



REFERENCE MODEL

The *openEHR* Support Information Model

Editors: {T Beale, S Heard}¹, {D Kalra, D Lloyd}²

Revision: 1.1

Pages: 33

-
1. Ocean Informatics Australia
 2. Centre for Health Informatics and Multi-professional Education, University College London

© 2003 The *openEHR* Foundation

The *openEHR* foundation

is an independent, non-profit community, facilitating the creation and 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
Patrons	To Be Announced

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

Copyright Notice

© Copyright *openEHR* Foundation 2001 - 2004

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 2001-2004. 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
1.1	Populated Terminology and code_set codes.	S Heard	11 Mar 2004
1.0	CR-000091. Correct anomalies in use of CODE_PHRASE and DV_CODED_TEXT. Add simple terminology service interface. CR-000095. Remove <i>property</i> attribute from Quantity package. Add simple measurement interface. Formally validated using ISE Eiffel 5.4.	T Beale DSTC, S Heard	09 Mar 2004
0.9.9	CR-000063. ATTESTATION should have a status attribute.	D Kalra	13 Feb 2004
0.9.8	CR-000068. Correct errors in INTERVAL class.	T Beale	20 Dec 2003
0.9.7	CR-000032. Basic numeric type assumptions need to be stated CR-000041. Visually differentiate primitive types in openEHR documents. CR-000043. Move External package to Common RM and rename to Identification (incorporates CR-000036 - Add HIER_OBJECT_ID class, make OBJECT_ID class abstract.)	DSTC, D Lloyd, T Beale	09 Oct 2003
0.9.6	CR-000013. Rename key classes. Based on CEN ENV13606. CR-000038. Remove <i>archetype_originator</i> from multi-axial archetype id. CR-000039. Change archetype_id section separator from ':' to '-'	T Beale	18 Sep 2003
0.9.5	CR-000036. Add HIER_OBJECT_ID class, make OBJECT_ID class abstract.	T Beale	16 Aug 2003
0.9.4	CR-000022. Code TERM_MAPPING.purpose.	G Grieve	20 Jun 2003
0.9.3	CR-000007. Added forgotten terminologies for Subject_relationships and Provider_functions.	T Beale	11 Apr 2003
0.9.2	Detailed review by Ocean, DSTC, Grahame Grieve. Updated valid characters in OBJECT_ID.namespace.	G Grieve	25 Mar 2003
0.9.1	Added specification for BOOLEAN type. Corrected minor error in ISO 639 standard strings - now conformant to TERMINOLOGY_ID. OBJECT_ID.version_id now optional. Improved document structure.	T Beale	18 Mar 2003
0.9	Initial Writing. Taken from Data types and Common Reference Models. Formally validated using ISE Eiffel 5.2.	T Beale	25 Feb 2003

Acknowledgements

The work reported in this paper has been funded in by a number of organisations, including The University College, London and Ocean Informatics, Australia.

