Multi-Level Health Information Modelling

Reference Manual

Release 2.2.1

Timothy W. Cook & Contributors

Table of Contents

Front Matter

Acknowledgements

This work has received financial and in-kind support from the following persons and organizations:

- Labatorio Nacional de Computacão Cientifica, Instituto Nacional Cientifica e Tecnologia - Medicine Assisted by Cientifica Computing (INCT-MACC)
- Instituto de Saúde da Comunidade da Universidade Federal Fluminense
- Universidade do Estado do Rio de Janeiro
- Timothy W. Cook, Independent Consultant
- Roger Erens, Independent Consultant

Using the MLHIM Reference Manual

This section describes typographical conventions and other information to help you get the most from this document.

The intended audience for this manual includes; software developers, systems analysts and knowledge workers in the healthcare domain. It is assumed that the reader has knowledge of object-oriented notation, concepts and software construction practices.

Release Information

The official published version is in PDF format in the English language. The ODT version is always considered a work in progress. Each version of the PDF release will carry a version number that is the date of release followed by the a language code and locality code. As an example; a release in English on January 1, 2011 will have the filename:

mlhim-ref-man-2011-01-01-en-US.pdf

Pronunciation

MLHIM is pronounced *muh-leem*. Click *Hear How It Sounds* for when it is used in spoken English language such as presentations or general discussions.

Conformance

Conformance to these specification are represented in a Language Implementation Specification (LIS). A LIS is formal document detailing the mappings and conventions used in relation to these specifications.

A LIS is in direct conformance to these specifications when:

1. All datatypes are defined and mapped.
2. the value spaces of the healthcare datatypes used by the entity to be identical to the value spaces specified herein
3. to the extent that the entity provides operations other than movement or translation of values, define operations on the healthcare datatypes which can be derived from, or are otherwise consistent with the characterizing operations specified herein

Compliance

These specifications:

- are in indirect conformance with ISO/DIS 21090/2008

- are in compliance with applicable sections of ISO 18308/2008
- are in compliance with applicable sections of ISO/TR 20514:2005
- are in compliance with applicable sections of ISO 13606-1:2007

Introduction

The Multi-Level Health Information Modeling (MLHIM) specifications are partially derived from ISO Healthcare Information Standards and the openEHR 1.0.2 specifications and the intent is that MLHIM 1.x be technologically inter-operable.

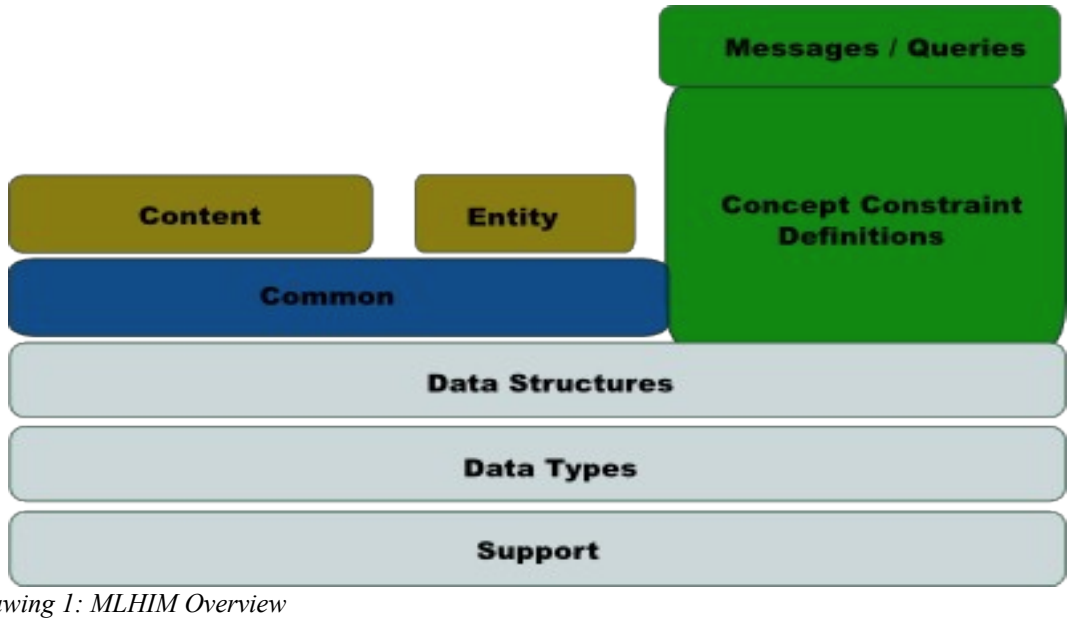
MLHIM 2.x (this document) introduces modernization through the use of XML technologies and improved modeling tools as well as application development platforms. These specifications can be implemented in any object oriented language. While a certain level of knowledge is assumed; the primary goal of these specifications is to make them 'readable' by the widest possible number of people. The primary motivator for these specifications is the complexity involved in the recording of the temporal-spatial relationships in healthcare information while maintaining the original semantics across all applications; for all time.

