

DIN EN 13606-2



ICS 35.240.80

Teilweiser Ersatz für
DIN V ENV 13606-2:2000-09

**Medizinische Informatik –
Kommunikation von Patientendaten in elektronischer Form –
Teil 2: Spezifikation für den Austausch von Archetypen;
Englische Fassung EN 13606-2:2007**

Health informatics –
Electronic health record communication –
Part 2: Archetypes interchange specification;
English version EN 13606-2:2007

Informatique de santé –
Dossier de santé informatisé communicant –
Partie 2: Spécification des échanges des archétypes;
Version anglaise EN 13606-2:2007

Gesamtumfang 135 Seiten

Normenausschuss Medizin (NAMed) im DIN

Nationales Vorwort

Diese Norm enthält unter Berücksichtigung des Präsidialbeschlusses 13/1983 den englischen Originaltext der Europäischen Norm EN 13606-2, Ausgabe 2007-08. Die Europäische Norm wurde in der WG I „Information Models“ des CEN/TC 251 „Medizinische Informatik“ erarbeitet. Der Arbeitsausschuss NA 063-07-02 AA „Interoperabilität“ des Fachbereiches 7 „Medizinische Informatik“ des Normenausschusses Medizin im DIN hat an der Erarbeitung mitgewirkt.

Für die in diesem Dokument zitierten Internationalen Normen wird im Folgenden auf die entsprechenden Deutschen Normen hingewiesen:

ISO 639-1	siehe DIN 2335, mod.
ISO 1087-1	siehe DIN 2342-1, mod.
ISO 8601	siehe DIN ISO 8601

Änderungen

Gegenüber DIN V ENV 13606-2:2000-09 wurden folgende Änderungen vorgenommen:

- a) Vornormcharakter wurde aufgehoben;
- b) Teil 2 gibt ein generisches Informationsmodell und eine formale Sprache zur Repräsentation und Kommunikation der Definitionen von Archetypen und ihrer Instanzen vor. Im alten Teil 2 wurde eine „Domain-Termlist“ definiert, die eine inhaltliche Bezeichnung der Elemente des Strukturmodells innerhalb eines konkreten EHCR ermöglichte;
- c) Text wurde vollständig neu strukturiert (5 Teile statt vorher 3 Teile) und technisch überarbeitet.

Frühere Ausgaben

DIN V ENV 13606-2:2000-09

Zusammenfassung

Die zur Übertragung von Auszügen einer elektronischen Gesundheitsakte vorgesehenen Objektstrukturen des Referenzmodells aus EN 13606-1 sind zunächst frei von inhaltlichen Vorgaben. Lediglich Meta-Informationen wie z. B. das Datum der Dokumentation einer Information oder Angaben zur dokumentierenden Person sind fest vorgesehen. Ansonsten können die Informationen beliebig entsprechend des Referenzmodells strukturiert werden. So genannte Archetypen dienen als zweite Modellebene des 2-Modell-Ansatzes (dual model approach) der Definition von Vorgaben für die Strukturierung der Objekte des Referenz-Modells für einen bestimmten, klinisch relevanten Zweck, z. B. ein klinisches Dokument. Archetypen schränken quasi die Freiheit bei der Instanziierung des Referenzmodells ein. Sie können zur Konformitätsprüfung sowie zur Unterstützung von Ein- und Ausgabe von mit Archetypen beschriebenen Informationsstrukturen genutzt werden.

Archetypen können z. B. von Krankenhäusern oder Organisationen wie den medizinischen Fachgesellschaften entwickelt und zur Verwendung in einem festgelegten Umfeld veröffentlicht werden, wobei diese Archetypen dann als Vorgabe für Informationseinheiten wie z. B. einen Arztbrief oder einen Sonographiebefund dienen können.

Teil 2 der EN 13606 enthält ein mit der Unified Modelling Language (UML) beschriebenes objektorientiertes *Archetype Model*, das Archetypen als Instanziierungen dieses Modells zu beschreiben erlaubt. Die Einschränkungen, die mit Archetypen formuliert werden können, betreffen nahezu alle Strukturen des Referenzmodells von der Definition von Wertebereichen für Attribute bis hin zur Vorgabe komplexer Objektkombinationen. Dementsprechend finden sich im Archetype Model Klassen für jede Granularität möglicher Einschränkungen.

Archetypen können von anderen Archetypen im Sinne einer einfachen Vererbung abgeleitet werden. Zudem können Archetypen Verweise auf andere Archetypen im Sinne einer Einbettung enthalten. Ein weiteres wichtiges Konzept ist die Unabhängigkeit der Beschreibung von der jeweils verwendeten natürlichen Sprache der Autoren eines Archetyps. So werden alle Konzepte eines Archetyps zunächst mit sprachunabhängigen Bezeichnern benannt, welche dann über eine Archetyp-Ontologie mit beliebig vielen Sprachen im Sinne eines Verzeichnisses verknüpft werden können.

Neben dem Archetype Model gibt Teil 2 der EN 13606 eine formale Sprache zur Definition von Archetypen vor, die Archetype Definition Language (ADL). ADL umfasst zwei Teilsprachen, cADL (constraint form of ADL) und dADL (data definition form of ADL). Während cADL zur Formulierung der eigentlichen Einschränkungen das Reference Model betreffend genutzt wird, können mit der dADL konkrete Instanziierungen von Archetypen bzw. Klassenstrukturen beschrieben werden. Dementsprechend wird bei Beschreibung eines Archetyps mit der ADL die cADL für die formale Definition der Einschränkungen verwendet. Die dADL wird dagegen verwendet für

- Angaben zu den natürlichen Sprachen, für die der Archetyp vorgesehen ist,
- die strukturierte, freitextliche Beschreibung des Archetyps im Hinblick auf Zweck und Verwendungshinweise,
- die Verknüpfung der mittels der cADL sprachunabhängig beschriebenen Konzepte mit Übersetzungen für jede der angegebenen Sprachen sowie gegebenenfalls mit Codes aus Klassifikations- oder Terminologie-Systemen wie z. B. UMLS oder SNOMED-CT,
- Angaben zur Revisionshistorie des jeweiligen Archetyps.

Weitere Beschreibungselemente eines Archetyps sind eine eindeutige Bezeichnung, gegebenenfalls Angaben zum „Vater-Archetyp“, von dem der Archetyp abgeleitet wurde, und zum Konzept, welches den Archetyp charakterisiert.

Teil 2 der EN 13606 liefert so den Bestandteil der Reihe von Normen und Norm-Entwürfen der EN 13606, der die Dynamik in der Entwicklung der Gesundheitsversorgung auf medizinischer und organisatorischer Ebene zu berücksichtigen erlaubt, ohne dass die Interoperabilität der partizipierenden Informationssysteme selbst gefährdet würde.

Nationaler Anhang NA (informativ)

Literaturhinweise

DIN 2335, *Sprachenzeichen*

DIN 2342-1, *Begriffe der Terminologielehre — Grundbegriffe*

DIN ISO 8601, *Datenelemente und Austauschformate — Informationsaustausch — Darstellung von Datum und Uhrzeit*

– Leerseite –

English Version

Health informatics —
Electronic health record communication —
Part 2: Archetypes interchange specification

Informatique de santé —
Dossier de santé informatisé communicant —
Partie 2: Spécification des échanges des archétypes

Medizinische Informatik —
Kommunikation von Patientendaten in elektronischer Form —
Teil 2: Spezifikation für den Austausch von Archetypen

This European Standard was approved by CEN on 21 June 2007.

CEN members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration. Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN Management Centre or to any CEN member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN member into its own language and notified to the CEN Management Centre has the same status as the official versions.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and United Kingdom.



EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

Management Centre: rue de Stassart, 36 B-1050 Brussels

Contents

Page

Foreword	4
0 Introduction	5
0.1 Archetypes	5
0.2 Archetype Repositories	6
0.3 Communicating Archetypes	6
0.4 Overview of the Archetype Model	6
0.5 Overview of ADL	13
0.6 Clinical examples of archetypes	16
1 Scope	16
2 Conformance	16
3 Normative references	16
4 Terms and definitions	17
5 Symbols and abbreviations	18
6 Archetype Representation Requirements	19
6.1 General	19
6.2 Archetype definition, description and publication information	19
6.3 Archetype node constraints	21
6.4 Data Value constraints	23
6.5 Profile in relation to EN 13606-1 Reference Model	24
7 Archetype Model	26
7.1 Introduction	26
7.2 Overview	29
7.3 The Archetype Package	33
7.4 The Archetype Description Package	35
7.5 The Constraint Model Package	39
7.6 The Assertion Package	46
7.7 The Primitive Package	50
7.8 The Ontology Package	56
7.9 The Domain Extensions Package	58
7.10 The Support Package	60
7.11 Generic Types Package	69
7.12 Domain-specific Extensions (Informative)	70
8 Archetype Definition Language (ADL)	71
8.1 dADL - Data ADL	71
8.2 cADL - Constraint ADL	89
8.3 Assertions	114
8.4 ADL Paths	118
8.5 ADL - Archetype Definition Language	119
Bibliography	130

Figures

Figure 1 — ADL Archetype Structure 14

Figure 2 — Package structure 28

Figure 3 — Overview of the main part of the Archetype Model – Part 1 29

Figure 4 — Overview of the Archetype Model - Part 2 30

Figure 5 — Archetype Package..... 33

Figure 6 — Archetype Description Package 35

Figure 7 — Constraint Model Package..... 39

Figure 8 — Assertion Package 46

Figure 9 — Primitive Package 50

Figure 10 — Ontology Package 56

Figure 11 — Domain Extensions Package..... 58

Figure 12 — Support Package 60

Figure 13 — Generic Types Package 69

Figure 14 — Example Domain-specific package 70

Foreword

This document (EN 13606-2:2007) has been prepared by Technical Committee CEN/TC 251 “Health informatics”, the secretariat of which is held by NEN.

This document shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by February 2008 and conflicting national standards shall be withdrawn at the latest by February 2008.

This document will supersede ENV 13606-2:2000.

This multipart standard under the general heading Health informatics – Electronic health record communication consists of the following parts:

Part 1: Reference model

Part 2: Archetypes interchange specification

Part 3: Reference archetypes and term lists

Part 4: Security

Part 5: Exchange models

According to the CEN/CENELEC Internal Regulations, the national standards organizations of the following countries are bound to implement this European Standard: Austria, Belgium, Bulgaria, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and United Kingdom.

0 Introduction

Comprehensive, multi-enterprise and longitudinal electronic health records will often in practice be achieved through the joining up of multiple clinical applications, databases (and increasingly devices) that are each tailored to the needs of individual conditions, specialties or enterprises.

This requires that EHR data from diverse systems be capable of being mapped to and from a single comprehensive representation, which is used to underpin interfaces and messages within a distributed network (federation) of EHR systems and services. This common representation has to be sufficiently generic and rich to represent any conceivable health record data, comprising part or all of an EHR (or a set of EHRs) being communicated.