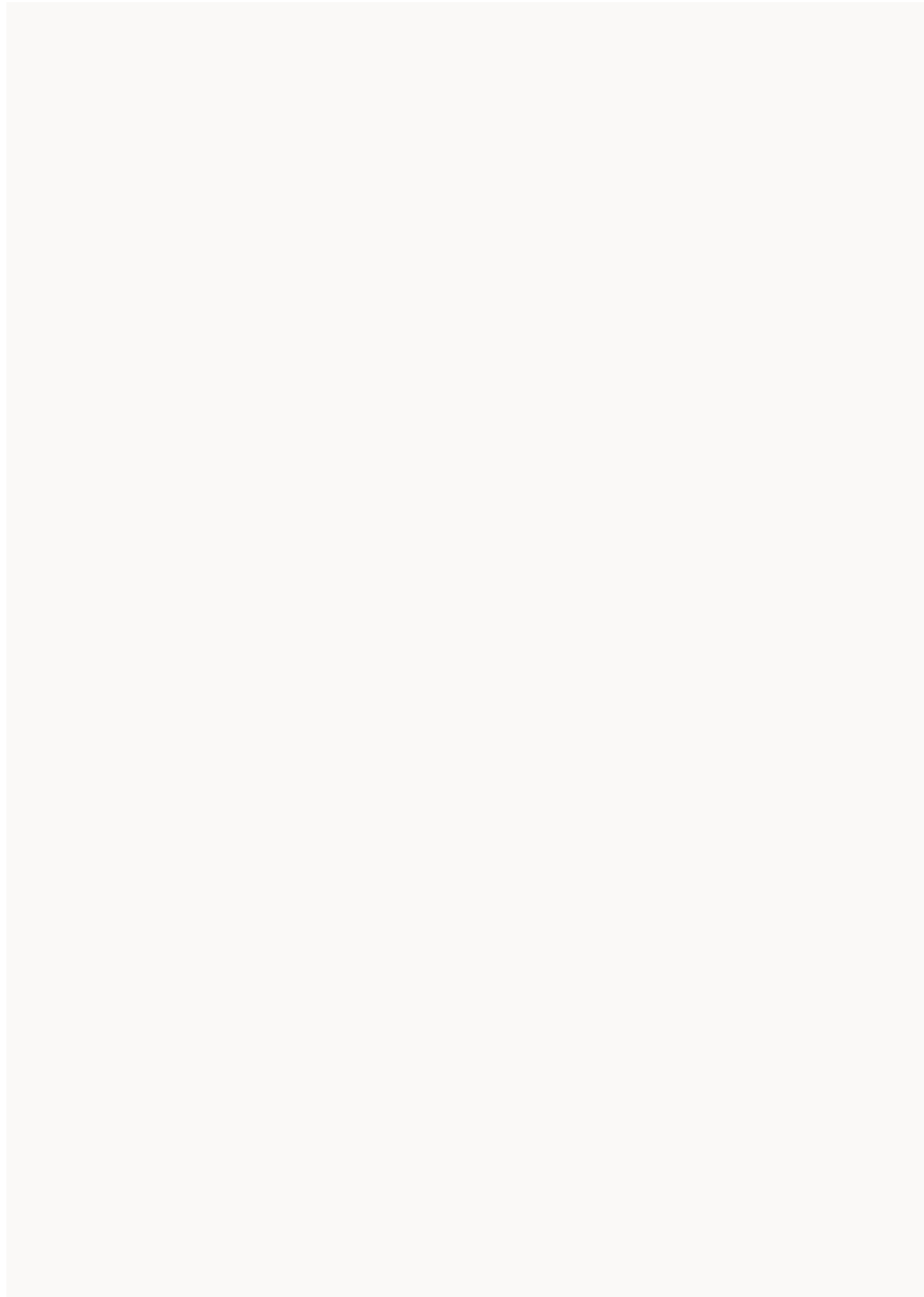


Table of Contents

1	Introduction	7
1.1	Purpose.....	7
1.2	Status.....	7
1.3	Peer review	7
2	Overview	9
3	Assumed Types	10
3.1	Overview	10
3.2	Type Descriptions	11
3.2.1	Any.....	11
3.2.2	Boolean Type	11
3.2.3	Numeric Type.....	13
3.2.4	Comparable Type	14
3.2.5	Interval <T> Type	14
3.2.6	String.....	14
3.2.6.1	UNICODE	15
3.2.7	Container <T>.....	15
4	RM.SUPPORT.TERMINOLOGY Package	16
4.1	Overview.....	16
4.2	Service Interface	16
4.2.1	Class Definitions.....	17
4.2.1.1	TERMINOLOGY_SERVICE_ACCESS Class	17
4.2.1.2	TERMINOLOGY_INTERFACE Class	18
4.2.1.3	CODE_SET_INTERFACE Class	18
4.3	Code Sets	18
4.3.1	Languages	18
4.3.2	Countries	19
4.3.3	Character Sets	19
4.3.4	Media Types.....	19
4.3.5	Compression algorithms	20
4.3.6	Integrity check algorithms	20
4.4	Vocabularies and Terminologies	21
4.4.1	Measurable Properties.....	21
4.4.2	Term Mapping Purpose.....	21
4.4.3	Participation Mode.....	21
4.4.4	Setting	23
4.4.5	Attestation Status	24
4.4.6	Version Lifecycle State	24
4.4.7	Audit Change Type	24
4.4.8	Related Party relationship.....	26
4.4.9	Event Math Function	27
4.4.10	Null Flavours	28
5	RM.SUPPORT.MEASUREMENT Package.....	29
5.1	Overview.....	29
5.2	Service Interface	29
5.2.1	Class Definitions.....	29
5.2.1.1	MEASUREMENT_SERVICE_ACCESS Class	29

6 RM.SUPPORT.DEFINITION Package 31
6.1 Constants 31

1 Introduction

1.1 Purpose

This document describes the *openEHR* Support Reference Model, which is used in common by all *openEHR* Reference Models. The intended audience includes:

- Standards bodies producing health informatics standards;
- Software development organisations developing EHR systems;
- Academic groups studying the EHR;
- The open source healthcare community;

1.2 Status

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

Currently the UML diagrams are hand-produced. None of the existing tools (e.g. Rose, Objecteering), includes sufficient support of UML or has good enough visual quality to use here. However, UML tools are constantly under investigation, and this situation may change in the future.

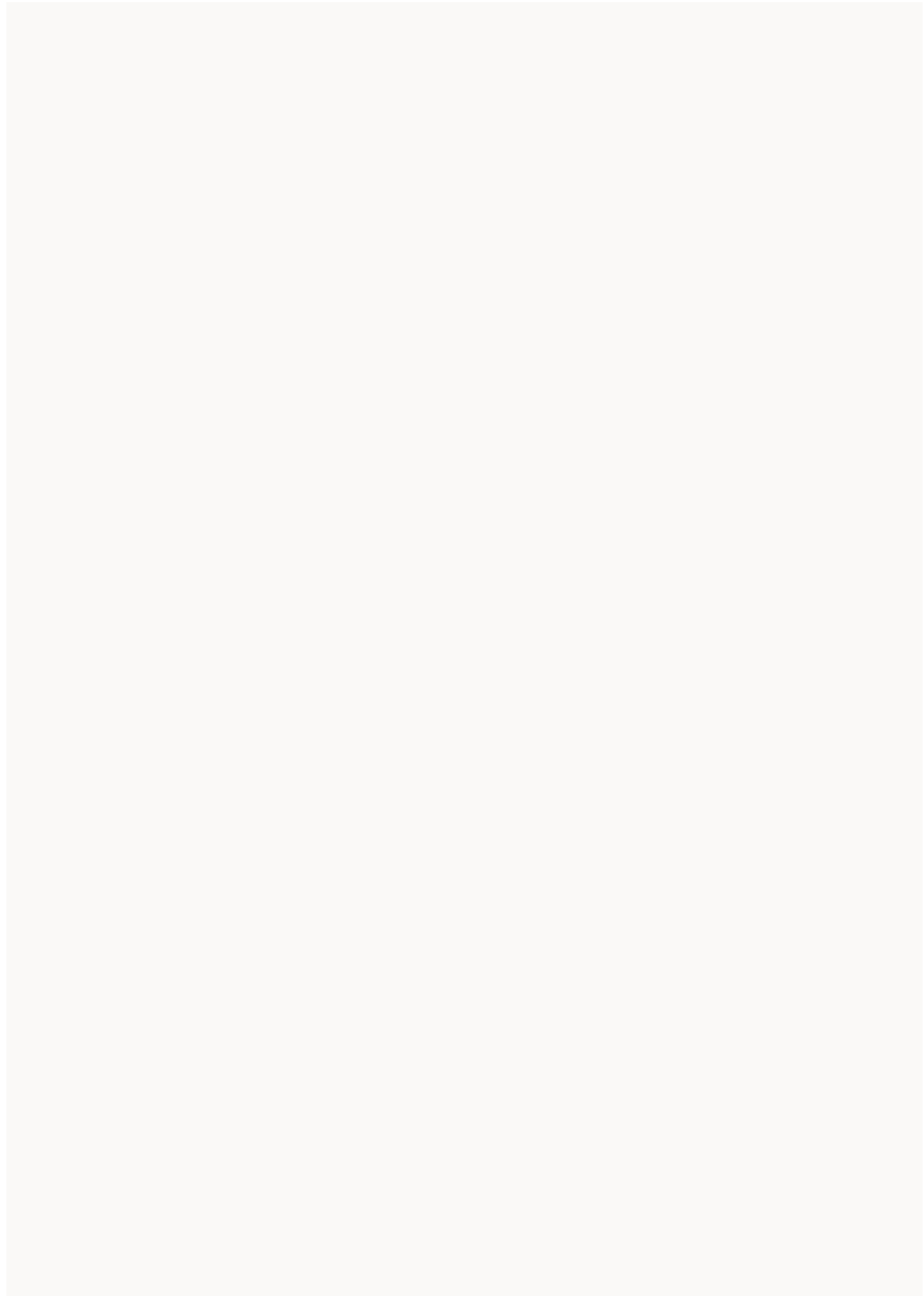
The latest version of this document can be found in PDF and HTML formats at http://www.openEHR.org/Doc_html/Model/Reference/support_rm.htm. New versions are announced on openehr-announce@openehr.org.

1.3 Peer review

Areas where more analysis or explanation is required are indicated with “to be continued” paragraphs like the following:

To Be Continued: more work required

Reviewers are encouraged to comment on and/or advise on these paragraphs as well as the main content. Please send requests for information to info@openEHR.org. Feedback should preferably be discussed on one of the appropriate mailing lists, openehr-technical@openehr.org or openehr-clinical@openehr.org.



2 Overview

The Support Reference Model comprises types which are used throughout other *openEHR* models, but are defined elsewhere, either by standards organisations or which are accepted *de facto* standards. The package structure is illustrated in FIGURE 1.

