value may be a context specific code (e.g. 0, 1, "M", "A", etc.) a concept identifier (e.g. "74400008", "16285-9"), a URI or, in the case of ADL, an "AT" code. Note that the meaning of the permissible value is assigned by its meaning.

**Diagrams**

[Terminology Binding Profile](#_b3fed7a67275752ffa266629e157b6c0)

**Direct Superclasses (Generalization)**

[TerminologyProfile::IdentifiedItem](#_4b28f60cd7e8328f1d31dbcfa39d2ff3),

[TerminologyProfile::IdentifiedItem](#_4b28f60cd7e8328f1d31dbcfa39d2ff3)

**Meta-classes**

UML Standard Profile::UML2 Metamodel::EnumerationLiteral

**Direct Subclasses (Specialization)**

[TerminologyProfile::ArchetypeTerm](#_2516a2c296345f1369282ef5b8ff5bf1)

**Attributes**

**•** meaning : UML Standard Profile::UML2 Metamodel::EnumerationLiteral [0..1]

The ConceptReference that provides the meaning for the permissible value

**Constraints**

* **valueMeaningIsConceptReference**

PermissibleValue.about must be a concept reference

[OCL]

self.base\_EnumerationLiteral.clientDependency->select(d|d.stereotypedBy('ConceptReferenceInstance'))->size()<=1

* **VATDF- value code validity**

VATDF- value code validity. Each value code (at-code) used in a term constraint in the archetype definition must be defined in the term\_definitions part of the terminology of the flattened form of the current archetype.

[English]

(self.base\_EnumerationLiteral.getNearestPackage().name='value\_sets') implies self.base\_EnumerationLiteral.getNearestPackage().nestingPackage.nestedPackage ->select(p|p.name='term\_definitions').ownedType ->select(t|t.oclIsKindOf(Enumeration)).oclAsType(Enumeration) ->forAll(e|e.ownedLiteral->exists(l|l.name=self.base\_EnumerationLiteral.name))

* **VTVSMD- value-set members defined**

VTVSMD- value-set members defined. The member codes of a value set must be defined in the term definitions of the terminology of the flattened form of the current archetype.

[OCL]

(self.base\_EnumerationLiteral.getNearestPackage().name='value\_sets') implies self.base\_EnumerationLiteral.getNearestPackage().nestingPackage.nestedPackage ->select(p|p.name='term\_definitions').ownedType ->select(t|t.oclIsKindOf(Enumeration)).oclAsType(Enumeration) ->forAll(e|e.ownedLiteral->exists(l|l.name=self.base\_EnumerationLiteral.name))

* **mustBeConceptReference**

meaning, if present, must reference an EnumerationLiteral that is stereotyped by ConceptReference

[OCL]

not(self.meaning.oclIsUndefined()) implies self.meaning.stereotypedBy('ConceptReference')

##### PossibleValue [Stereotype]

**Description**

A link to a ConceptReference that supplies a possible (allowed) value for a ConceptReferenceConstraint.

**Diagrams**

[Terminology Binding Profile](#_b3fed7a67275752ffa266629e157b6c0),

**Meta-classes**

UML Standard Profile::UML2 Metamodel::Abstraction

##### ReferenceModel [Stereotype]

**Description**

This stereotype identifies a package as a Reference Model -- a package which contains the collection of UML Classes that can be constrained by the Archetypes in an Archetype Library. The Reference Model stereotype also allows the specification of the publisher, namespace and version of a Reference Model in a form compatible with a modeling language such as ADL.

An archetype library is associated with exactly one <<ReferenceModel>>, and it may use the properties that the model defines in the construction of archetype identifier strings. See the openEHR Archetype Definition Language exposition of archetype identification for one example of how this may work.

A Reference Model is identified via the name of its publisher in combination with a RM Package. A Reference Model is structured to support multiple domains, while reusing patterns and core classes. The domain packages define the semantics of enterprise level health information types, including the EHR and demographics. The domain packages are named by the 'RM Package' tag from a referencing <<ArchetypeLibrary>> and based on the notion of 'closure'. The notion of ‘closure’ is represented as a top level package from which the focal class can be reached. In general, a given class can be reached from more than one top level package, but an archetype of that class will only be suitable for one of those packages. For example, the openEHR class CLUSTER is used by classes in both the ehr and demographic top level packages. However, an archetype of CLUSTER will usually be designed for use with only one of those packages. The Cluster archetype physical\_examination for example will only make sense in data defined by the ehr package. Consequently, it will have an archetype identifier of the form openEHR-EHR-CLUSTER.physical\_examination. In AML, the closure of a <<ReferenceModel>> includes the transitive nested content of the <<ReferenceModel>>.