The approach adopted in this standard, underpinned by international research on the EHR, has been to define a rigorous and generic Reference Model that is suitable for all kinds of data and data structures within an EHR, and in which all labelling and context information is an integral part of each construct. An EHR Extract will contain all of the names, structure and context required for it to be interpreted faithfully on receipt even if its organisation and the nature of the clinical content have not been “agreed” in advance.

However the wide-scale sharing of health records, and their meaningful analysis across distributed sites, also requires that a consistent approach is used for the clinical (semantic) data structures that will be communicated via the Reference Model, so that equivalent clinical information is represented consistently. This is necessary in order for clinical applications and analysis tools safely to process EHR data that have come from heterogeneous sources.

0.1 Archetypes

The challenge for EHR interoperability is therefore to devise a generalised approach to representing every conceivable kind of health record data structure *in a consistent way*. This needs to cater for records arising from any profession, speciality or service, whilst recognising that the clinical data sets, value sets, templates etc. required by different health care domains will be diverse, complex and will change frequently as clinical practice and medical knowledge advance. This requirement is part of the widely acknowledged health informatics challenge of *semantic interoperability*.

The approach adopted by this standard distinguishes a Reference Model, used to represent the generic properties of health record information, and Archetypes (conforming to an Archetype Model), which are meta-data used to define patterns for the specific characteristics of the clinical data that represent the requirements of each particular profession, speciality or service.

The Reference Model is specified as an ODP Information Viewpoint model, representing the global characteristics of health record components, how they are aggregated, and the context information required to meet ethical, legal and provenance requirements. In this standard, the Reference Model is defined in Part 1. This model defines the set of classes that form the generic building blocks of the EHR. It reflects the stable characteristics of an electronic health record, and would be embedded in a distributed (federated) EHR environment as specific messages or interfaces (as specified in Part 5 of this standard).

Archetypes are effectively pre-coordinated combinations of named RECORD_COMPONENT hierarchies that are agreed within a community in order to ensure semantic interoperability, data consistency and data quality.

For an EHR_Extract as defined in Part 1 of this standard, an archetype specifies (and effectively constrains) a particular hierarchy of RECORD_COMPONENT sub-classes, defining or constraining their names and other relevant attribute values, optionality and multiplicity at any point in the hierarchy, the data types and value ranges that ELEMENT data values may take, and may include other dependency constraints. Archetype instances themselves conform to a formal model, known as an Archetype Model (which is a constraint model, also specified as an ODP Information Viewpoint Model). Although the Archetype Model is stable, individual archetype instances may be revised or succeeded by others as clinical practice evolves. Version control ensures that new revisions do not invalidate data created with previous revisions.

Archetypes may be used within EHR systems to govern the EHR data committed to a repository. However, for the purposes of this interoperability standard, no assumption is made about the use of archetypes within the EHR Provider system whenever this standard is used for EHR communication. It is assumed that the

original EHR data, if not already archetyped, may be mapped to a set of archetypes, if desired, when generating the EHR_EXTRACT.

The Reference Model defined in Part 1 of this standard has attributes that can be used to specify the archetype to which any RECORD_COMPONENT within an EHR_EXTRACT conforms. The class RECORD_COMPONENT includes an attribute *archetype_id* to identify the archetype and node to which that RECORD_COMPONENT conforms. The *meaning* attribute, in the case of archetyped data, refers to the primary concept to which the corresponding archetype node relates. However, it should be noted that Part 1 does not require that archetypes are used to govern the hierarchy of RECORD_COMPONENTS within an EHR_EXTRACT: the archetype-related attributes are optional in that model. It is recognised that the international adoption of an archetype approach will be gradual, and may take some years.

0.2 Archetype Repositories

The range of archetypes required within a shared EHR community will depend upon its range of clinical activities. The total set needed on a national basis is presently unknown, but there might eventually be several thousand archetypes globally. The ideal sources of knowledge for developing such archetype definitions will be clinical guidelines, care pathways, scientific publications and other embodiments of best practice. However, “de facto” sources of agreed clinical data structures might also include:

- the data schemata (models) of existing clinical systems;
- the lay-out of computer screen forms used by these systems for data entry and for the display of analyses performed;
- data-entry templates, pop-up lists and look-up tables used by these systems;
- shared-care data sets, messages and reports used locally and nationally;
- the structure of forms used for the documentation of clinical consultations or summaries within paper records;
- health information used in secondary data collections;
- the pre-coordinated terms in terminology systems.

Despite this list of *de facto* ways in which clinical data structures are currently represented, these formats are very rarely interoperable. The use of standardised archetypes provides an interoperable way of representing and sharing these specifications, in support of consistent (good quality) health care record-keeping and the semantic interoperability of shared EHRs.

In the longer term, it is anticipated that the involvement of national health services, academic organisations and professional bodies in the development of archetypes will enable this approach to contribute to the pursuit of quality evidence-based clinical practice. In the future regional or national public domain libraries of archetype definitions might be accessed via the Internet, and downloaded for local use within EHR systems.

0.3 Communicating Archetypes

This part standard specifies the requirements for a comprehensive and interoperable archetype representation, and defines the ODP Information Viewpoint representation for the Archetype Model and an optional archetype interchange format called Archetype Definition Language (ADL).

This standard does not require that any particular model be adopted as the internal architecture of archetype repositories, services or components used to author, store or deploy archetypes in collaboration with EHR services. It does require that these archetypes are capable of being mapped to the Archetype Model defined in this part-standard in order to support EHR communication and interoperability within an EHR-sharing community.

0.4 Overview of the Archetype Model

This section provides a general informative description of the model that is specified in Clause 7 of this part standard.

The overall archetype model consists of identifying information, a description (its meta-data), a definition (expressed in terms of constraints on instances of an object model), and an ontology. Identifying information and lifecycle state are part of the *ARCHETYPE* class. The archetype description is separated into revision history information and descriptive information about the archetype. Revision history information is concerned with the committal of the archetype to a repository, and takes the form of a list of audit trail items, while descriptive information describes the archetype itself (regardless of whether it has been committed to a repository of any kind).

The archetype definition, the 'main' part of the archetype model, is an instance of a *C_COMPLEX_OBJECT*, since the root of the constraint structure of an archetype shall always take the form of a constraint on a non-primitive object type. The fourth main part of the archetype model, the ontology, is represented by its own class, and is what allows the archetypes to be natural language- and terminology-neutral.

An enumeration class, *VALIDITY_KIND* is also included in the Archetype package. This is intended to be used as the type of any attribute in this constraint model whose value is logically 'mandatory', 'optional', or 'disallowed'. It is used in this model in the classes *C_Date*, *C_Time* and *C_Date_Time*

Archetypes contain some natural language elements, including the description and ontology definitions. Every archetype is therefore created in some original language, which is recorded in the *original_language* attribute of the *ARCHETYPE* class. An archetype is translated by doing the following:

- translating every language-dependent element to the new language;
- adding a new *TRANSLATION_DETAILS* instance to *ARCHETYPE.translations*, containing details about the translator, organisation, quality assurance and so on.

The *languages_available* function provides a complete list of languages in the archetype.

The Archetype Definition

The main definitional part of an archetype consists of alternate layers of object and attribute constraining nodes, each containing the next level of nodes. In this section, the word 'attribute' refers to any data property of a class, regardless of whether regarded as a 'relationship' (i.e. association, aggregation, or composition) or 'primitive' (i.e. value) attribute. At the leaves are primitive object constraint nodes constraining primitive types such as String, Integer etc. There are also nodes that represent internal references to other nodes, constraint reference nodes that refer to a text constraint in the constraint binding part of the archetype ontology, and archetype constraint nodes, which represent constraints on other archetypes allowed to appear at a given point. The full list of node types is as follows:

- *C_complex_object*: any interior node representing a constraint on instances of some non-primitive type, e.g. *ENTRY*, *SECTION* ;
- *C_attribute*: a node representing a constraint on an attribute (i.e. UML 'relationship' or 'primitive attribute') in an object type;
- *C_primitive_object*: an node representing a constraint on a primitive (built-in) object type;
- *Archetype_internal_ref*: a node that refers to a previously defined object node in the same archetype. The reference is made using a path;
- *Constraint_ref*: a node that refers to a constraint on (usually) a text or coded term entity, which appears in the ontology section of the archetype, and in ADL, is referred to with an "acNNNN" code. The constraint is expressed in terms of a query on an external entity, usually a terminology or ontology;
- *Archetype_slot*: a node whose statements define a constraint that determines which other archetypes may appear at that point in the current archetype. Logically it has the same semantics as a *C_COMPLEX_OBJECT*, except that the constraints are expressed in another archetype, not the current one.

The Archetype Description

EN 13606-2:2007 (E)

What is normally considered the 'meta-data' of an archetype, i.e. its author, date of creation, purpose, and other descriptive items, is described by the *ARCHETYPE_DESCRIPTION* and *ARCHETYPE_DESCRIPTION_ITEM* classes. The parts of this that are in natural language, and therefore may require translated versions, are represented in instances of the *ARCHETYPE_DESCRIPTION_ITEM* class. If an *ARCHETYPE_DESCRIPTION* has more than one *ARCHETYPE_DESCRIPTION_ITEM*, each of these should carry exactly the same information in a different natural language.

When an archetype is translated for use in another language environment, each *ARCHETYPE_DESCRIPTION_ITEM* needs to be copied and translated into the new language.

The *AUDIT_DETAILS* class is concerned with the creation and modification of the archetype in a repository. Each instance of this class in an actual archetype represents one act of committal to the repository, with the attributes documenting who, when and why.

NOTE Revision of an archetype should be limited to modification of the descriptive information, adding language translations and/or term bindings. If the definition part of an archetype is no longer valid it should instead be replaced with a new archetype to ensure that corresponding EHR data instances each conform to the same archetype specification.

The Archetype Ontology

All linguistic entities are defined in the ontology part of the archetype. There are four major parts in an archetype ontology: term definitions, constraint definitions, term bindings and constraint bindings. The former two define the meanings of various terms and textual constraints which occur in the archetype; they are indexed with unique identifiers, which are used within the archetype definition body. The latter two ontology sections describe the mappings of terms used internally to external terminologies.

Archetype Specialisation

Archetypes may be specialised: an archetype is considered a specialisation of another archetype if it specifies that archetype as its parent, and only makes changes to its definition such that its constraints are 'narrower' than those of the parent. Any data created via the use of the specialised archetype shall be conformant both to it and to its parent.

Every archetype has a 'specialisation depth'. Archetypes with no specialisation parent have depth 0, and specialised archetypes add one level to their depth for each step down a hierarchy required to reach them.

Archetype Composition

Archetypes may be composed to form larger structures semantically equivalent to a single large archetype. Archetype slots are the means of composition, and are themselves defined in terms of constraints.

Data types and the Support package

The model is dependent on three groups of assumed types, whose names and assumed semantics are described by ISO/IEC 11404. The first group comprises the most basic types:

- Any
- Boolean
- Character
- Integer
- Real
- Double
- String

The second is the assumed library types:

- Date
- Time
- Date_Time
- Duration

These types are supported in most implementation technologies, including XML, Java and other programming languages. They are not defined in this specification, allowing them to be mapped to the most appropriate concrete types in each implementation technology.