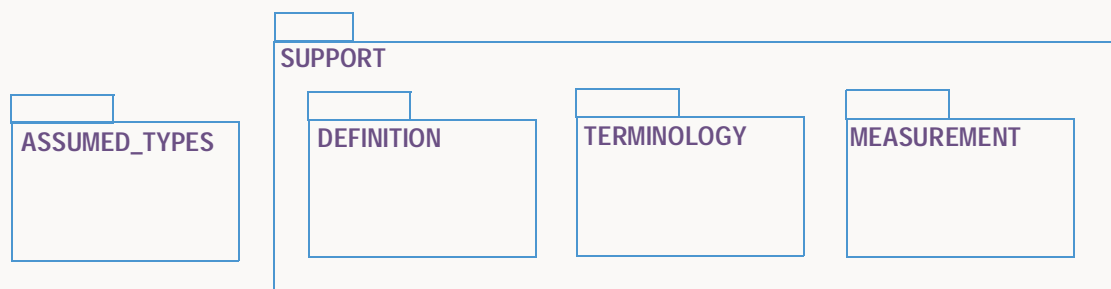


FIGURE 1 RM.SUPPORT and ASSUMED_TYPES Packages

The three Support packages define the semantics respectively for constants, terms and scientific measurement which are assumed by the rest of the *openEHR* specifications.

3 Assumed Types

3.1 Overview

This section describes types assumed by all *openEHR* models. The following types are assumed to exist and to have very similar definitions in implementation technologies. The set of types chosen here is based on a lowest common denominator set from well-known interoperability formalisms, including OMG IDL, W3C XML-schema, and well-known programming languages.

Type	Description
Character	represents any 8-bit character
String	represents strings of any encoding, including unicode
Boolean	represents logical True/False values; usually physically represented as an integer, but need not be
Integer	represents 32-bit integers
Integer_64	represents 64-bit integers
Real	represents single-precision floating point real numbers
Double	type which represents double-precision floating point real numbers
Array<T>	physical container of items indexed by number
List<T>	container of items, implied order, non-unique membership
Set<T>	container of items, no order, unique membership
Bag<T>	container of items, no order, non-unique membership

FIGURE 2 illustrates inheritance relationships between the primitive types, and adds three abstract types, which are not themselves needed in *openEHR*, but serve to clarify the relationships between clinical types. Note that these relationships need not literally be the case; for example, the `String` type might also be `Comparable`, but this assumption is not required for *openEHR* to work, so it is not included here. By the same token, the type called here `Comparable` might actually be a type called `Numeric`, or similar in an actual type system. Likewise, the numeric types might not inherit from the types `Numeric` and `Comparable` in a concrete type system - this is just a convenient shorthand to show that they are required to implement a minimum of operators such as '>', '+'.

A small number of assumptions are made about these types within the main *openEHR* specifications. These assumptions are documented below in terms of interface definitions. Each of these definitions contains *only the assumptions required for the given type* - it is not a complete interface definition. The name and semantics of any function used here for an assumed type might not be identical to those found in some implementation technologies, but should be very close. Any mapping required should be stated in the relevant ITS.

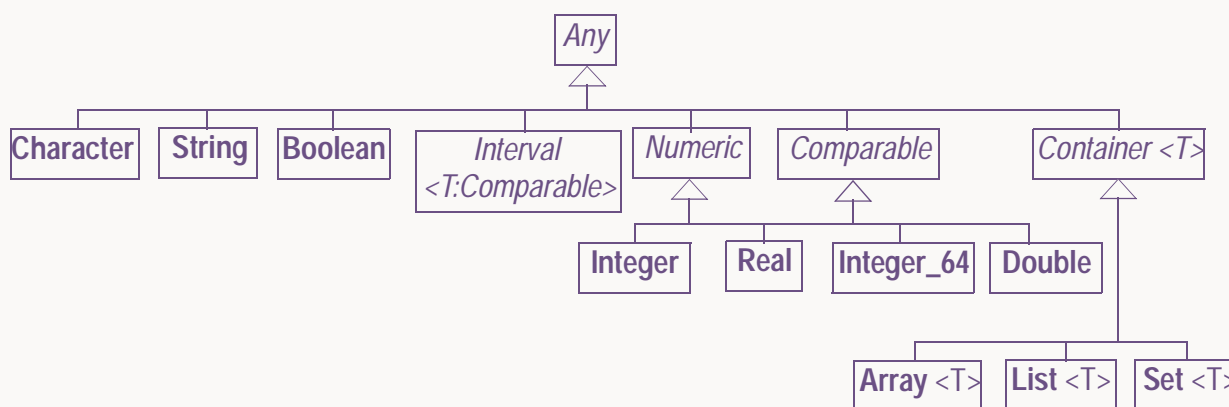


FIGURE 2 UML Diagram of Assumed Primitive Types

3.2 Type Descriptions

The following specifications are currently adapted from the Eiffel BASE libraries, published by ISE¹. They may be replaced by more normative specifications in the future, pending availability of the latter.

3.2.1 Any

INTERFACE	<i>Any (abstract)</i>	
Description	All types	
Abstract	Signature	Meaning
	<code>is_equal(other: Any): Boolean</code>	Value equality
Functions	Signature	Meaning
	<code>infix '=' (other: Any): Boolean</code>	Reference equality
Invariants		

3.2.2 Boolean Type

INTERFACE	Boolean	
Purpose	Boolean type.	
Abstract	Signature	Meaning

1. <http://www.eiffel.com>

INTERFACE	Boolean	
	<p>infix "and" (other: <i>like</i> Current): Boolean</p> <p><i>require</i> <i>other_exists</i>: other /= void</p> <p><i>ensure</i> <i>de_morgan</i>: Result = not (not Current or not other)</p> <p><i>commutative</i>: Result = (other and Current)</p> <p><i>consistent_with_semi_strict</i>: Result implies (Current and then other)</p>	<p>Logical conjunction</p>
	<p>infix "and then" (other: <i>like</i> Current): Boolean</p> <p><i>require</i> <i>other_exists</i>: other /= void</p> <p><i>ensure</i> <i>de_morgan</i>: Result = not (not Current or else not other)</p>	<p>Boolean semi-strict conjunction with <i>other</i></p>
	<p>infix "or" (other: <i>like</i> Current): Boolean</p> <p><i>require</i> <i>other_exists</i>: other /= void</p> <p><i>ensure</i> <i>de_morgan</i>: Result = not (not Current and not other)</p> <p><i>commutative</i>: Result = (other or Current)</p> <p><i>consistent_with_semi_strict</i>: Result implies (Current or else other)</p>	<p>Boolean disjunction with <i>other</i></p>
	<p>infix "or else" (other: <i>like</i> Current): Boolean</p> <p><i>require</i> <i>other_exists</i>: other /= void</p> <p><i>ensure</i> <i>de_morgan</i>: Result = not (not Current and then not other)</p>	<p>Boolean semi-strict disjunction with <i>other</i></p>
	<p>infix "xor" (other: like Current): Boolean</p> <p><i>require</i> <i>other_exists</i>: other /= void</p> <p><i>ensure</i> <i>definition</i>: Result = ((Current or other) and not (Current and other))</p>	<p>Boolean exclusive or with <i>other</i></p>

