



Archetype Semantics

<i>Editors:</i> T Beale ^a		
<i>Revision:</i> 0.5	<i>Pages:</i> 17	<i>Date of issue:</i> 16 Mar 2007

a. Ocean Informatics

Keywords: semantics, archetype

© 2003-2007 The *openEHR* Foundation.

The *openEHR* Foundation is an independent, non-profit community, facilitating the sharing of health records by consumers and clinicians via open-source, standards-based implementations.

Founding Chairman David Ingram, Professor of Health Informatics, CHIME, University College London

Founding Members Dr P Schloeffel, Dr S Heard, Dr D Kalra, D Lloyd, T Beale

email: info@openEHR.org **web:** <http://www.openEHR.org>

Copyright Notice

© Copyright openEHR Foundation 2001 - 2007
All Rights Reserved

1. This document is protected by copyright and/or database right throughout the world and is owned by the openEHR Foundation.
2. You may read and print the document for private, non-commercial use.
3. You may use this document (in whole or in part) for the purposes of making presentations and education, so long as such purposes are non-commercial and are designed to comment on, further the goals of, or inform third parties about, openEHR.
4. You must not alter, modify, add to or delete anything from the document you use (except as is permitted in paragraphs 2 and 3 above).
5. You shall, in any use of this document, include an acknowledgement in the form: "© Copyright openEHR Foundation 2001-2007. All rights reserved. www.openEHR.org"
6. This document is being provided as a service to the academic community and on a non-commercial basis. Accordingly, to the fullest extent permitted under applicable law, the openEHR Foundation accepts no liability and offers no warranties in relation to the materials and documentation and their content.
7. If you wish to commercialise, license, sell, distribute, use or otherwise copy the materials and documents on this site other than as provided for in paragraphs 1 to 6 above, you must comply with the terms and conditions of the openEHR Free Commercial Use Licence, or enter into a separate written agreement with openEHR Foundation covering such activities. The terms and conditions of the openEHR Free Commercial Use Licence can be found at http://www.openehr.org/free_commercial_use.htm

Amendment Record

Issue	Details	Who	Completed
R E L E A S E 1.0.1			
0.5	Initial Writing.	T Beale	16 Mar 2007

Table of Contents

1 Introduction

1.1 Purpose

This document provides a description of the semantics of populations *openEHR* archetypes, i.e. archetypes that obey the *openEHR* Archetype Object Model (AOM) specification. The semantics of single archetypes are described in the ADL and AOM specifications.

1.2 Related Documents

Prerequisite documents for reading this document include:

- The *openEHR* Architecture Overview;
- The *openEHR* Archetype Definition Language (ADL) specification;
- The *openEHR* Archetype Object Model (AOM) specification.

1.3 Status

This document is under development, and is published as a proposal for input to standards processes and implementation works.

The latest version of this document can be found in PDF format at http://svn.openehr.org/specification/TRUNK/publishing/architecture/am/archetype_semantics.pdf. New versions are announced on openehr-announce@openehr.org.

2 Overview

This specification describes the formal semantics of populations of archetypes, i.e. beyond the semantics of a single archetype. The areas dealt with include identification; archetype path relationships; conformance; and the various archetype-archetype relationships including specialisation, composition, versioning etc.

To Be Continued:

3 Archetype Identification

Archetypes can be identified with various kinds of identifiers. Currently, the multi-axial archetype identifier specified by the `rm.support.identification.ARCHETYPE_ID` class in the *openEHR Support IM* is used. Other identifier types such as an ISO Oid could be deployed in the future, most likely in parallel with the multi-axial identifier.

3.1 Multi-axial Archetype Identifier

The archetype identifier is declared in the heading section of an archetype, e.g.

```
openEHR-EHR-OBSERVATION.haematology.v1
```

The multi-axial archetype identifier encodes the partitioning of the archetype space in the identifier. Each identifier instance denotes a single archetype within a versioned 2-dimensional space, with the dimensions being:

- reference model entity, i.e. target of archetype;
- domain concept.