The third group is the Generic types:

- List<T> (ordered, duplicates allowed)
- Set<T> (unordered, no duplicates)
- Hash <T, K > (keyed list of items of any type)
- Interval <T> (interval of instances of any ordered type)

Although these types are supported in most implementation technologies, they are not yet represented in UML. The semantics of these types is therefore defined in the Generic_Types package of the UML model. The remaining necessary types are defined in the Support Package of the Archetype Model.

- ARCHETYPE_ID
- HIER_OBJECT_ID
- TERMINOLOGY_ID
- CODE_PHRASE
- CODED_TEXT

The support package also includes two enumeration classes to provide controlled datasets needed by this part standard.

The Constraint Model Package

Any archetype definition is an instance of a *C_COMPLEX_OBJECT*, which expresses constraints on a class in the underlying Reference Model (Part 1 of this standard), recorded in the attribute *rm_type_name*. A *C_COMPLEX_OBJECT* consists of attributes of type *C_ATTRIBUTE*, which are constraints on the attributes (i.e. any property, including relationships) of that Reference Model class. Accordingly, each *C_ATTRIBUTE* records the name of the constrained attribute (in *rm_attribute_name*), the existence and cardinality expressed by the constraint (depending on whether the attribute it constrains is a multiple or single relationship), and the constraint on the object to which this *C_ATTRIBUTE* refers via its *children* attribute (according to its reference model) in the form of further *C_OBJECTS*.

The key subtypes of *C_OBJECT*, are:

- *C_COMPLEX_OBJECT*;
- *C_PRIMITIVE_OBJECT*;
- *ARCHETYPE_SLOT*.

ARCHETYPE_INTERNAL_REF and *CONSTRAINT_REF* are used to express, respectively, a 'slot' where further archetypes may be used to continue describing constraints; a reference to a part of the current archetype that expresses exactly the same constraints needed at another point; and a reference to a constraint on a constraint defined in the archetype ontology, which in turn points to an external knowledge resource, such as a terminology.

The effect of the model is to create archetype description structures that are a hierarchical alternation of object and attribute constraints. The repeated object/attribute hierarchical structure of an archetype provides the basis for using *paths* to reference any node in an archetype. Archetype paths follow a syntax that is a subset of the W3C Xpath syntax.

All Node Types

All nodes in an archetype constraint structure are instances of the supertype *ARCHETYPE_CONSTRAINT*, which provides a number of important common features to all nodes.

The any_allowed Boolean, if True, indicates that any value permitted by the reference model for that attribute is allowed by the archetype; its use permits the logical idea of a completely "open" constraint to be simply expressed, avoiding the need for any further substructure.

Attribute Nodes

Constraints on attributes are represented by instances of the two subtypes of *C_ATTRIBUTE*: *C_SINGLE_ATTRIBUTE* and *C_MULTIPLE_ATTRIBUTE*. For both subtypes, the common constraint is whether the corresponding instance (defined by the *rm_attribute_name* attribute) must exist. Both subtypes have a list of children, representing constraints on the object value(s) of the attribute.

Single-valued attributes are constrained by instances of the type *C_SINGLE_ATTRIBUTE*, which uses the children to represent multiple alternative object constraints for the attribute value.

Multiply-valued attributes are constrained by an instance of *C_MULTIPLE_ATTRIBUTE*, which allows multiple co-existing member objects of the container value of the attribute to be constrained, along with a cardinality constraint, describing ordering and uniqueness of the container.

Cardinality is only required for container objects such as *List<T>*, *Set<T>* and so on, whereas *existence* is always required. If both are used, the meaning is as follows: the existence constraint says whether the container object will be there (at all), while the cardinality constraint says how many items shall be in the container, and whether it acts logically as a list, set or bag.

Primitive Types

Constraints on primitive types are defined by the classes inheriting from *C_PRIMITIVE*, namely *C_STRING*, *C_INTEGER* and so on.

Constraint References

A *CONSTRAINT_REF* is a proxy for a set of constraints on an object that would normally occur at a particular point in the archetype as a *C_COMPLEX_OBJECT*, but where the actual definition of the constraint is expressed as the binding to a query or expression into an external service (such as an ontology or terminology service) e.g.:

- a set of allowed *CODED_TERMS* e.g. the types of hepatitis;
- an *INTERVAL<QUANTITY>* forming a reference range;
- a set of units or properties or other numerical item.

Assertions

The *C_ATTRIBUTE* and subtypes of *C_OBJECT* enable constraints to be expressed in a structural fashion. In addition to this, any instance of a *C_COMPLEX_OBJECT* may include one or more *invariants*. Invariants are statements in a form of predicate logic, which may be used to state constraints on parts of an object. They are not needed to state constraints on a single attribute (since this can be done with an appropriate *C_ATTRIBUTE*), but are necessary to state constraints on more than one attribute, such as a constraint that 'systolic pressure should be $\geq$ diastolic pressure' in a blood pressure measurement archetype. Such invariants may be expressed using a syntax derived from the OMG's OCL syntax.

Assertions are also used in *ARCHETYPE_SLOTS*, in order to express the 'included' and 'excluded' archetypes for the slot.

Assertions are modelled as a generic expression tree of unary prefix and binary infix operators.

Node_id and Paths

The `node_id` attribute in the class `C_OBJECT` and inherited to all subtypes has two functions:

- it allows archetype object constraint nodes to be individually identified, and in particular, guarantees sibling node unique identification;
- it is the main link between the archetype definition (i.e. the constraints) and the archetype ontology, because each `node_id` is a 'term code' in the ontology.

The existence of `node_ids` in an archetype is what allows archetype paths to be created, which refer to each node.

Domain-specific Extensions

The main part of the archetype constraint model allows any type in a reference model to be archetyped - i.e. constrained - in a standard way: by a regular cascade of `C_COMPLEX_OBJECT` / `C_ATTRIBUTE` / `C_PRIMITIVE_OBJECT` objects. However, lower level logical 'leaf' types may need special constraint semantics that are not conveniently achieved with the standard approach. To enable such classes to be integrated into the generic constraint model, the class `C_DOMAIN_TYPE` is included. This enables the creation of specific "C_" classes, inheriting from `C_DOMAIN_TYPE`, which represent custom semantics for particular reference model types. For example, a class called `C_QUANTITY` might be created which has different constraint semantics from the default effect of a `C_COMPLEX_OBJECT` / `C_ATTRIBUTE` cascade representing such constraints in the generic way (i.e. systematically based on the reference model).

Assumed Values

When archetypes are defined to have optional parts, an ability to define 'assumed' values is useful. For example, an archetype for the concept 'blood pressure measurement' might contain an optional fragment describing the patient position, with choices 'lying', 'sitting' and 'standing'. Since that part of the ENTRY is optional, data could be created according to the archetype that does not contain this information. However, a blood pressure cannot be taken without the patient in some position, so it may be clinically valid to define an implied or 'assumed' value. The archetype allows this to be explicitly stated so that all users/systems know what value to assume when optional items are not included in the data. Assumed values are definable at the leaf level only and may be specified in the `C_PRIMITIVE` classes.

The notion of assumed values is distinct from that of 'default values': default values do appear in data, while assumed values don't.

The Assertion Package

Assertions are expressed in archetypes in typed first-order predicate logic (FOL). They are used in two places: to express archetype slot constraints, and to express invariants in complex object constraints. In both of these places, their role is to constrain something inside the archetype. Constraints on external resources such as terminologies are expressed in the constraint binding part of the archetype ontology.

The concrete syntax of assertion statements in archetypes is designed to be compatible with the OMG Object Constraint Language (OCL). Archetype assertions are statements that contain the following elements:

- variables, which are attribute names, or ADL paths terminating in attribute names (i.e. equivalent of referencing class feature in a programming language);
- manifest constants of any primitive type, plus date/time types
- arithmetic operators: +, *, -, ^ (exponent)
- relational operators: >, <, >=, <=, =, !=, matches
- Boolean operators: not, and, or, xor

EN 13606-2:2007 (E)

- quantifiers applied to container variables: `for_all`, `exists`

The Primitives Package

Ultimately any archetype definition will devolve down to leaf node constraints on instances of primitive types. The primitives package defines the semantics of constraint on such types. Most of the types provide at least two alternative ways to represent the constraint; for example the `C_DATE` type allows the constraint to be expressed in the form of a pattern or an *Interval<Date>*.

The Ontology Package

All linguistic and terminological entities in an archetype are represented in the ontology part of an archetype, whose semantics are given in the Ontology package.

An archetype ontology comprises of the following things.

- A list of terms defined local to the archetype. These are identified by ``atNNNN'` codes, and perform the function of archetype node identifiers from which paths are created. There is one such list for each natural language in the archetype.
- A list of external constraint definitions, identified by ``acNNNN'` codes, for constraints defined external to the archetype, and referenced using an instance of a `CONSTRAINT_REF`. There is one such list for each natural language in the archetype.
- Optionally, a set of one or more bindings of term definitions to term codes from external terminologies.
- Optionally, a set of one or more bindings of the external constraint definitions to external resources such as terminologies.

Specialisation Depth

Any given archetype occurs at some point in a hierarchy of archetypes related by specialisation, where the depth is indicated by the *specialisation_depth* attribute. An archetype which is not a specialisation of another has a *specialisation_depth* of 0. Term and constraint codes introduced in the ontology of specialised archetypes (i.e. which did not exist in the ontology of the parent archetype) are defined in a strict way, using ``.'` (period) markers. For example, an archetype of specialisation depth 2 will use term definition codes like the following:

- ``at0.0.1'` - a new term introduced in this archetype, which is not a specialisation of any previous term in any of the parent archetypes;
- ``at0001.0.1'` - a term which specialises the ``at0001'` term from the top parent. An intervening ``0.'` is required to show that the new term is at depth 2, not depth 1;
- ``at0001.1.1'` - a term which specialises the term ``at0001.1'` from the immediate parent, which itself specialises the term ``at0001'` from the top parent.

This systematic definition of codes enables software to use the structure of the codes to more quickly and accurately make inferences about term definitions up and down specialisation hierarchies. Constraint codes on the other hand do not follow these rules, and exist in a flat code space instead.

Term and Constraint Definitions

Local term and constraint definitions are modelled as instances of the class `ARCHETYPE_TERM`, which is a code associated with a list of name/value pairs. For any term or constraint definition, this list shall at least include the name/value pairs for the names "text" and "description". It might also include such things as "provenance", which would be used to indicate that a term was sourced from an external terminology. The attribute `term_attribute_names` in `ARCHETYPE_ONTOLOGY` provides a list of attribute names used in term and constraint definitions in the archetype, including "text" and "description", as well as any others that are used in various places.

Generic Types Package

This package is included to confirm the semantics of the generic types used in this part standard. Although `List<T>`, `Set<T>`, `Hash<T,K>`, and `Interval<T>` are generic types supported by many programming environments, they are not directly supported in UML. In this package, new types such as `List<String>` are defined using Binding Dependencies between a new Basic Type such as `List<String>` and a Class (`LIST` in this example) that defines the minimum required semantics for all Lists.