INTERFACE	Boolean	
	infix "implies" (other: <i>like</i> Current): Boolean require <i>other_exists</i> : other /= void ensure <i>definition</i> : Result = (not Current or else other)	Boolean implication of `other' (semi-strict)
Invariants	<i>involution_negation</i> : is_equal (not (not Current)) <i>non_contradiction</i> : not (Current and (not Current)) <i>completeness</i> : Current or else (not Current)	

3.2.3 Numeric Type

INTERFACE	Numeric (abstract)	
Purpose	Abstract notional parent class of numeric types, which are types which have various arithmetic operators defined.	
Abstract	Signature	Meaning
	infix "*" (other: like Current): like Current require <i>other_exists</i> : other /= void ensure <i>result_exists</i> : Result /= void	Product by `other'. Actual type of result depends on arithmetic balancing rules.
	infix "+" (other: like Current): <i>like</i> Current require <i>other_exists</i> : other /= void ensure <i>result_exists</i> : Result /= void <i>commutative</i> : equal (Result, other + Current)	Sum with `other' (commutative). Actual type of result depends on arithmetic balancing rules.
	infix "-" (other: like Current): like Current require <i>other_exists</i> : other /= void ensure <i>result_exists</i> : Result /= void	Result of subtracting `other'. Actual type of result depends on arithmetic balancing rules.
Invariants		

3.2.4 Comparable Type

INTERFACE	<i>Comparable (abstract)</i>	
Purpose	Abstract notional parent class of comparable types, which are types which have the '<' defined.	
Abstract	Signature	Meaning
	infix '<' (other: NUMERIC): Boolean	Arithmetic comparison. In conjunction with '=', enables the definition of the operators '>', '>=', '<=', '<>'. In real type systems, this operator might be defined on another class for comparability.
Invariants		

3.2.5 Interval <T> Type

INTERFACE	<i>Interval <T:Comparable></i>	
Purpose	Interval of comparable items.	
Attributes	Signature	Meaning
	lower: T	lower bound
	upper: T	upper bound
	lower_unbounded: Boolean	lower boundary open (i.e. = -infinity)
	upper_unbounded: Boolean	upper boundary open (i.e. = +infinity)
Functions	Signature	Meaning
	has(e:T): Boolean	True if (lower >= e and e <= upper)
Invariants	<i>Limits_consistent:</i> (not upper_unbounded and not lower_unbounded) <i>implies</i> lower <= upper <i>Limits_comparable:</i> (not upper_unbounded and not lower_unbounded) <i>implies</i> lower.strictly_comparable_to(upper)	

3.2.6 String

INTERFACE	<i>String</i>
Description	String items

INTERFACE	String	
Functions	Signature	Meaning
	infix '+' (other: String): String	Concatenation operator - causes 'other' to be appended to this string
	+is_equal (other: String): Boolean	Value comparison function - implementation for strings
	is_empty : Boolean	True if string is empty, i.e. equal to "".
Invariants		

3.2.6.1 UNICODE

It is assumed in the *openEHR* specifications that Unicode is supported by the type `String`. Unicode is needed for all Asian, Arabic and other script languages, for both data values (particularly plain text and coded text) and for many predefined string attributes of the classes in the *openEHR* Reference Model. It encompasses all existing character sets.

3.2.7 Container <T>

INTERFACE	<i>Container <T> (abstract)</i>	
Description	Abstract parent of of the container types <code>List<T></code> , <code>Set<T></code> , <code>Bag<T></code> , <code>Array<T></code> .	
Functions	Signature	Meaning
	has (an_item: T): Boolean	Test for membership
	count : Integer	Number of items
Invariants		

4 RM.SUPPORT.TERMINOLOGY Package

4.1 Overview

This section describes the *openEHR* terminology and code sets which provide values for the dozen or so structural attributes in the *openEHR* Reference Model, along with a simple way of accessing them. There are two types of coded terms used. The first are ‘proper’ coded terms, where each code is a concept identifier, for which there can be a rubric and description in multiple languages. In other words, they way of ‘saying’ the concept is dependent on the language one is working in. Most clinical terminologies are in this category, e.g. ICD10, ICPC. Terminologies in this category are modelled in *openEHR* by the `TERMINOLOGY` class, and by terms expressed as instances the `DV_CODED_TEXT` class, each of which has as an attribute a defining `CODE_PHRASE` - the actual code.

The second category is codes which are self-defining, and which do not have separate rubrics. The ISO country and language codes are examples of this, as are code groups for such concepts as ‘integrity check algorithm names’. This category is modelled in *openEHR* by the `CODE_SET` which is made up of `CODE_PHRASES`.

The `TERMINOLOGY` and `CODE_SET` classes are defined below in a simple terminology interface, while the `DV_CODED_TEXT` and `CODE_PHRASE` types are defined in the *openEHR* Data Types Information Model.

Both code set definitions and terminology groups provide mappings to other recognised terminologies or vocabularies. Given that the attributes defined here are mostly structural attributes (i.e. pre-defined in the *openEHR* Reference Model), mappings tend to be to terms in vocabularies defined by standards organisations such as CEN and HL7, rather than large clinical vocabularies such as ICD10 (WHO). *OpenEHR* does not specify the use of these vocabularies

4.2 Service Interface

A simple terminology service interface is defined according to FIGURE 3, enabling *openEHR* terms to be referenced formally from within the Reference Model.

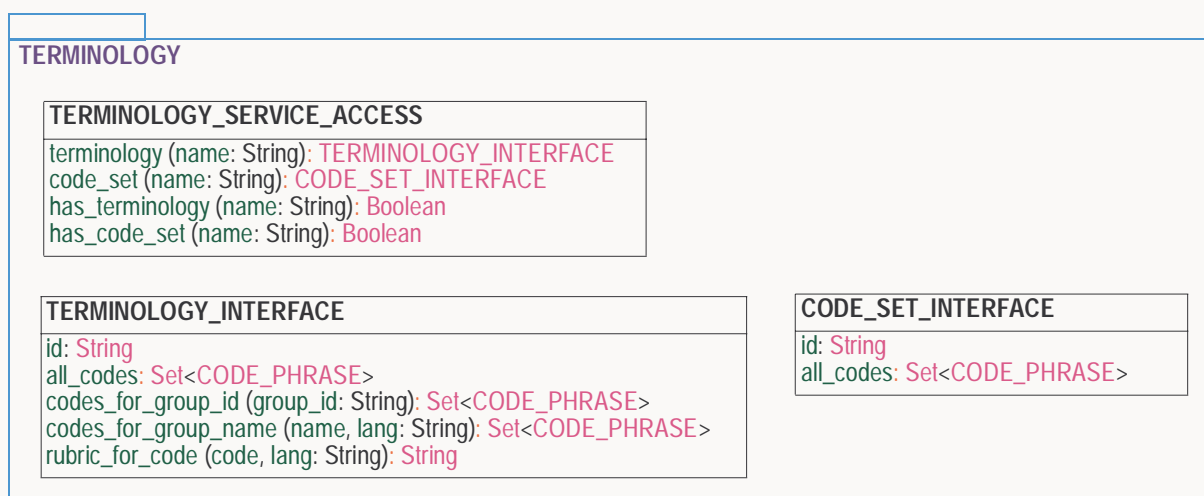


FIGURE 3 Simple Terminology Interface

Structural attributes in the Reference Model, such as `FEEDER_AUDIT.change_type` are defined by an invariant in the enclosing class, such as the following:

Change_type_valid: terminology(“openehr”).codes_for_group_name(“audit change type”, “en”).has(change_type.defining_code)

This is a formal way of saying that the attribute *change_type* must have a value such that its *defining_code* (its CODE_PHRASE) is in the set of CODE_PHRASEs in the *openEHR* Terminology which are in the group called (in english) “audit change type”.