As with any multi-axial identifier, the underlying principle of an archetype id is that all parts of the id must be able to be considered immutable. This means that no variable characteristic of an archetype (e.g. accrediting authority, which might change due to later accreditation by another authority, or may be multiple) can be included in its identifier. The syntax of an Archetype id is as follows:

archetype_id: qualified_rm_entity '.' domain_concept '.' version_id

qualified_rm_entity: rm_originator '.' rm_name '.' rm_entity_name

domain_concept: concept_name { '.' specialisation }*

version_id: 'v' NUMBER

rm_originator: NAME

The name of the authority issuing the underlying reference model to which this archetype applies e.g. openEHR, HL7, CEN etc.

rm_name: NAME

The name of the model as labelled by the model_originator e.g. ehr, demographics, CDA etc.

rm_entity_name: NAME

The name of the class in the model (model_name) to which this archetype applies.

domain_concept: NAME

A unique string that describes the concept expressed in the archetype e.g. weight

specialisation: NAME

A string which describes this specialisation e.g. 'birth' as a specialisation of 'weight' to describe birth-weight

NUMBER: [0-9]*

NAME: [a-z][a-z0-9()/%\$#&]*

The field meanings are as follows:

rm_originator: id of organisation originating the reference model on which this archetype is based;

rm_name: id of the reference model on which the archetype is based;

rm_entity: ontological level in the reference model;
domain_concept: the domain concept name, including any specialisations;
version_id: numeric version identifier.

Examples of archetype identifiers include:

```
openEHR-EHR-SECTION.physical_examination.v2  
openEHR-EHR-SECTION.physical_examination-prenatal.v1
```

A basic rule for the multi-axial archetype identifier is that it changes as soon as anything is done to the archetype that makes data created using the previous form invalid with respect to the new form. For this reason, version is included in the identifier. The rules for change management are described below.

3.2 Other Identifiers

ISO Oids can be used to unambiguously identify archetypes within storage systems, online repositories etc., regardless of where the archetype sits in the concept space. In order that archetypes can be authored at any place and time without access to OIDs, the OID is optional at all times until publication. The OID can have been issued at any point in the publication lifecycle, but must be unique in the authority's database. Local specialisations do not require OIDs.

4 Conformance

4.1 Data / Archetype Conformance

Data are deemed to *directly conform* to an archetype if:

- every *mandatory path* in the archetype can be found in the data;
- every *path* in the data has a corresponding path in the archetype, i.e. the exact at-code found at every node can be found in the archetype;
- the sibling *occurrence* count of every object in the data is within the occurrences specified at the node found at the relevant path in the archetype;
- the *type* of every object conforms to the type specified at the node found at the relevant path in the archetype;
- the *value* of every leaf attribute in the data is within the range specified at the node found at the relevant path in the archetype;
- every *coded value* in the archetype is within the set of codes specified for the node in the archetype, or, in the case of external constraints, within the set of codes corresponding to the terminology query identifier bound to the ac-code found at the relevant path in the archetype.

This kind of conformance occurs with data that were created directly from the archetype in question. Data may also have been created from a more specialised child of the archetype in question. In this case, the data are deemed to conform if for all items found on paths containing codes from the parent archetype (e.g. at0001), or any derived code (e.g. at0001.3):

- the paths *match* paths in the parent archetype;
- the same conformance conditions as for direct conformance apply, but on matching paths.

There may be codes in the data not derived from a parent archetype, such as at0.0.1. These are found on nodes introduced new in a child archetype.

4.2 Archetype-archetype Conformance

5 Path Relationships

To Be Continued:

6 Archetype Relationships

6.1 Overview

To Be Continued:

6.2 Semantically-related Archetypes

6.2.1 Revision

Revision is the smallest kind of change that can be done to an archetype, and the only one not causing a change in identifier. The new revision is indicated in the `revision_history` section of an archetype. A new revision can safely replace the previous revision in an Archetype Service. Revisions are made in order to:

- modify the `description` section;
- modify the `ontology` section by changing the text descriptions of codes, or adding or changing terminology bindings;
- add or change translations (affects `language` and/or `ontology` and `description` sections);
- broaden an existing constraint in the `definition` section; this often means the changing of a mandatory constraint to an optional one.