Domain-specific Extension (Informative)

Domain-specific classes can be added to the archetype constraint model by inheriting from the class `C_DOMAIN_TYPE`. 7.12.1 (Scientific/Clinical Computing Types) shows the general approach, used to add constraint classes for commonly used concepts in scientific and clinical computing, such as 'ordinal', 'coded term' and 'quantity'. The constraint types shown are `C_ORDINAL`, `C_CODED_TEXT` and `C_QUANTITY` which can optionally be used in archetypes to replace the default constraint semantics represented by the use of instances of `C_OBJECT` / `C_ATTRIBUTE`.

0.5 Overview of ADL

Archetype Definition Language (ADL) is a formal language for expressing archetypes. ADL uses two other syntaxes, cADL (constraint form of ADL) and dADL (data definition form of ADL) to describe constraints on data that are instances the information model specified in Clause 7 of this part-standard.

Archetypes expressed in ADL resemble programming language files, and have a defined syntax. ADL itself is a very simple *glue* syntax, which uses two other syntaxes for expressing structured constraints and data, respectively. The cADL syntax is used to express the archetype **definition**, while the dADL syntax is used to express data, which appears in the **language**, **description**, **ontology**, and **revision_history** sections of an ADL archetype. The top-level structure of an ADL archetype is shown below.

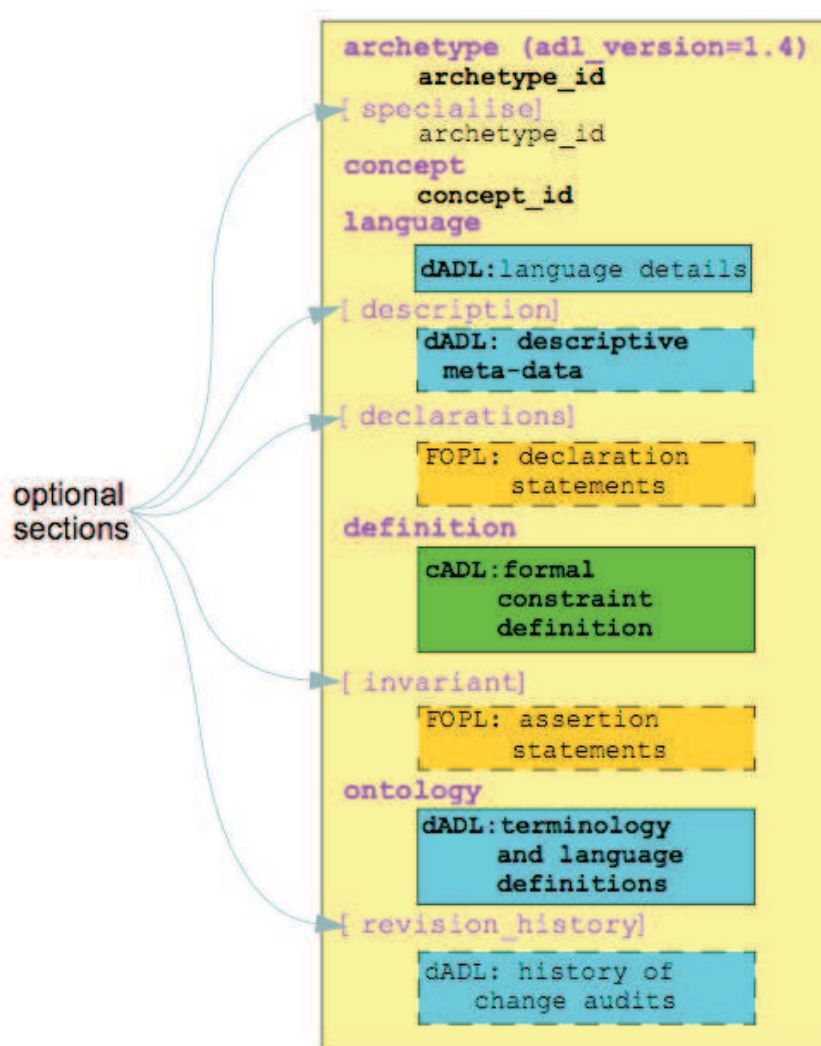


Figure 1 — ADL Archetype Structure

Clause 8 of this part-standard specified dADL, cADL, ADL path syntax, and the combined ADL syntax, archetypes and domain-specific type libraries.

EXAMPLE

The following is an example of a simple archetype. The notion of *guitar* is defined in terms of *constraints* on a *generic* model of the concept **INSTRUMENT**. The names mentioned down the left-hand side of the definition section (INSTRUMENT, size etc) are alternately class and attribute names from an object model. Each block of braces encloses a specification for some particular set of instances that conform to a specific concept, such as guitar or neck, defined in terms of constraints on types from a generic class model. The leaf pairs of braces enclose constraints on primitive types such as Integer, String and Boolean.

```

archetype (adl_version=1.4)
  adl-test-instrument.guitar.draft
concept
  [at0000]                                -- guitar
language
  original_language = <"en">
  translations = <"de", ...>
definition
  INSTRUMENT[at0000] matches {
    size matches {|60..120|      }          -- size in cm
    date_of_manufacture matches {yyyy-mm-??}
      -- year & month ok
    parts cardinality matches {0..*} matches {
      PART[at0001] matches {              -- neck
        material matches {[local::at0003]} -- timber
      }
      PART[at0002] matches {              -- body
        material matches {[local::at0003]} -- timber
      }
    }
  }
}
ontology
  term_definitions = <
    [en] = <
      items = <
        ["at0000"] = <
          text = <"guitar">;
          description = <"stringed instrument">
        >
        ["at0001"] = <
          text = <"neck">;
          description = <"neck of guitar">
        >
        ["at0002"] = <
          text = <"timber">;
          description = <"straight, seasoned timber">
        >
        ["at0003"] = <
          text = <"nickel alloy">;
          description = <"frets">
        >
      >
    >
  >

```

0.6 Clinical examples of archetypes

It is not feasible to include examples of archetypes within this part standard, since they are quite voluminous in document form, but the reader is encouraged to review a selection of archetypes that are available on-line from

<http://svn.openehr.org/knowledge/archetypes/dev/index.html>

This site offers both an ADL representation and an html view of a wide range of archetypes. These examples include language translations and terminology bindings.

1 Scope

This work item consists of the revision of the four part standard ENV 13606 to a full European standard (EN).

This standard specifies the information architecture required for interoperable communications between systems and services that need or provide EHR data. This standard is not intended to specify the internal architecture or database design of such systems.

The subject of the record or record extract to be communicated is an individual person, and the scope of the communication is predominantly with respect to that person's care.

Uses of healthcare records for other purposes such as administration, management, research and epidemiology, which require aggregations of individual people's records, are not the focus of this standard but such secondary uses could also find the standard useful.

Part 2 of this standard defines an Archetype Model to be used to represent Archetypes when communicated between repositories, and between archetype services. It defines an optional serialised representation, which may be used as an exchange format for communicating individual archetypes. Such communication might, for example, be between archetype libraries or between an archetype service and an EHR persistence or validation service.

2 Conformance

The communication of an archetype that is used to constrain part of an EHR_EXTRACT shall conform to the information model defined in Clause 7 of this part standard, and may optionally conform to the specification of Archetype Definition Language defined in Clause 8 of this part standard.

This standard does not prescribe any particular representation of archetypes to be used internally within an archetype repository, server or EHR system. However, it is recommended that any representation used meets the requirements listed in Clause 6 of this part standard.

3 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

EN 14822-2:2005, *Health informatics - General purpose information components - Part 2: Non-clinical*

ISO 639, *Codes for the representation of names of languages*

ISO 1087-1:2000, *Terminology work - Vocabulary - Part 1: Theory and application*

ISO 8601, *Data elements and interchange formats – Information interchange – Representation of dates and times*

ISO/IEC 10646, *Information technology - Universal Multiple-Octet Coded Character Set (UCS)*

ISO/IEC 11179-3:2003, *Information technology - Metadata registries (MDR) - Part 3: Registry metamodel and basic attributes*,

ISO/IEC 11404, *Information technology – Programming languages, their environments and system software interfaces – Language-independent datatypes*

ISO/TS 18308:2004, *Health informatics - Requirements for an electronic health record architecture*

ISO/TR 20514:2005, *Health informatics - Electronic health record - Definition, scope and context*

4 Terms and definitions

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

4.1

abstract class

in Unified Modelling Language, a “virtual” common parent to two or more classes; the abstract class will never be instantiated. Its value in modelling terms is to provide a container for attributes and associations that might apply to several other classes (its sub-classes)

4.2

archetype instance

individual metadata class instance of an Archetype Model, specifying the clinical concept and the value constraints that apply to one class of Record Component instances in an electronic health record extract

4.3

archetype model

information model of the metadata to represent the domain-specific characteristics of electronic health record entries, by specifying values or value constraints for classes and attributes in the electronic health record Reference Model

4.4

archetype repository

persistent repository of archetype definitions, accessed by a client authoring tool or by a run-time component within an electronic health record service

4.5

audit trail

chronological record of activities of information system users which enables prior states of the information to be faithfully reconstructed

4.6

concept

unit of knowledge created by a unique combination of characteristics

[ISO 1087-1:2000]

NOTE Concepts are not necessarily bound to particular languages. They are, however, influenced by the social or cultural background often leading to different categorizations.

4.7

electronic health record

repository of information regarding the health of a subject of care, in computer processable form

[ISO/TR 20514:2005]

4.8

electronic health record entry

health record data in general

EXAMPLE Clinical observations, statements, reasoning, intentions, plans or actions, without particular specification of their formal representation, hierarchical organisation or of the particular Record Component class(es) that might be used to represent them

EN 13606-2:2007 (E)

4.9

electronic health record extract

part or all of the electronic health record of a subject of care, communicated in compliance with EN 13606

4.10

electronic health record system

system for recording, retrieving and manipulating information in electronic health records

4.11

generic

applicable to requirements or information models across healthcare professions, domains and countries

4.12

metadata

data that defines and describes other data

[ISO/IEC 11179-3:2003]

4.13

patient

subject of care

4.14

semantic

relating to meaning in language

4.15

semantic interoperability

ability for data shared by systems to be understood at the level of fully defined domain concepts

[ISO/TS 18308:2004]

4.16

shareable electronic health record

electronic health record with a standardised information model which is independent of electronic health record systems and accessible by multiple authorised users

[ISO/TR 20514:2005]

4.17

subject of care

person scheduled to receive, receiving, or having received health care

[EN 14822-2:2005, modified]

5 Symbols and abbreviations

ADL

Archetype Definition Language

EHR

Electronic Health Record

ODP

Open Distributed Processing

UML

Unified Modelling Language

XML

Extended Markup Language

6 Archetype Representation Requirements

6.1 General

This section lists a set of formal requirements for an archetype representation. This provides the basis on which the archetype model specified in the next section has been designed. It has been necessary to define these requirements within this part-standard as there is little published work on requirements for such a model, unlike the EHR itself for which ISO/TS 18308 has been adopted.