A similar invariant is used for attributes of type CODE_PHRASE, which come from a code_set:

Media_type_terminology: media_type /= Void **and then**
code_set(“media types”).all_codes.has(media_type)

4.2.1 Class Definitions

4.2.1.1 TERMINOLOGY_SERVICE_ACCESS Class

CLASS	TERMINOLOGY_SERVICE_ACCESS	
Purpose	Defines an object providing proxy access to a terminology service.	
Functions	Signature	Meaning
	terminology (name: String): TERMINOLOGY_INTERFACE <i>require</i> name /= Void and then has_terminology (name: String) <i>ensure</i> Result /= Void	Return an interface to the terminology named ‘name’
	code_set (name: String): CODE_SET_INTERFACE <i>require</i> name /= Void and then has_code_set (name: String) <i>ensure</i> Result /= Void	Return an interface to the code_set named ‘name’
	has_terminology (name: String): Boolean <i>require</i> name /= Void and then not name.is_empty	True if terminology named ‘name’ known by this service.
	has_code_set (name: String): Boolean <i>require</i> name /= Void and then not name.is_empty	True if code_set named ‘name’ known by this service.
Invariants		

4.2.1.2 TERMINOLOGY_INTERFACE Class

CLASS	TERMINOLOGY_INTERFACE	
Purpose	Defines an object providing proxy access to a terminology.	
Functions	Signature	Meaning
	id: String	Identification of this Terminology
	all_codes: Set<CODE_PHRASE>	Return all codes known in this terminology
	codes_for_group_id (group_id: String): Set<CODE_PHRASE>	Return all codes under grouper 'group_id' from this terminology
	codes_for_group_name (name, lang: String): Set<CODE_PHRASE>	Return all codes under grouper whose name in 'lang' is 'name' from this terminology
	rubric_for_code (code, lang: String): String	Return all rubric of code 'code' in language 'lang'.
Invariants	<i>id_exists</i> : id != Void <i>and then not</i> id.is_empty	

4.2.1.3 CODE_SET_INTERFACE Class

CLASS	CODE_SET_INTERFACE	
Purpose	Defines an object providing proxy access to a code_set.	
Functions	Signature	Meaning
	id: String	Identification of this Terminology
	all_codes: Set<CODE_PHRASE>	Return all codes known in this terminology
Invariants	<i>id_exists</i> : id != Void <i>and then not</i> id.is_empty	

4.3 Code Sets

Code sets are not shown in full here, since their codes are derived from resources published by outside authorities; however, the *openEHR* code-set databases contain the full set of codes in each case.

4.3.1 Languages

This ISO code set defined by the ISO 639 standard consists of the “alpha-2” form of names of languages. This does not cover all languages, whereas ISO 639 “alpha-3” covers many more languages

of cultural or indigenous interest, but which nevertheless are unlikely to be supported by current software or operating systems. See <http://www.loc.gov/standards/iso639-2/langhome.html>.

Issuer: ISO Code set name: “languages”		
Code	Description	Mappings
“ab”	“Abkhazian”	
...	...	
“bg”	“Bulgarian”	
...	...	
“zh”	“Chinese”	
...	...	

4.3.2 Countries

This ISO code set defined by the ISO 3166 standard consists of 2-character names of countries and country subdivisions. For a definitive online rendition see <http://www.unicode.org/unicode/online-dat/countries.html>.

Issuer: ISO Code set name: “countries”		
Code	Description	Mappings
“af”	“Afghanistan”	
“al”	“Albania”	
...	...	

4.3.3 Character Sets

This IANA (Internet Naming Authority) code set consists of the names of recognised character sets. See <http://www.iana.org/assignments/character-sets> for authoritative source.

Issuer: IANA Code set name: “character sets”		
Code	Description	Mappings
ISO-10646-UTF-1		
...		
ISO_8859-3:1988		
...		

4.3.4 Media Types

This IANA (Internet Naming Authority) code set consists of the names of MIME media types. See <http://www.iana.org/assignments/media-types/text/> for authoritative source.

Issuer: IANA Code set name: “media types”		
Code	Description	Mappings
“text/plain”	Plain text encoded according to RFC3676	HL7_MediaType::14826
“text/html”	HTML text encoded according to RFC2854	HL7_MediaType::14828
“text/richtext”	Rich text encoded according to RFC2046	
“text/rtf”	Rich text encoded according to ftp://indri.pri-mate.wisc.edu/pub/RTF/RTF-Spec.rtf .	HL7_MediaType::14831
“text/sgml”		HL7_MediaType::14829

Issuer: IANA Code set name: "media types"		
Code	Description	Mappings
"text/rfc822-headers"		
"text/xml"		HL7_MediaType::14830
"audio/basic"		HL7_MediaType::14836
"audio/mpeg"		HL7_MediaType::14837
"application/pdf"		HL7_MediaType::14833
"application/msword"		HL7_MediaType::14834
...

4.3.5 Compression algorithms

This code set consists of the names of algorithms used to compress data, and is drawn from HL7's CompressionAlgorithms domain.

Issuer: openehr Code set name: "compression algorithms"		
Code	Description	Mappings
"compress"	Original UNIX <i>compress</i> algorithm and file format using the LZC algorithm (a variant of LZW).	HL7_CompressionAlgorithm::10624
"deflate"	The <i>deflate</i> compressed data format as specified in RFC 1951. See ftp://ftp.isi.edu/in-notes/rfc1951.txt .	HL7_CompressionAlgorithm::10621
"gzip"	A compressed data format that is compatible with the widely used GZIP utility as specified in RFC 1952. See ftp://ftp.isi.edu/in-notes/rfc1952.txt .	HL7_CompressionAlgorithm::10622
"zlib"	A compressed data format that also uses the deflate algorithm. Specified as RFC 1950 See ftp://ftp.isi.edu/in-notes/rfc1950.txt	HL7_CompressionAlgorithm::10623
"other"	Some other type of compression; might be retrievable upon direct inspection of data.	