Semantic Integrity Constraint

- *Backward compatibility*: data created by precursor of revised archetype are compatible with revised version.

Change Rules

Allowed changes:

- any change to `description` section;
- any change to translations of whole archetype, not affecting semantics, i.e. meanings of terms;
- changes to term bindings;
- changes to constraint bindings where the new term subset is a superset of the previous one (i.e. a broader subset);
- changes to cardinality and optionality such that they are broader than the previous values.

Excluded changes:

- changes to identifier, ADL version, specialisation status;
- `original_language` cannot be changed;
- removal of archetype codes, since this would render data created with the previous revision invalid in some parts;
- removal of any part of the `definition` section.

6.2.2 Specialisation

A specialisation of an archetype is similar in concept to a descendant class in an object-oriented model. Specialised archetypes can redefine nodes and add node in a way that respects the conformance relationship between archetypes; essentially this means that redefine nodes must be narrower or

the same as the corresponding nodes in the parent archetype in terms of occurrences, reference model type and other constraints.

The node term code of a redefined node is defined as a 'dot' specialisation of the parent, e.g. 'at0013.1' is a valid code for the redefined version of the 'at0013' node in a top-level archetype. New nodes are given 'dot' node identifiers specific to the specialisation level; for example 'at0.1' is a code introduced new in a second level specialisation of an archetype.

Specialised archetypes are created when there is a need to customise an existing general purpose archetype, e.g. a 'laboratory result' archetype (containing generic 'panel items') into a particular subtype, such as 'cholesterol result' (containing named 'LDL', 'HDL', 'Total' etc items which are specialisations of the 'panel item' nodes in the parent). Sometimes the customisation is just the addition of new nodes.

Semantic Integrity Constraint

- Specialised archetype must create data that conform to the parent archetype, in the sense described above.

Change Rules

Required changes:

- the identifier of the new archetype must be redefined to a derived form of the identifier of the parent, by creating a specialised concept section of the indentifer;

Allowed changes:

- any change covered by the 'revision' relationship;
- existing object nodes can be redefined as follows:
 - * the at-code of the node is redefined to a derivative at-code of the level of the child archetype, e.g. at0004 is redefined to at0004.3. The meaning of the redefined code specified in the ontology should be be subsumed by that of the parent in the ontological sense (i.e. it must in the 'a kind of' relationship, and every real-world phenomenon or entity that classifies under the child term must also classify under the parent term);
 - * the reference model type must remain the same or be redefined to a reference model subtype of the type specified in the node being redefined;
 - * if the object node in the parent archetype has occurrences > 1, more than one redefined child node may be created, each with a unique derivative at-code.
- new object nodes can be added, as long as the type conforms to the type and defined by the archetype for the attribute, and as long as the occurrences of the new node conforms with the cardinality of the attribute, taking into account the occurrences of sibling attributes defined in the parent. New nodes are given non-derivative specialised at-codes, e.g. at0.0.1.

Excluded changes:

- exclusions specified in revision relationship;

6.2.3 New Version

A new version of an archetype is created when the existing version of an archetype is incompatible with the current data collection requirement. Given that well-designed archetypes contain mostly optional parts, this occurs rarely. The usual reason is that a mandatory item in the existing archetype

is now regarded as inappropriate; a value range or coded term set may also be incorrect from the point of view of current requirements. Almost all other required changes can be achieved by a new revision or specialisation.

A new version is created when the archetype required models the same content as the existing archetype, and is intended to replace it in time. If this is not the case, a new (unrelated) archetype is created. When a new version is created, the same archetype identifier is used, with the version number in the final segment incremented, thus generating a new archetype identifier, as in the following example:

```
openEHR-EHR-EVALUATION.problem.v1  
openEHR-EHR-EVALUATION.problem.v2
```

In general, nothing can be formally said about the semantic relationship between different versions of an archetype. However, it is strongly recommended in all archetype environments that:

- changes between versions of an archetype be minimised;
- a conversion algorithm to make data created from previous version(s) compatible with the latest version should be supplied with the latest version.

6.3 Composition

To Be Continued:

END OF DOCUMENT