6.2 Archetype definition, description and publication information

6.2.1 The definition of an archetype shall include the following information

- 6.2.1.1** The globally-unique identifier of this archetype definition.
- 6.2.1.2** The identifier of the repository in which this archetype originated or is now primarily held, or of the authority responsible for maintaining it. This repository will be the one in which the definitive publication status of this archetype will be managed.
- 6.2.1.3** The concept that best defines the overall clinical scope of instances conforming to this archetype as a whole, expressed as a coded term or as free text in a given natural language.
- 6.2.1.4** The health informatics domain to which this archetype applies (e.g. EHR). This will map to a set of Reference Models with which this archetype may be used.
- 6.2.1.5** The underlying Reference Model for which this archetype was ideally fashioned (Note: an archetype might be capable of use with more than one relevant Reference Model within a given health informatics domain, but it is expected that the archetype will be optimised for one).
- 6.2.1.6** The natural language in which this archetype was originally defined, represented by its ISO 639-code. In the event of imprecise translations, this is the definitive language for interpretation of the archetype.

6.2.2 The definition of an archetype may include the following information, if applicable

- 6.2.2.1** The globally-unique identifier for the archetype of which this archetype is a specialisation and to which it shall also conform.
- 6.2.2.2** The globally-unique identifier of the former archetype that this definition replaces, if it not the first version of an archetype.
- 6.2.2.3** The reason for defining this new version of a pre-existing archetype.
- 6.2.2.4** The identifier of the replacement for this archetype, if it has been superseded. Note: it might only be possible to add this information by reference within a version-controlled repository; how this is effected is not in scope for this standard.
- 6.2.2.5** An archetype shall have one or more description sets, defining its usage and purpose. Multiple versions of this information may be included, represented in different natural languages or to inform different kinds of potential user.

6.2.3 An archetype description set shall include the following information

- 6.2.3.1** The uniquely-identified party responsible for providing this description set. This identification might optionally include the organisation which that party represents or the authority on which he or she is acting. This may include contact information for that party.
- 6.2.3.2** The natural language in which this description set is provided, represented by its ISO 639-code.
- 6.2.3.3** A formal statement defining the scope and clinical purpose of this archetype, expressed as a coded term or as free text in a given natural language.

EN 13606-2:2007 (E)

NOTE It is recommended that these criteria are expressed as coded terms to improve queries for relevant archetypes from the repository.

EXAMPLE The clinical scope and purpose might specify:

- 1) the principal clinical specialty for which it is intended;
- 2) A list of clinical, medical or procedural terms (keywords): diagnoses, acts, drugs, findings etc.
- 3) the kind of patient in whom it is intended to be used (age, gender, etc.)

6.2.4 An archetype description set may include the following information, if applicable

6.2.4.1 A formal statement of the intended use of this archetype.

NOTE Ideally this would be a coded expression, although a suitable terminology for this is not yet available.

6.2.4.2 A formal statement of situations in which users might erroneously believe this archetype should be used. This may also stipulate any kinds of Reference Model for which it is unsuitable.

6.2.4.3 A detailed explanation of the purpose of this archetype, including any features of particular interest or note. This may include an indication of the persons for which this definition is intended e.g. for students. This information might be included explicitly, and/or by reference (e.g. via a URL).

6.2.4.4 A description, reference or link to the published medical knowledge that has underpinned the definition of this archetype.

6.2.5 An archetype definition shall include a statement of its publication status

6.2.5.1 An archetype definition may evolve through a series of publication states, for example an approval process, without otherwise being changed. These successive states shall be retained as part of the archetype, for audit purposes. However, the modification of the publication status of an archetype shall not itself constitute a formal revision of the identifier by which the archetype is referenced within an EHR_EXTRACT, since the constraint specification will not have been changed.

6.2.6 The publication status of an archetype shall specify the following information

6.2.6.1 The publication status of this archetype, taken from the following list:

- Test/demo
- Tentative
- Draft
- Private
- Public
- Preferred
- Deprecated

6.2.6.2 The date when this particular publication status applied (Note: the first instance of a publication status for this archetype will also be the date when it was first composed).

6.2.6.3 The unique identifier of the party committing this archetype to the repository and thereby asserting this publication status. This identification might optionally include the organisation which that party represents.

6.2.6.4 The unique identifier of the body authorising this change in publication status.

6.2.6.5 The date when it is anticipated that the present publication status, and the archetype content itself, ought to be reviewed to confirm it remains clinically valid.

6.3 Archetype node constraints

6.3.1 General

An archetype definition will include a specification of the hierarchical schema to which instances of data (e.g. EHR data) shall conform. This schema defines the hierarchical organisation of a set of nodes, the relationships between them, and constraints on the permitted values of attributes and data values. These will also conform to the underlying Reference Model(s) for which this definition is applicable.

6.3.2 Archetype node references

6.3.2.1 Any node in the archetype hierarchy might be defined explicitly or, by reference, be specified to be part or whole of a pre-existing archetype.

6.3.2.2 A reference to a pre-existing archetype or archetype fragment may be explicit, by specifying the archetype identifier, and optionally the archetype node of the archetype fragment insertion point.

6.3.2.3 A reference to an archetype fragment may be internal to (i.e. part of) the current archetype.

6.3.2.4 An archetype node may be specified to be one of a set of possible archetypes, by defining an explicit list of candidates and/or by specifying a set of constraints on any of the attributes of an archetype definition.

6.3.2.5 In addition to specifying one or more archetype fragments by reference or constraint, it shall be possible to include an explanation of the rationale for incorporating that specification at the given point in the current archetype hierarchy.

6.3.3 The specification of an archetype node (if not by reference) shall include the following information

6.3.3.1 An internally-unique identifier of this archetype node. When combined with the globally-unique identifier of this archetype definition, it shall be a globally unique reference to the node itself.

6.3.3.2 The class in the instance hierarchy, mapping to the underlying Reference Model for which this archetype was ideally fashioned, that shall be instantiated in order to conform to this archetype node. For an EHR hierarchy conforming to this standard, this class shall be specified using one of the following values:

- FOLDER
- COMPOSITION
- SECTION
- ENTRY
- CLUSTER
- ELEMENT

The number of occurrences, expressed as a range, that may be instantiated corresponding to this archetype node within an instance hierarchy.

6.3.3.3 Other constraints and rules may optionally be specified to govern the creation of instances corresponding to this archetype node.

6.3.3.4 Constraint rules might be expressed as logical conditions, and may include reference to environment parameters such as the current time or location or participants, or be related to the (pre-

EN 13606-2:2007 (E)

existing) values other nodes in the instance hierarchy. Constraint rules might be used to represent the relationship between EHR data and workflow or care pathway processes.

6.3.3.5 Constraint rules might be expressed as inclusion or exclusion criteria.

6.3.3.6 Any constraint rule specification shall identify the formalism (including version) in which it is expressed (e.g. ADL, OWL).

6.3.4 Binding archetype nodes to terms

6.3.4.1 Every node of an archetype schema hierarchy shall be associated with at least one clinical term, which most accurately expresses the intended concept to be represented by that node on instantiation in the corresponding instance hierarchy. This term will usually be included or referenced within the instance.

6.3.4.2 Any node of an archetype may be mapped to any number of additional concepts, terms and synonyms from terminology systems, to support either the interrogation of the archetype repository or of the corresponding instances.

6.3.4.3 Any concept mapping term or text shall specify the purpose that this mapping serves, using one of the following list of values.

- Principal concept
- Term binding
- Synonym
- Language translation

6.3.4.4 Any reference to a coded term shall include the code, rubric, and the coding system (including version) from which the code and rubric have been taken. In addition, it shall be possible to specify the natural language in which this term was mapped, or in which a translation is expressed.

6.3.5 Attribute and association constraints

6.3.5.1 An archetype node may specify constraints on any attributes or associations that correspond to the attributes and associations of that node in the underlying Reference Model.

NOTE These constraints may pre-determine or restrict some or all of the contextual information that is included within the corresponding instance, as represented within the Reference Model. Context information, such as the person to whom a particular observation or inference relates, is formally-represented in most generic EHR-like models to facilitate safe querying and retrieval, even if that information might be inferred from the archetype name or an axis within a terminology system used for the data value. Some archetypes or fragments will pre-determine the values of some of these, which shall be capable of specification within the archetype definition (for example, to constrain the subject of information to be a relative of the patient, and not the patient, in an archetype for family history). The applicability of any aspect of context to a given archetype node will be determined by the context attributes in the corresponding node of the target EHR Reference Models. For example, in the EN 13606 Reference Model the subject of information is represented at the Entry level. There is significant commonality within the set of context information between reference models (for example, between EN 13606 and HL7 CDA Release 2), and ideally a common set of labels for these should be adopted to permit relevant context constraints to be applied to more than one Reference Model.

6.3.6 For any given Reference Model attribute or association it shall be possible to specify the following information

6.3.6.1 The name of the attribute or association, mapping to the underlying Reference Model for which this archetype was ideally fashioned, to which this constraint applies.

6.3.6.2 A common label for this aspect of context. For EHR archetypes, this value is to be taken from a (provisional) list of values as specified in Table 1.

6.3.6.3 For a given Reference Model, if an attribute or association corresponding to this aspect of context is mandatory to be included within a valid EHR instance.

6.3.6.4 The number of instances (expressed as a range) corresponding to this aspect of context that may be instantiated.

6.3.6.5 If multiple instances are permitted, it shall be possible to specify if these are to be represented as an ordered or unordered list.

6.3.6.6 If multiple instances are permitted, it shall be possible to specify if the corresponding data values (of leaf nodes or attributes) shall be unique.

6.3.6.7 Constraints may be specified for the data values of leaf nodes or leaf attributes.

6.3.6.8 Other constraints and rules may optionally be specified to govern the creation of instances corresponding to a Reference Model attribute or association.

6.4 Data Value constraints

6.4.1 It shall be possible to specify constraints and rules for the data values of leaf nodes in the Reference Model hierarchy, or for any other attributes of any archetype node.

6.4.2 It shall be possible to specify the following data value constraint information.

6.4.2.1 If the data value is permitted to have a null value, and optionally to specify a reason (e.g. to specify a null flavour value).

6.4.2.2 If the constraint or rule is an inclusion or exclusion criterion.

6.4.2.3 The formalism (including version) in which this constraint specification is represented.

6.4.2.4 The intended fixed (prescribed) value for conforming instances.

6.4.2.5 The intended default value for conforming instances.

6.4.2.6 A list of permitted candidate values for conforming instances (i.e. to be a subset of those values legally permissible in the underlying Reference Model).

6.4.3 For quantity data types it shall be possible to specify:

- a value range within which values for conforming instances shall lie;
- a range within which values are considered clinically exceptional or critical;
- the intended measurement units for conforming instances.

6.4.4 For date and time data types it shall be possible to specify:

- a value range within which values for conforming instances shall lie;
- the intended measurement units for conforming instances.