4.3.6 Integrity check algorithms

This code set consists of the names of algorithms used to generate hashes for the purpose of integrity checks on data; its initial values are drawn from the HL7 IntegrityCheckAlgorithm domain.

Issuer: openehr Code set name: "integrity check algorithms"		
Code	Description (en)	Mappings
"SHA-1"	Secure hash algorithm - 1. Defined in FIPS PUB 180-1: Secure Hash Standard. As of April 17, 1995.	HL7_IntegrityCheckAlgorithm::17386
"SHA-256"	secure hash algorithm - 256. Defined in FIPS PUB 180-2: Secure Hash Standard	HL7_IntegrityCheckAlgorithm::17387
...	...	

4.4 Vocabularies and Terminologies

4.4.1 Measurable Properties

This vocabulary codifies purposes for physical properties corresponding to formal unit specifications, and allows comparison of Quantities with different units but which measure the same property. The vocabulary values are taken from:

- CEN ENV 12435 - “Medical Informatics - Expression of results of measurements in health sciences”
- HL7 “Unified Codes for Units of Measure”

Terminology: <i>openehr</i> Group_name(“en”): “ <i>measurable properties</i> ”			
Concept id	Rubric (en)	Description (en)	Mappings
...

4.4.2 Term Mapping Purpose

This vocabulary codifies purposes for term mappings as used in the class `TERM_MAPPING`. The use-case for this vocabulary is yet to be determined.

Terminology: <i>openehr</i> Group_name(“en”): “ <i>term mapping purpose</i> ”			
Concept id	Rubric (en)	Description (en)	Mappings
...	to be determined	...	

4.4.3 Participation Mode

This vocabulary codifies modes of participation of parties in an interaction (used in `PARTICIPATION` class). The initial set has been defined to be the same as HL7’s ParticipationMode vocabulary domain.

Terminology: <i>openehr</i> Group_name(“en”): “ <i>participation mode</i> ”			
Concept id	Rubric (en)	Description (en)	Mappings
193	“not specified”	Mode of participation is not specified; use only for legacy data.	
216	“face-to-face communication”	Face to face communications between parties in the same room.	HL7_ParticipationMode::16545
223	“interpreted face-to-face communication”	Face to face communications between parties in the same room with an interpreter	HL7_ParticipationMode::16545
217	“signing (face-to-face)”	Live face-to-face communication using a recognised sign language.	
195	“live audiovisual; videoconference; videophone”	Any audio-visual communication in real time	

Terminology: <i>openehr</i> Group_name("en"): " <i>participation mode</i> "			
Concept id	Rubric (en)	Description (en)	Mappings
198	"videoconferencing"	Live audio-visual communication over videoconferencing or other similar equipment.	HL7_ParticipationMode::16548
197	"videophone"	Live audio-visual communication	
218	"signing over video"	Live video communication using sign language.	
224	"interpreted video communication"	Live audio-visual communication involving an interpreter	
194	"asynchronous audio-visual; recorded video"	Audio-visual communication that is not live	
196	"recorded video"	Recorded video or video mail	
202	"live audio-only; telephone; internet phone; teleconference"	Any live audio-only communication.	HL7_ParticipationMode::V16544 (includes live)
204	"telephone"	Live verbal communication over a telephone.	HL7_ParticipationMode::16546
203	"teleconference"	Live verbal communication over teleconference	HL7_ParticipationMode::16546
204	"internet telephone"	Live verbal communication over a the internet.	HL7_ParticipationMode::16546
222	"interpreted audio-only"	Any live audio-only communication using an interpreter.	HL7_ParticipationMode::V16544 (includes live)
199	"asynchronous audio-only; dictated; voice mail"	Audio-only communication that is not live.	
200	"dictated"	Non-interactive audio-only information recorded on some medium, such as cassette tape.	HL7_ParticipationMode::16547
201	"voice-mail"	Audio messaging system	
212	"live text-only; internet chat; SMS chat; interactive written note"	Any live text-only communication	
213	"internet chat"	Live text-only communication over the internet	
214	"SMS chat"	Live text-only chat over mobile/cell phone	
215	"interactive written note"	Live text-only communication using written notes	HL7_ParticipationMode::16550
206	"asynchronous text; email; fax; letter; handwritten note; SMS message"	Any text-only communication including email, written text, SMS message etc.	HL7_ParticipationMode::V16549
211	"handwritten note"	Written communication by handwritten document.	HL7_ParticipationMode::16550
210	"printed/typed letter"	Written communication by typewritten document.	HL7_ParticipationMode::16551

Terminology: <i>openehr</i> Group_name("en"): " <i>participation mode</i> "			
Concept id	Rubric (en)	Description (en)	Mappings
207	"email"	Written communication by email.	HL7_ParticipationMode::16553 [include HL7_ParticipationMode::16554 (electronic data)]
208	"facsimile/telefax"	Non-interactive written communication using a fax machine.	HL7_ParticipationMode::16552
221	"translated text"	Non-interactive written communication requiring translation	HL7_ParticipationMode::V16549
209	"SMS message"	Messages sent via mobile/cell phone	
219	"physically present"	Participation by actions, where the participant is physically present.	HL7_ParticipationMode::16556
220	"physically remote"	Participation by actions, where the participant is not physically present, and the actions are transmitted by electronic means.	HL7_ParticipationMode::16557

4.4.4 Setting

This vocabulary codifies broad types of settings in which clinical care is delivered. It is not intended to be a perfect classification of the real world, but instead a practical coarse-grained categorisation to aid querying.

Terminology: <i>openehr</i> Group_name("en"): " <i>setting</i> "			
Concept id	Rubric (en)	Description (en)	Mappings
225	"home"	Care delivered in the patient's home by patient or health professional.	
227	"emergency care"	Care delivered in emergency situation, e.g. by ambulance workers.	
228	"primary medical care"	Care delivered by a doctor within a primary care framework (generalist, non-referred).	
229	"primary nursing care"	Care delivered by nurses within a primary care framework (community based, generalist clinic).	
230	"primary allied health care"	Care delivered by allied health practitioners such as physiotherapists, osteopaths, chiropractors, optometrists, chiropodist/pediatrist etc. within a primary care framework (community based, generalist clinic)	
231	"midwifery care"	Midwifery care in any framework	
232	"secondary medical care"	Care delivered in an institutional or specialist setting - usually as a result of a referral.	
233	"secondary nursing care"	Care delivered by nurses within a secondary care framework (inpatient, specialist clinic).	
234	"secondary allied health care"	Care delivered by allied health care professionals within a secondary care framework (inpatient, specialist clinic).	

Terminology: <i>openehr</i> Group_name("en"): "setting"			
Concept id	Rubric (en)	Description (en)	Mappings
235	"complementary health care"	Care delivered by chinese, ayurvedic, naturopath, homeopath etc practitioner.	
236	"dental care"	Care delivered in a dental practitioner setting.	
237	"nursing home care"	Care to the needs of patients in nursing homes, delivered in an institutional setting.	
238	"other care"	Care delivered in setting not described by other terms in this vocabulary.	