**Diagrams**

[Terminology Binding Profile](#_b3fed7a67275752ffa266629e157b6c0),

**Meta-classes**

UML Standard Profile::UML2 Metamodel::Package

**Attributes**

**•** rmPublisher : UML Standard Profile::UML2 Metamodel::PrimitiveTypes::String [0..1]

The name of the Reference Model publisher. Corresponds to *rm\_publisher* in AOM 2.0

**•** rmNamespace : UML Standard Profile::UML2 Metamodel::PrimitiveTypes::String [0..1]

The owning domain name of the archetype. Corresponds to the *namespace* attribute in AOM2.0.

**•** rmVersion : UML Standard Profile::UML2 Metamodel::PrimitiveTypes::String [0..1]

Designates the version id of the reference model on which the archetype is based. Corresponds to *rm\_release* in AOM 2.0

##### ResourceReference [Stereotype]

**Description**

ResourceReference couples a local identifier with an optional URI which references the target resource. ResourceReference models the CTS2 [cts2] NameAndMeaningReference data type, where the domain is determined by the specializing stereotype and the name by the name of the base EnumerationLiteral. The CTS2 href attribute is not part of ResourceReference as it is an aspect of a service instance, not a model.

**Diagrams**

[Terminology Binding Profile](#_b3fed7a67275752ffa266629e157b6c0)

**Meta-classes**

UML Standard Profile::UML2 Metamodel::EnumerationLiteral

**Direct Subclasses (Specialization)**

[TerminologyProfile::CodeSystemReference](#_7de70b4fbd9e6a164f7f00cde47dfd5a),

[TerminologyProfile::CodeSystemVersionReference](#_9a8de95c38ebe2d6ce506cbc9bef7b7a),

[TerminologyProfile::ValueSetDefinitionReference](#_a4fedb7858ead8d2272640d51b53719a),

[TerminologyProfile::ValueSetReference](#_1a1ca20b54028ee5e2eb20af35411f6e)

**Attributes**

**•** uri : UML Standard Profile::UML2 Metamodel::PrimitiveTypes::String [0..1]

The URI of the referenced resource. "A globally unique URI that identifies the intended meaning of the identifier." [cts2]

**Constraints**

* **uniqueId**

Every identifier must come from a different namespace

[OCL]

self.id->forAll(i1 | self.id->forAll(i2 | i1.enumeration = i2.enumeration implies i1 = i2))

* **language sign description represent tuple**

The ordered sequences language, sign, description must be the same length and language must be part of a > Enumeration.

[OCL]

self.language->size()=self.sign->size() and self.language->size()=self.description->size() and self.language.enumeration->forAll(l|l.stereotypedBy('Language'))

##### Runtime [Stereotype]

**Description**

Indicates that the base Property represents a dynamic or "runtime" element that cannot be constrained using AML.

**Diagrams**

[Terminology Binding Profile](#_b3fed7a67275752ffa266629e157b6c0),

**Meta-classes**

UML Standard Profile::UML2 Metamodel::Property

##### ScopedIdentifier [Stereotype]

**Description**

The ScopedIdentifier stereotype models both the ISO 11179-3[iso11179] namespace, "... a set of designations and/or scoped identifiers for a particular business need" and Scoped\_Identifier, "the identifier of an identified item within a specified namespace". ScopedIdentifier extends ENumeration, where Enumeration plays the role of the scoping namespace and the owned EnumerationLiterals the contained identifiers. A ScopedIdentifier may include an optional URI that identifies the scoping namespace and, if necessary, a uri pattern that defines how uri's for the contained identifiers are constructed.

As an example, the SNOMED CT identifier namespace would have a uri of "http://snomed.info/id/", indicating that an EnumerationLiteral named 74400008 would be represented as "http://snomed.info/id/74400008".