6.4.5 For textual data types it shall be possible to specify:

- a string pattern defining a range of possible values;
- the intended coding scheme to be used for conforming instances.

6.4.6 Constraint rules might be expressed as logical conditions, and may include reference to environment parameters such as the current time or location or participants, or be related to the (pre-existing) values other nodes in the instance hierarchy.

6.4.7 The reference to a pre-existing value shall specify that instance precisely and unambiguously. For example, it may be necessary to include a reference to:

EN 13606-2:2007 (E)

- the archetype identifier
- the archetype node identifier
- the attribute or association name
- the occurrence in the instance hierarchy e.g.
 - first
 - most-recent
 - any
 - n ordered by y (the nth element of a set of instances ordered on y)
 - highest value
 - lowest value
 - one or more instances within a (definable) recent time interval
- the intended relationship between this specified instance value and the data value being constrained e.g.
 - the same value as
 - a subset or substring of
 - greater than, greater than or equal to, less than, less than or equal to
 - earlier than, later than, etc.
 - if ... then...
 - shall not be the same as

6.4.8 These relative constraints may be nested, and include logical or set operators in order to represent compound rules.

6.5 Profile in relation to EN 13606-1 Reference Model

This set of tables defines a generic set of contexts that are used within the EN 13606-1 EHR Reference Model to specify how observed values, intentions and inferences relate to the subject of care or to other parts of that subject's EHR. Table 1 summarises these areas of context, and Table 2 maps these to the EN 13606 Reference Model.

Table 1 — List of context areas and elements

Context Area	Context Element	Description
Meaning	Meaning	Formal concept defining the EHR object to be instantiated at a given node in the EHR.
Subject of information	Subject of information	Rules for the permitted values of the subject of information; for example to specify that the subject of information may not be the patient, or shall be a genetic relative.
Act status	Act status	Rules for the permitted values of the act status; for example, to specify that the act status shall be a planned/ordered activity.
Temporal relationship	Temporal relationship	Specification of the temporal relationship of the information to the time of its recording e.g. former, ongoing, future.
Structure	Structure	The spatial structure to be used to represent (render) a data structure e.g. list, table, tree.
Observation time	Observation time	Rules for the permitted values of the observed/intended time (or time interval) for this observation set.
Link		Most links (13606 LINK, HL7 Act Relationship) are defined on an <i>ad hoc</i> basis within individual instances of EHR data. However, there may be times when particular kinds of clinical data shall always have certain links defined. For example, certain care acts might always need to reference a pre-existing consent document in the EHR, or a pre-existing clinical finding that justifies the activity.
	Nature	The kind of link (CEN nature, HL7 ActRelationship class code) that shall or may be composed
	Role	The role that the target plays in this link
	Follow_link	Rules for when the link is required to be followed when the source or target component is retrieved from an EHR system (CEN follow_link, HL7 separableInd).
Participation		If certain participants may or shall be defined in the EHR node instance.
	Function	The functional role that shall be held by the participant.
	Mandatory attestation	If this participant is required to attest the instance; this requirement does not specify if this node shall be individually attested or if it may be attested as part of a larger collection of EHR nodes, for example at the document level.
	Attestation reason	Specifies a fixed reason for the attestation; e.g. if the attestation is performing a particular legal function.

Table 2 — EN 13606 Reference Model – context profile

13606 RM class	Context Area	Corresponding 13606 RM attribute
FOLDER	Meaning	Meaning
	Link	LINK
COMPOSITION	Meaning	Meaning
	Link	LINK
	Participation	composer other_participations
SECTION	Meaning	Meaning
	Link	LINK
ENTRY	Meaning	Meaning
	Link	LINK
	Participation	other_participations
	Subject of information	subject_of_information
	Act status	act_status
CLUSTER	Meaning	Meaning
	Link	LINK
	Structure	structure_type
	Observation time	obs-time
ELEMENT	Meaning	Meaning
	Link	LINK
	Observation time	obs-time

7 Archetype Model

7.1 Introduction

7.1.1 General

The model is presented using a constrained form of UML diagrams, described below in a UML Profile.

7.1.2 UML Profile

The classes of the model together with their associations and inheritances are grouped into Packages, and these are shown in a number of separate diagrams below in this clause. Package boundaries are shown as blue lines with the name of the package, also shown in blue, in the top left box of the package outline.

Class boxes usually have 3 compartments:

The top compartment contains the Name of the class in upper case, and may also show the owning class in parentheses when the class belongs to a package other than the package that is the subject of the diagram. Some diagrams also show Class Constraints in the top compartment.

The second compartment, if present, contains Attributes, showing attribute name, attribute type, and multiplicity. Multiplicity may also be further qualified by the 'ordered' marker. Attribute names are shown in lower case. Attribute types are shown in Title case if the type is one of the basic types, and in Upper case if the type is another class.

The third compartment, if present, contains Operations, showing operation name, return type, and parameters passed. Operation names and types follow the same casing rules as attributes.

A Class box with only two compartments has Class name and Attributes: a Class box with only one compartment has just a Name.

Colour itself has no significance, but is used to enhance readability. Inheritance lines are shown in black, and association lines in maroon. Colour fills of class boxes are sometimes used to highlight particular groupings of Classes. Grey Class boxes are used to indicate that the Class details are shown on another diagram.

Associations between classes are always 'single ended' with the association name and multiplicity placed at the far end of the line. If a double-ended association is required, it is shown as two single ended associations, one in each direction, between the two classes. This restriction has been applied to make it possible to document each association automatically, as though it were an attribute of the near-end class. Navigation arrows are not used.

7.1.3 Detailed documentation of the model

The order of documentation is by package and, within package, by class.

Each class has a starting section showing the owning package, any inheritance, inner elements, and any internal model documentation, and is followed by up to four tabular sections for:

- a) Attributes
- b) Attributes derived from associations
- c) Operations
- d) Constraints

The associations themselves are shown in the diagrams using UML notation, but are documented as derived Associations using the following transformations:

Association far end name	becomes	Attribute name
Association far end class	becomes	Attribute type

Association Multiplicity	generates	Container Type	and	Attribute Optionality	Original Multiplicity
0..*		Set<far end CLASS>		0..1	0..*
0..* {ordered}		List<far end CLASS>		0..1	0..* {ordered}
1..*		Set<far end CLASS>		1	1..*
1..* {ordered}		List<far end CLASS>		1	1..* {ordered}
*		Set<far end CLASS>		0..1	*
0..1		Not a container		0..1	N/A
1		Not a container		1	N/A

7.1.4 Package structure

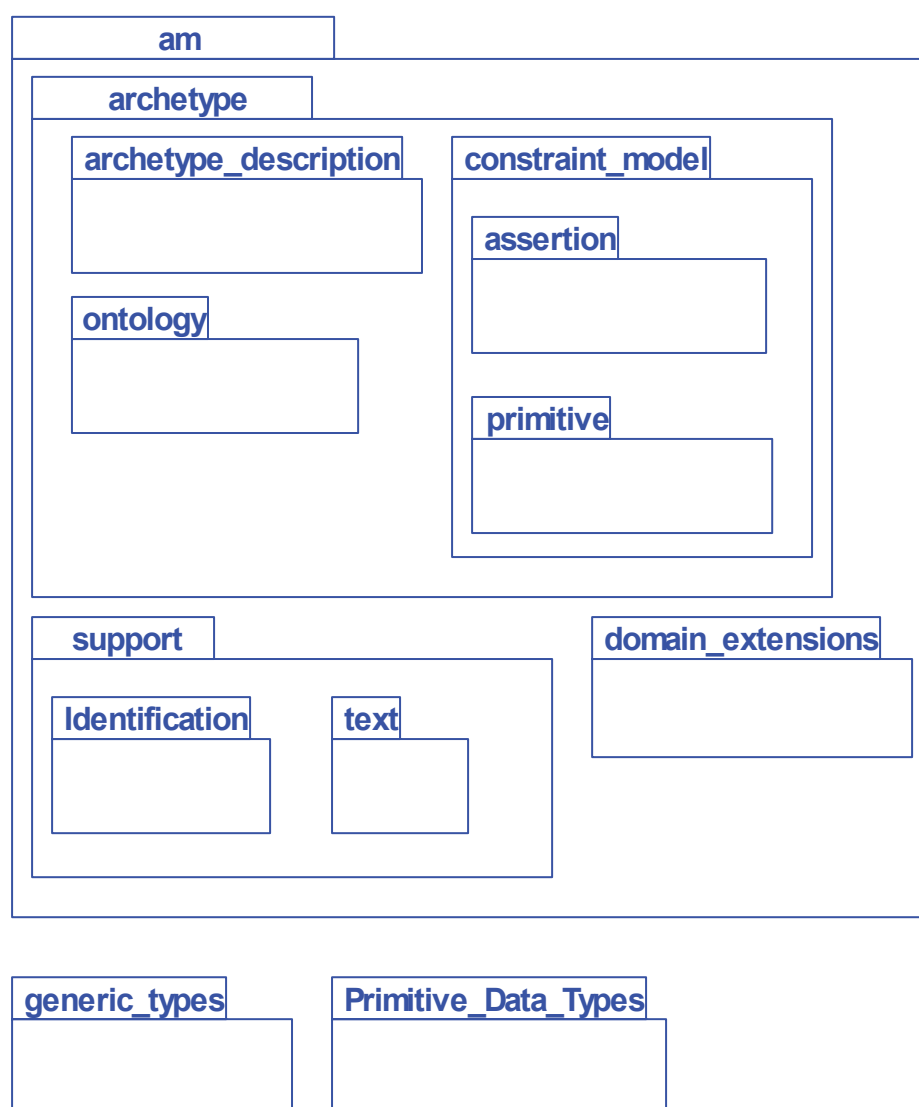


Figure 2 — Package structure

The overall Archetype Model shown in Figure 3 & Figure 4 defines the generic representation of archetypes for interoperability and communication purposes.