4.4.5 Attestation Status

This vocabulary codifies attestation statuses of Compositions or other elements of the health record, and is drawn from the HL7 ParticipationSignature domain, as used in CDA.

Terminology: <i>openehr</i> Group_name("en"): "attestation status"			
Concept id	Rubric (en)	Description (en)	Mappings
240	"signed"	This attestation has been signed by its required signatory/ies.	HL7_ParticipationSignature::10284
241	"intended"	This attestation is awaiting the signature of its signatory/ies.	HL7_ParticipationSignature::13977
242	"required"	This attestation requires the signature of its signatory/ies.	HL7_ParticipationSignature::10283

4.4.6 Version Lifecycle State

This vocabulary codifies lifecycle states of Compositions or other elements of the health record.

Terminology: <i>openehr</i> Group_name("en"): "version lifecycle state"			
Concept id	Rubric (en)	Description (en)	Mappings
244	"draft"	Item is in draft state: not ready for viewing by other users.	
245	"active"	Item is active and available for shared use.	
246	"inactive"	Item is marked inactive due to logical deletion or other similar operation.	
247	"awaiting approval"	Item is awaiting to approval to go into active state.	

4.4.7 Audit Change Type

This vocabulary codifies the kinds of changes to data which are recorded in audit trails.

Terminology: <i>openehr</i> Group_name("en"): "audit change type"			
Concept id	Rubric (en)	Description (en)	Mappings
249	"creation"	Change type was creation.	HL7_CDA: CEN:

Terminology: <i>openehr</i> Group_name("en"): "<i>audit change type</i>"			
Concept id	Rubric (en)	Description (en)	Mappings
250	"amendment"	Change type was amendment.	HL7_CDA: CEN:
251	"modification"	Change type was modification.	HL7_CDA: CEN:
252	"synthesis"	Change type was synthesis - creation by a conversion gateway.	HL7_CDA: CEN:
253	"unknown"	Type of change unknown.	HL7_CDA: CEN:

4.4.8 Related Party relationship

This vocabulary codifies the relationship between the subject of care and some other party mentioned in the health record.

Terminology: <i>openehr</i> Group_name("en"): <i>"related party relationship"</i>			
Concept id	Rubric (en-uk)	Description (en)	Mappings
0	"self"	The party is the subject of EHR	HL7_RoleCode:: CEN:
3	"foetus"	The party is a foetus	HL7: CEN:
10	"mother"	The party is the mother of the subject of EHR	HL7: CEN:
9	"father"	The party is the father of the subject of the EHR	HL7: CEN:
6	"donor"	The party is a donor of organs or other body products to the EHR subject.	HL7: CEN:
253	"unknown"	Relationship to party is unknown.	HL7: CEN:
261	"adopted daughter"	Relationship of adopted daughter to subject of EHR	HL7: CEN:
260	"adopted son"	Relationship of adopted son to subject of EHR	HL7: CEN:
259	"adoptive father"	Relationship of adoptive father to subject of EHR	HL7: CEN:
258	"adoptive mother"	Relationship of adoptive mother to subject of EHR	HL7: CEN:
256	"biological father"	Relationship of biological father to subject of EHR	HL7: CEN:
255	"biological mother"	Relationship of biological mother to subject of EHR	HL7: CEN:
23	"brother"	Relationship of brother to subject of EHR	HL7: CEN:
28	"child"	Relationship of child to subject of EHR	HL7: CEN:
265	"cohabitee"	Lives with the subject of EHR	HL7: CEN:
257	"cousin"	Relationship of cousin to subject of EHR	HL7: CEN:
29	"daughter"	Relationship of daughter to subject of EHR	HL7: CEN:
264	"guardian"	Relationship of guardian to subject of EHR	HL7: CEN:
39	"maternal aunt"	Relationship of maternal aunt to subject of EHR	HL7: CEN:
8	"maternal grandfather"	Relationship of maternal grandfather to subject of EHR	HL7: CEN:
7	"maternal grandmother"	Relationship of maternal grandmother to subject of EHR	HL7: CEN:
38	"maternal uncle"	Relationship of maternal uncle to subject of EHR	HL7: CEN:

Terminology: <i>openehr</i> Group_name("en"): "related party relationship"			
Concept id	Rubric (en-uk)	Description (en)	Mappings
189	"neonate"	Relationship of neonate to subject of EHR	HL7: CEN:
254	"parent"	Relationship of parent to subject of EHR	HL7: CEN:
22	"partner/spouse"	The husband or wife or life partner of the subject of EHR	HL7: CEN:
41	"paternal aunt"	Relationship of paternal aunt to subject of EHR	HL7: CEN:
36	"paternal grandfather"	Relationship of paternal grandfather to subject of EHR	HL7: CEN:
37	"paternal grandmother"	Relationship of paternal grandmother to subject of EHR	HL7: CEN:
40	"paternal uncle"	Relationship of paternal uncle to subject of EHR	HL7: CEN:
27	"sibling"	Relationship of sibling to subject of EHR	HL7: CEN:
24	"sister"	Relationship of sister to subject of EHR	HL7: CEN:
31	"son"	Relationship of son to subject of EHR	HL7: CEN:
263	"step father"	Relationship of step father to subject of EHR	HL7: CEN:
262	"step mother"	Relationship of step mother to subject of EHR	HL7: CEN:
25	"step or half brother"	Relationship of step or half brother to subject of EHR	HL7: CEN:
26	"step or half sister"	Relationship of step or half sister to subject of EHR	HL7: CEN:

4.4.9 Event Math Function

This vocabulary codifies mathematical functions of non-instantaneous events.

Terminology: <i>openehr</i> Group_name("en"): "event math function"			
Concept id	Rubric (en)	Description (en)	Mappings
145	"minimum"	Value of the interval-event is the minimum value of the discrete events which the interval-event summarises.	
144	"maximum"	Value of the interval-event is the maximum value of the discrete events which the interval-event summarises.	
267	"mode"	Value of the interval-event is the modal (most common) value of the discrete events which the interval-event summarises.	

Terminology: <i>openehr</i> Group_name("en"): "event math function"			
Concept id	Rubric (en)	Description (en)	Mappings
268	"median"	Value of the interval-event is the median (centre value in sorted series) value of the discrete events which the interval-event summarises.	
146	"mean"	Value of the interval-event is the average value of the discrete events which the interval-event summarises.	
147	"delta"	Value of the interval-event is the net change over the period which the interval-event summarises.	
148	"total"	Value of the interval-event is the sum of the values of the discrete events which the interval-event summarises (typically differential flow measurements, e.g. blood loss).	

4.4.10 Null Flavours

This vocabulary codifies "flavours of null" for missing data items.