**Diagrams**

[Terminology Binding Profile](#_b3fed7a67275752ffa266629e157b6c0)

**Meta-classes**

UML Standard Profile::UML2 Metamodel::Enumeration

**Direct Subclasses (Specialization)**

[TerminologyProfile::DescribedIdentifier](#_7935aa90cdd719e5718a690fd5dd0137)

**Attributes**

**•** uri : UML Standard Profile::UML2 Metamodel::PrimitiveTypes::String [0..1]

The URI of the namespace. As an example, one might have a ScopedIdentifier named "owl" with a URI of "http://www.w3.org/2002/07/owl#".

**•** identifierURIPattern : UML Standard Profile::UML2 Metamodel::PrimitiveTypes::String [0..1]

A URI substitution pattern, where "$1" indicates where the name of an owned EnumerationLiteral would be substituted to create a URI. Example: http://loinc.org/id/$1. If no URI substitution pattern is supplied, URI's are assumed to be constructed by concatenating the name of an enumeration literal onto the value of the *uri* attribute.

##### TermResourceTranslation [Stereotype]

**Description**

ResourceTranslation is a collection of designations and descriptions/definitions in a target language. ResourceTranslation represents a refactoring of the ISO 11179-3[iso11179] Designatable\_Item, where Designation sign is represented as the name of the extended EnumerationLiteral and the Definition text as the associated comment(s). The language attribute is represented by the language tag.

**Diagrams**

[Terminology Binding Profile](#_b3fed7a67275752ffa266629e157b6c0)

**Meta-classes**

UML Standard Profile::UML2 Metamodel::Enumeration

**Attributes**

**•** language : UML Standard Profile::UML2 Metamodel::EnumerationLiteral [0..1]

The identifier of the target language.

**Constraints**

* **translationEntries**

All of the ownedLiterals must be stereotyped by Entry.

[OCL]

self.base\_Enumeration.ownedLiteral->forAll(ol|ol.stereotypedBy('Entry'))

* **uniqueEntries**

The ref tags of the ownedLiterals must all be unique. No two translation entries may reference the same identifier.

[OCL]

self.base\_Enumeration.ownedLiteral->size() = self.base\_Enumeration.ownedLiteral.appliedStereotype('IdEntry').oclAsType(IdEntry).ref->asSet()->size()

##### ValueSetDefinitionReference [Stereotype]

**Description**

"A reference to a set of rules for constructing a value set along with the corresponding value set if known" [cts2]

**Diagrams**

[Terminology Binding Profile](#_b3fed7a67275752ffa266629e157b6c0)

**Direct Superclasses (Generalization)**

[TerminologyProfile::ResourceReference](#_1b2eec63ad4ef6c72d57b9985e0346ff)

**Meta-classes**

UML Standard Profile::UML2 Metamodel::EnumerationLiteral

**Attributes**

**•** valueSet : UML Standard Profile::UML2 Metamodel::EnumerationLiteral [0..1]

The reference to the value set that is defined by this value set definition. This tag reduces the requirement to assign unique uri's to each value set definition. If the definition itself has a uri, the valueSet tag link can provide sufficient information to get a known definition without having to generate a new uri.

**Constraints**

* **specializesValueSet**

A ValueSetDefinitionReference is a specialization of a ValueSetReference.

[OCL]

self.base\_Enumeration.general->select(g|g.stereotypedBy('ValueSetReference'))->size()<=1

* **referencesValueSet**

The referenced EnumerationLiteral must be stereotyped by ValueSet.

[English]

The referenced EnumerationLiteral must be stereotyped by ValueSet.

##### ValueSetReference [Stereotype]

**Description**

"A reference to a named set of entity references." [cts2] ValueSetReference references a set of ConceptReferences. The members of the set can vary over time and context and depend on (a) the particular value set definition (aka. version) of the value set d and (b) the particular version of the code system(s) that are used to to resolve the rules in the value set definition.

**Diagrams**

[Terminology Binding Profile](#_b3fed7a67275752ffa266629e157b6c0)

**Direct Superclasses (Generalization)**

[TerminologyProfile::ResourceReference](#_1b2eec63ad4ef6c72d57b9985e0346ff)

**Meta-classes**

UML Standard Profile::UML2 Metamodel::EnumerationLiteral

**Constraints**

* **definition**

This Enumeration must have exactly one > Generalization.

[OCL]

self.base\_Enumeration.general->select(g|g.stereotypedBy('ValueSetDefinitionReference'))->size()<=1

### ConstraintProfile [Profile]

# Appendix A: AML MetaModel