7.2 Overview

7.2.1 General

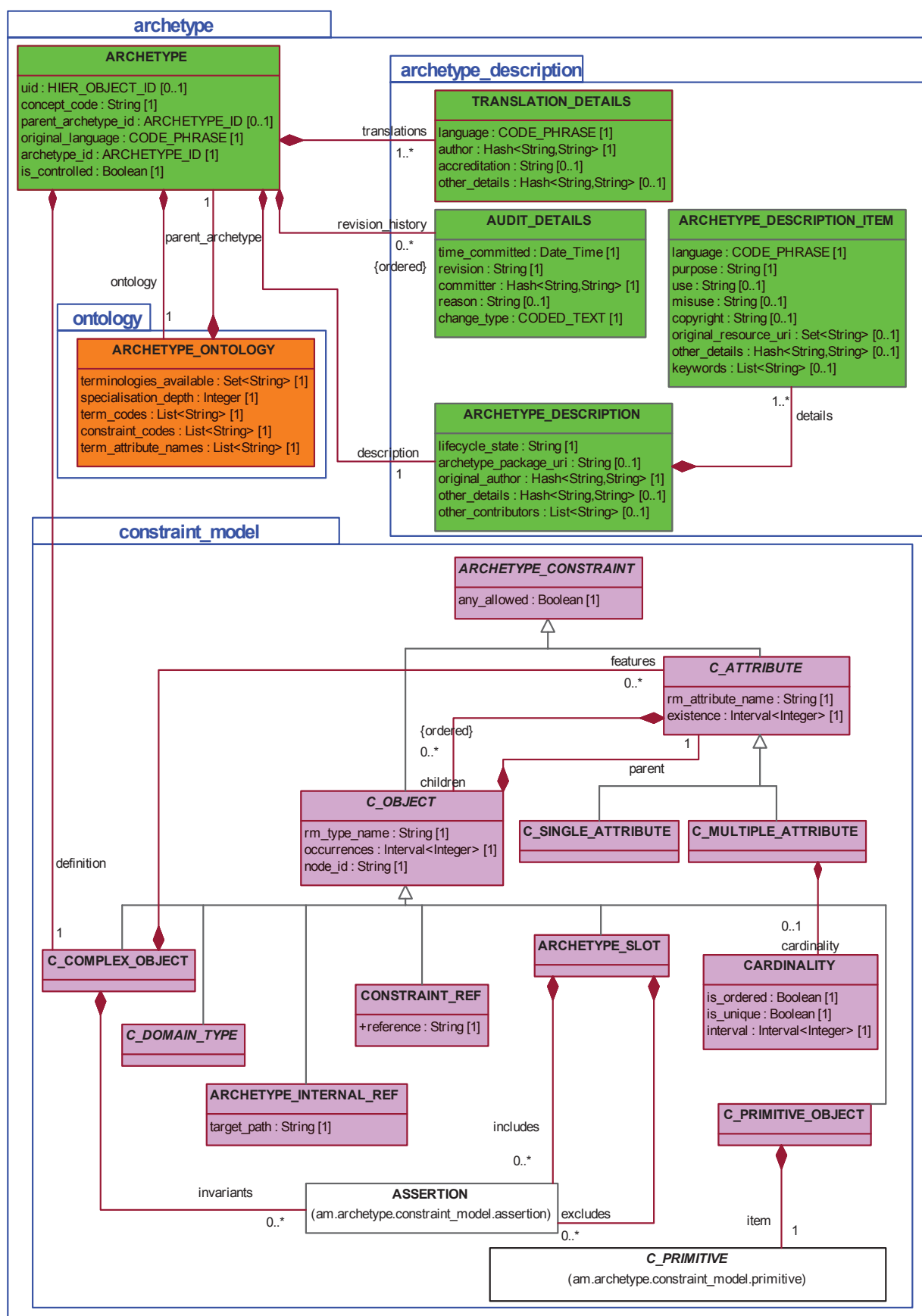


Figure 3 — Overview of the main part of the Archetype Model – Part 1

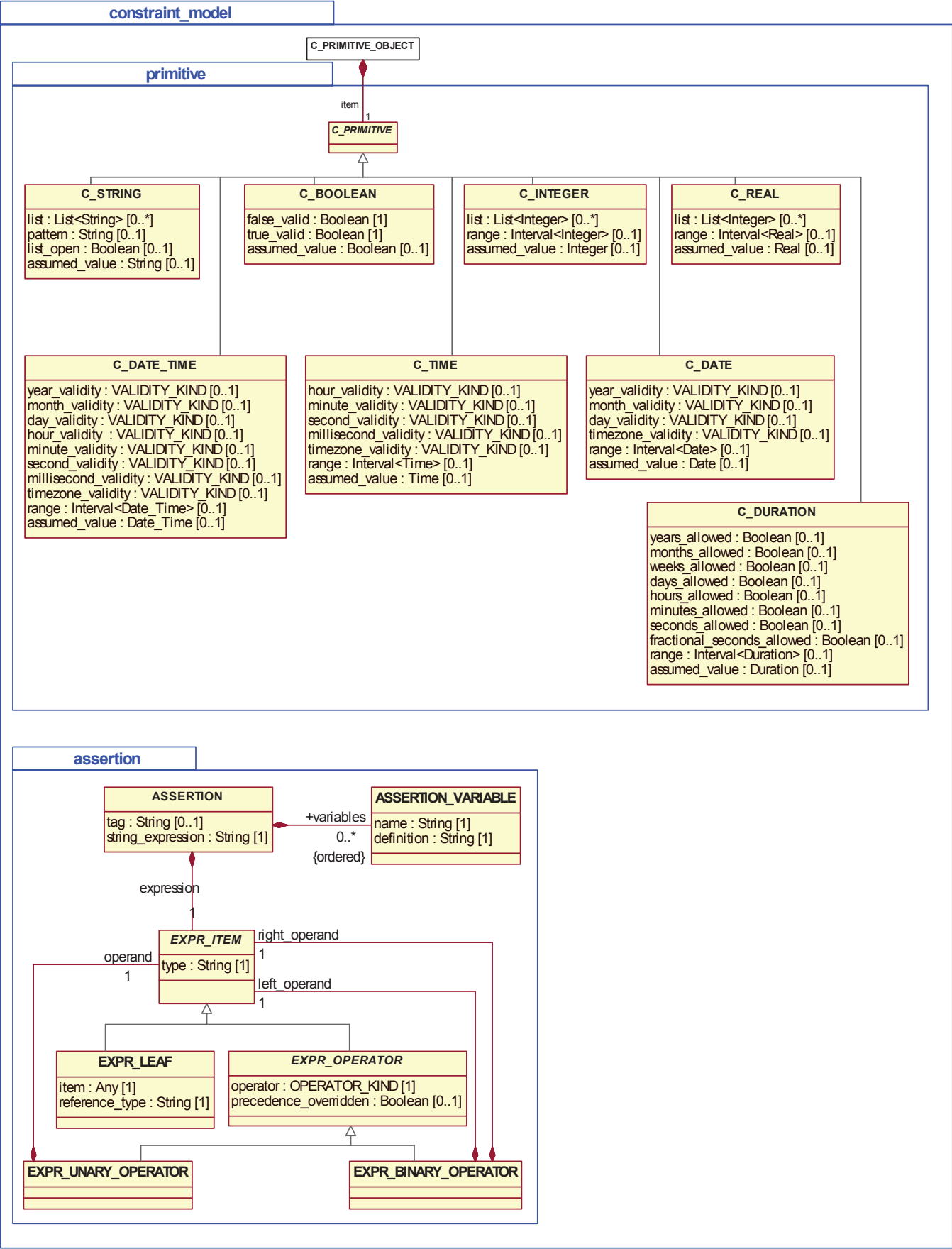


Figure 4 — Overview of the Archetype Model - Part 2

Documentation derived from the UML Model

NOTE The following table of hyperlinks to the packages and Classes of the model is presented as an aid to navigation around the model.

am

archetype

ARCHETYPE

archetype_description

ARCHETYPE_DESCRIPTION

ARCHETYPE_DESCRIPTION_ITEM

AUDIT_DETAILS

TRANSLATION_DETAILS

constraint_model

ARCHETYPE_CONSTRAINT

ARCHETYPE_INTERNAL_REF

ARCHETYPE_SLOT

C_ATTRIBUTE

C_COMPLEX_OBJECT

C_DOMAIN_TYPE

C_MULTIPLE_ATTRIBUTE

C_OBJECT

C_PRIMITIVE_OBJECT

C_SINGLE_ATTRIBUTE

CARDINALITY

CONSTRAINT_REF

assertion

ASSERTION

ASSERTION_VARIABLE

EXPR_BINARY_OPERATOR

EXPR_ITEM

EXPR_LEAF

EXPR_OPERATOR

EXPR_UNARY_OPERATOR

primitive

C_BOOLEAN

C_DATE

C_DATE_TIME

C_DURATION

C_INTEGER

C_PRIMITIVE

C_REAL

C_STRING

C_TIME

ontology

ARCHETYPE_ontology

ARCHETYPE TERM

domain_extensions

C_CODED_TEXT

C_ORDINAL

C_QUANTITY

C_QUANTITY_ITEM

ORDINAL

support

OPERATOR_KIND

VALIDITY_KIND

Identification

ARCHETYPE_ID

HIER_OBJECT_ID

OBJECT_ID

TERMINOLOGY_ID

text

CODE_PHRASE

CODED_TEXT

TERM_MAPPING

TEXT

7.2.2 Package :: am

Inner Elements	
Name	Type
<u>archetype</u>	Package
<u>domain_extensions</u>	Package
<u>support</u>	Package

7.3 The Archetype Package

7.3.1 General

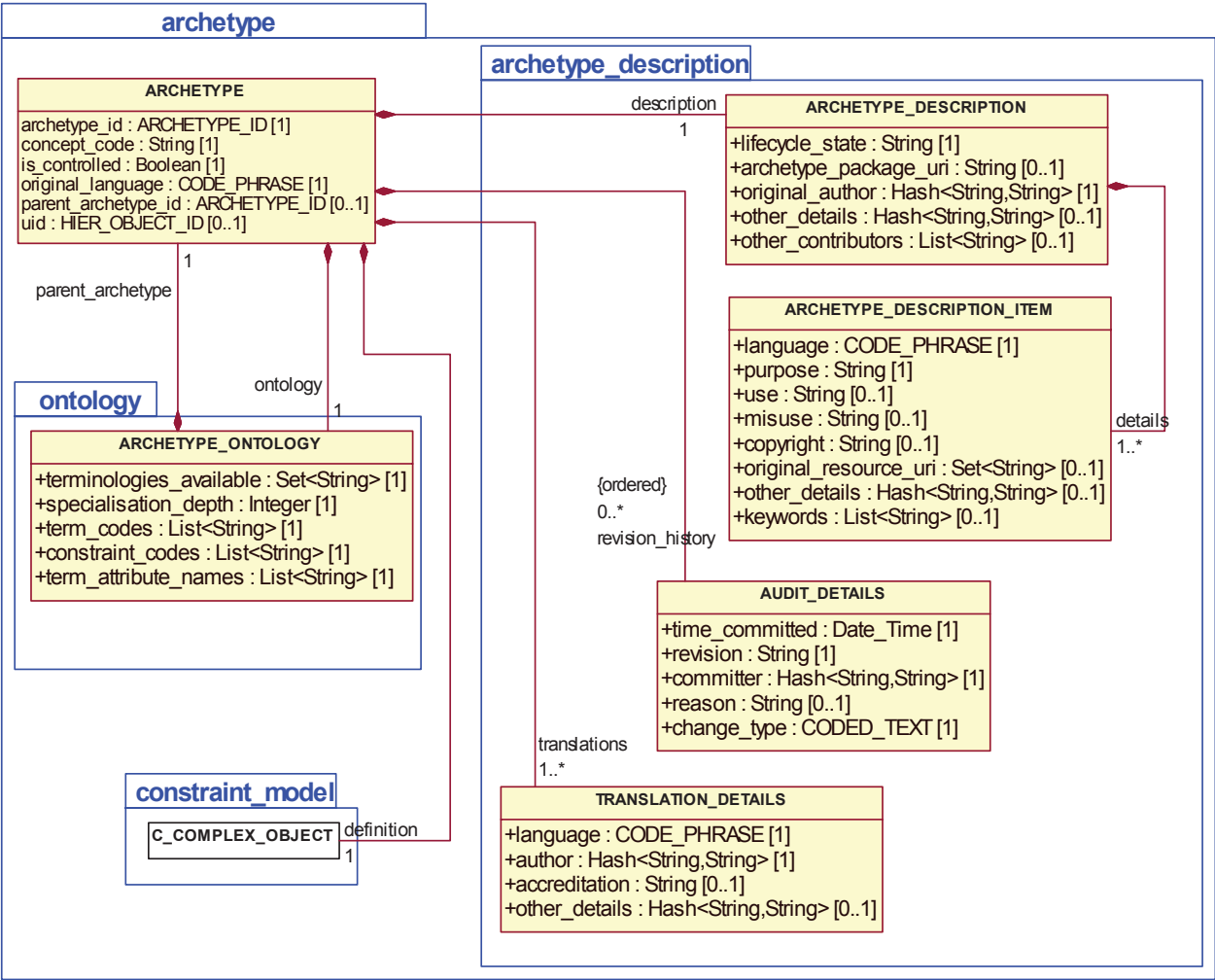


Figure 5 — Archetype Package

7.3.2 Package :: archetype

Inner Elements	
Name	Type
ARCHETYPE	Class
archetype_description	Package
constraint_model	Package
ontology	Package

Package: archetype
Class **Archetype**

The main class of the Archetype package.