Terminology: <i>openehr</i> Group_name("en"): "null flavours"			
Concept id	Rubric (en)	Description (en)	Mappings
271	"no information"	No information provided; nothing can be inferred as to the reason why, including whether there might be a possible applicable value or not.	HL7_NullFlavor::V10610
253	"unknown"	A possible value exists but is not provided.	HL7_NullFlavor::V10612
272	"masked"	The value has not been provided due to privacy settings.	HL7_NullFlavor::17932
273	"not applicable"	No valid value exists for this data item.	HL7_NullFlavor::10611

5 RM.SUPPORT.MEASUREMENT Package

5.1 Overview

The Measurement package defines a minimum of semantics relating to quantitative measurement, units, and conversion, enabling the Quantity package of the *openEHR* Data Types Information Model to be correctly expressed. As for the Terminology package, a simple service interface is assumed, which provides useful functions to other parts of the reference model. The definitions underlying measurement and units come from a variety of sources, including:

- CEN ENV 12435, Medical Informatics - Expression of results of measurements in health sciences (see <http://www.centc251.org>);
- the Unified Code for Units of Measure (UCUM), developed by Gunther Schadow and Clement J. McDonald of The Regenstrief Institute (available in HL7v3 ballot materials; <http://www.hl7.org>).

These of course rest in turn upon a vast amount of literature and standards, mainly from ISO on the subject of scientific measurement.

5.2 Service Interface

A simple measurement data service interface is defined according to FIGURE 4, enabling quantitative semantics to be used formally from within the Reference Model. Note that this service as currently defined in no way seeks to properly model the semantics of units, conversions etc - it provides only the minimum functions required by the *openEHR* Reference Model.

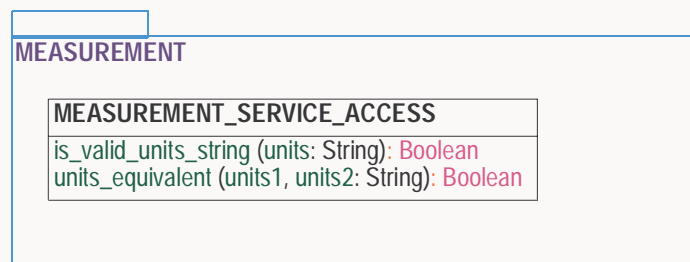


FIGURE 4 Simple Measurement Interface

5.2.1 Class Definitions

5.2.1.1 MEASUREMENT_SERVICE_ACCESS Class

CLASS	MEASUREMENT_SERVICE_ACCESS	
Purpose	Defines an object providing proxy access to a measurement information service.	
Functions	Signature	Meaning

CLASS	MEASUREMENT_SERVICE_ACCESS	
	<p>is_valid_units_string (units: String): Boolean <i>require</i> units /= Void</p>	<p>True if the units string 'units' is a valid string according to the HL7 UCUM specification.</p>
	<p>units_equivalent (units1, units2: String): Boolean <i>require</i> units1 /= Void <i>and then</i> is_valid_units_string(units1) units2 /= Void <i>and then</i> is_valid_units_string(units2)</p>	<p>True if two units strings correspond to the same measured property.</p>
Invariants		

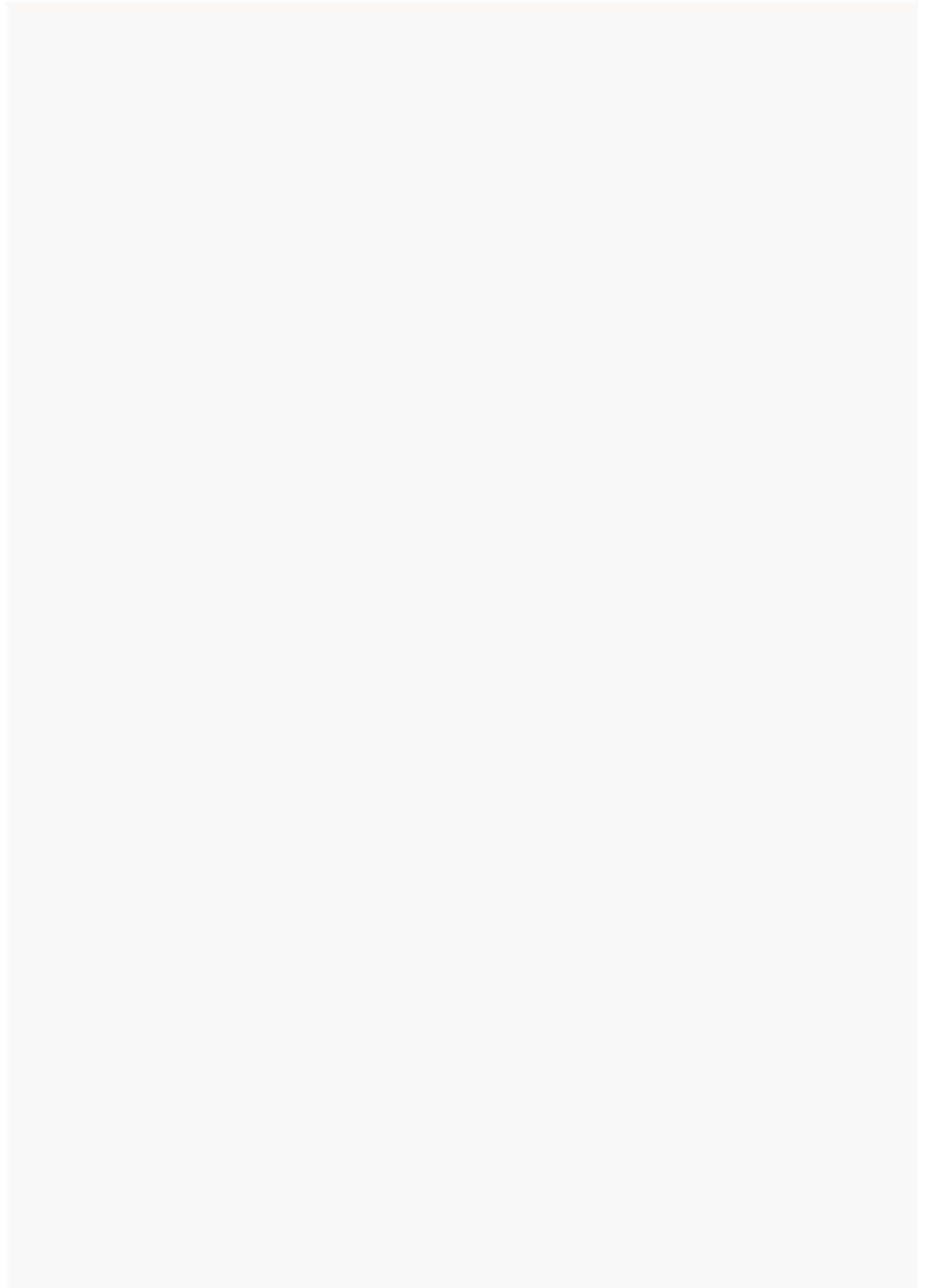
6 RM.SUPPORT.DEFINITION Package

This section describes symbolic definitions used by the *openEHR* models.

6.1 Constants

Constants used in the *openEHR* Reference Model specifications:

- **CR:** Character is '\015'
- **LF:** Character is '\012'



END OF DOCUMENT