Attributes			
Signature	Optionality	Multiplicity	Documentation
archetype_id : <u>ARCHETYPE_ID</u>	1	--	Multi-axial identifier of this archetype in archetype space.
concept_code : <u>String</u>	1	--	The normative meaning of the archetype as a whole.
is_controlled : <u>Boolean</u>	1	--	True if this archetype is under change control in which case revision history is created.
original_language : <u>CODE_PHRASE</u>	1	--	Language in which this archetype was initially authored.
parent_archetype_id : <u>ARCHETYPE_ID</u>	0..1	--	Identifier of the specialisation parent of this archetype.
uid : <u>HIER_OBJECT_ID</u>	0..1	--	OID identifier of this archetype.

Attributes from Associations			
Signature	Optionality	Multiplicity	Documentation
revision_history : List<AUDIT_DETAILS>	0..1	0..* ordered	The revision history of the archetype; only required if is_controlled = True
description : ARCHETYPE_DESCRIPTION	1	--	Description and lifecycle information of the archetype.
ontology : ARCHETYPE_ONTOLOGY	1	--	The ontology of the archetype.
definition : C_COMPLEX_OBJECT	1	--	Root node of this archetype
translations : Set<TRANSLATION_DETAILS>	1	1..*	List of details for each natural translation included in this archetype.

Constraints	
Name	Expression
revision_history_validity	inv: is_controlled implies (revision_history <> Void and revision_history.is_empty)
archetype_id_validity	inv: archetype_id <> Void
description_exists	inv: description <> Void
ontology_exists	inv: ontology <> Void
definition_exists	inv: definition <> Void
uid_validity	inv: uid <> Void implies not uid.is_empty
original_language_valid	inv: original_language <>void and translations. language <>Void and terminology_service.code_set('languages') .has(original_language)
has_parent	post: is_specialised implies parent_archetype_id <>Void

7.4 The Archetype Description Package

7.4.1 General

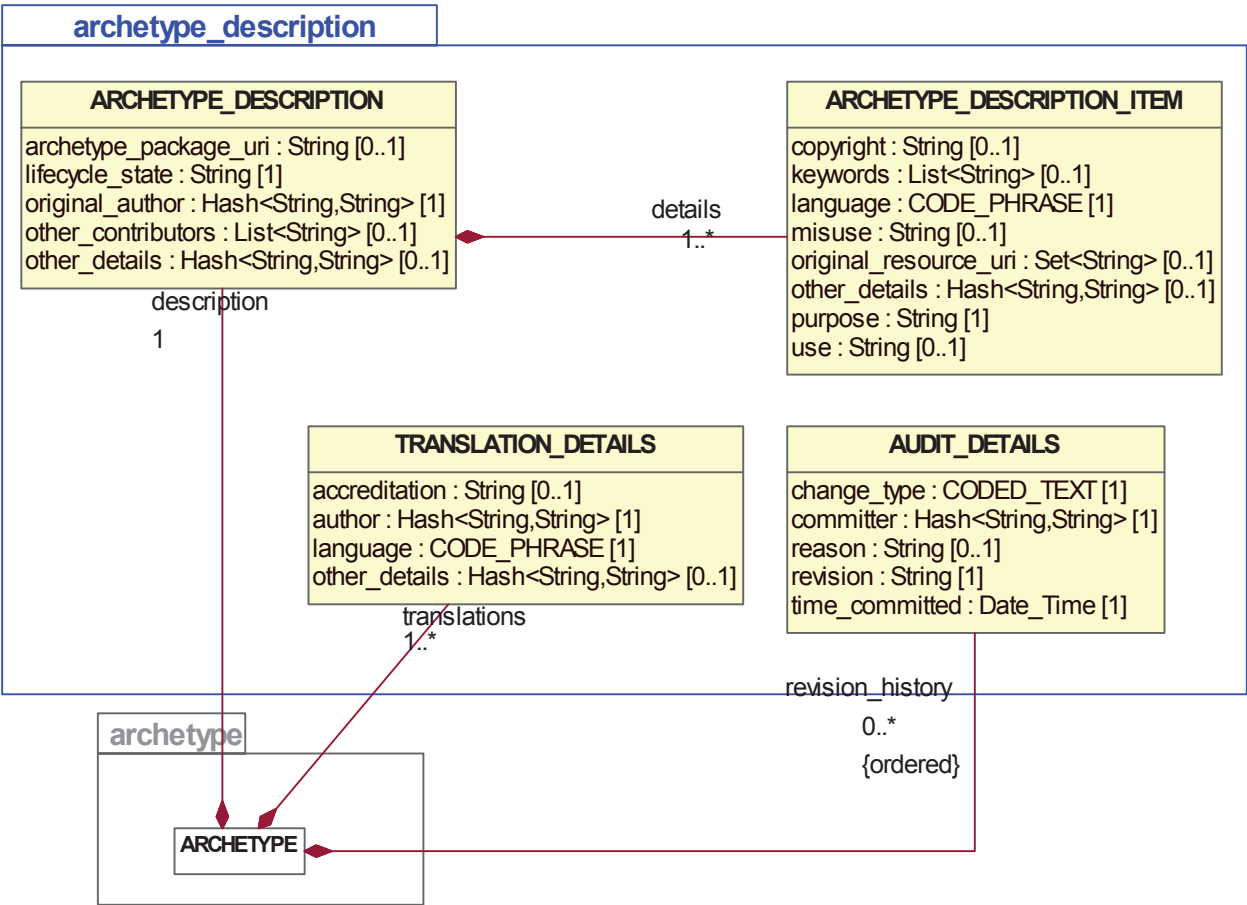


Figure 6 — Archetype Description Package

7.4.2 Package :: archetype_description

The 'meta-data' of an archetype.

Inner Elements	
Name	Type
ARCHETYPE_DESCRIPTION	Class
ARCHETYPE_DESCRIPTION_ITEM	Class
AUDIT_DETAILS	Class
TRANSLATION_DETAILS	Class

Package: archetype_description
Class **AUDIT_DETAILS**

Attributes			
Signature	Optionality	Multiplicity	Documentation
change_type : <u>CODED_TEXT</u>	1	--	Type of change.
committer : <u>Hash<String,String></u>	1	--	Identification details of the author of the main content of this archetype, expressed as a list of name-value pairs.
reason : <u>String</u>	0..1	--	Natural language reason for change.
revision : <u>String</u>	1	--	Revision corresponding to this change.
time_committed : <u>Date_Time</u>	1	--	Date/time of this change

Constraints	
Name	Expression
committer_validity	inv: committer <> Void and not committer.is_empty
committer_organisation_validity	inv: committer_organisation <> Void implies not committer_organisation.is_empty
time_committed_exists	inv: time_committed <> Void
reason_valid	inv: reason <> Void implies not reason.is_empty
revision_valid	inv: revision <> Void and not revision.is_empty
change_type_exists	inv: change_type <> Void and terminology_service.terminology('openehr').codes_for_group_name('audit_change_type', 'en').has(change_type.defining_code)

Package: archetype_description
Class ARCHETYPE_DESCRIPTION

Defines the descriptive meta-data of an archetype.

Attributes			
Signature	Optionality	Multiplicity	Documentation
archetype_package_uri : <u>String</u>	0..1	--	URI of package to which this archetype belongs.
lifecycle_state : <u>String</u>	1	--	Lifecycle state of the archetype: initial, submitted, experimental, awaiting_approval, approved, superseded, obsolete.
original_author : <u>Hash<String,String></u>	1	--	Original author of this archetype, expressed as a list of name-value pairs.
other_contributors : <u>List<String></u>	0..1	--	Names of other contributors to the archetype
other_details : <u>Hash<String,String></u>	0..1	--	Additional non language-sensitive archetype meta-data, as a list of name/value pairs.

Attributes from Associations			
Signature	Optionality	Multiplicity	Documentation
details : Set<ARCHETYPE_DESCRIPTION_ITEM>	1	1..*	The descriptive meta-data of an archetype.

Constraints	
Name	Expression
original_author_validity	inv: original_author <>Void and not original_author.is_empty
details_exists	inv: details <> Void and not details.is_empty
original_author_organisation_validity	inv: original_author_organisation <> Void implies not original_author_organisation.is_empty
language_validity	inv: details->for_all(d parent_archetype.languages_available.has(d.language))
parent_archetype_valid	inv: parent_archetype <> Void and parent_archetype.description = Current

Package: archetype_description

Class ARCHETYPE_DESCRIPTION_ITEM

Language-specific detail of archetype description. When an archetype is trans-lated for use in another language environment, eachARCHETYPE_DESCRIPTION_ITEM needs to be copied and translated into the newlanguage.

Attributes			
Signature	Optionality	Multiplicity	Documentation
copyright : <u>String</u>	0..1	--	Optional copyright statement for the archetype as a knowledge resource.
keywords : <u>List<String></u>	0..1	--	Keywords by which this Archetype may be referenced.
language : <u>CODE_PHRASE</u>	1	--	The localised language in which the items in this description item are written.
misuse : <u>String</u>	0..1	--	Description of any contexts in which it should not be used.
original_resource_uri : <u>Set<String></u>	0..1	--	URI of original clinical document(s) or description of which archetype is a formalisation, in the language of this description item.
other_details : <u>Hash<String,String></u>	0..1	--	Additional language-sensitive archetype meta-data, as a list of name/value pairs.
purpose : <u>String</u>	1	--	Purpose of the archetype.
use : <u>String</u>	0..1	--	Description of the uses of the archetype, i.e. contexts in which it could be used.