We invite you to join us in this effort to maintain the specifications and build great, translatable healthcare tools and applications for global use.

International input is encouraged in order for the MLHIM specifications to be truly inter-operable, available to everyone in all languages and most of all; implementable by mere mortals.

In actual implementation, the packages/classes should be implemented per the standard language naming format. A Language Implementation Specification (LIS) should be created for each language. For example; MLHIM-Python-LIS.odt or MLHIM-Java-LIS.odt.

Drawing 1 depicts the overall layered relationship of the packages.



Drawing 1: MLHIM Overview

Each of the package chapters lists the classes to be implemented. The format is:

- The generic class name
- The purpose
- Use
- Potential misuses
- Class or classes it inherits from
- Class type: Abstract: True/False
- Elements

MLHIM intentionally does not specify behavior within a class. Only the data and constraints are specified. Behavior may differ between various applications and should not be specified at the information model level.

The generic class names in the specification documents are in CamelCase type. Since this is most typical of implementation usage. This convention avoids confusion with major programming languages. There are bindings for the XML Schemas that have been generated for popular programming languages under the banner of the Open Source Health Information Platform (OSHIP). See the chapter on OSHIP for more details.

Only the reference model is implemented in software. The domain knowledge models are implemented in the XML Schema language and they represent constraints on the reference model. These knowledge models are called Concept Constraint Definitions and the acronyms CCD and CCDs are used throughout MLHIM documents to mean these XML Schema files. This means that CCDs form a model that data instances can be created from and according to a specific CCD, the data will be valid. However, any data instance should be able to be imported into any MLHIM based application since the root data model is the reference model. But, the full semantics of that data will not be known unless and until the defining CCD is available to that application.

The above highlighted paragraph describes the foundation of semantic interoperability in MLHIM implementations. You must understand this and the implications it carries to be successful with implementing MLHIM based applications. See the Constraint Definition section for an in-depth discussion of Concept Constraint Definitions (CCDs).

MLHIM Modelling

The MLHIM specifications are arranged into packages. These packages represent logical groupings of classes; providing ease of consistent implementation. The fundamental concepts, expressed in the reference model classes, are based on basic philosophical concepts of real world entities. These broad concepts can then be constrained to more specific concepts; using models created by domain experts, in this case healthcare experts.

In MLHIM 1.x.x these constraints were known as archetypes, expressed in what is called the archetype definition language (ADL). In MLHIM 2.x and later, we use an XML Schema (XSD) representation called a Concept Constraint Definition (CCD).

CCDs may contain other CCDs in a structure called a Slot. This provides a basis for selection and reuse, at runtime, of commonly occurring concepts within a larger concept. A CCD is a maximal data model for a concept. Therefore it may be further constrained at the

implementation level through the use of implementation templates in the given framework. These templates shall be constructed in the implementation and may or may not be sharable across applications.

The real advantage to using the MLHIM approach to health care information modelling is that it provides for a wide variety of healthcare applications to be developed based on the broad concepts defined in the reference model. Then by having domain experts within the healthcare field define and create the CCDs; they can be shared across multiple applications so that the structure of the data is not locked into one specific application. But can be exchanged among many different applications. This properly implements the separation of roles between IT people and domain experts.

To demonstrate the differences between the MLHIM approach and the typical data model design approach; We will use two common metaphors.

1. The first is for the data model approach to developing software. This is where a set of database definitions are created based on a requirements statement representing an information model. An application is then developed to support all of the functionality required to input, manipulate and output this data as information; all built around the data model. This approach is akin to a jigsaw puzzle (the software application) where the shape of the pieces are the syntax and the design and colors are the semantics, of the information represented in an aggregation of data components described by the model. This produces an application that, like the jigsaw puzzle, can provide for pieces (information) to be exchanged only between exact copies of the same puzzle. If you were to try to put pieces from one puzzle, into a different puzzle you might find that a piece has the same shape (syntax) but the picture on the piece (semantics) will not be the same. Even though they both belong to the same domain; jigsaw puzzles. You can see that getting a piece from one puzzle to correctly fit into another is going to require manipulation of the basic syntax (shape) and /or semantics (picture) of the piece. This can also be extended to the relationship that the puzzle has a defined limit of its six sides. It cannot, reasonably, be extended to incorporate new pieces (concepts) discovered after the initial design.
2. The multi-level approach used in MLHIM is akin to creating models (applications) using the popular toy blocks made by Lego and other companies. If you compare a box of these interlocking blocks to the reference model and the instructions to creating a specific toy model (software application), where these instructions present a concept constraint. You can see that the same blocks can be used to represent multiple toy models without any change to the physical shape, size or color of each block. Now we can see that when new concepts are created within healthcare, they can be represented as instructions for building a new toy model; using the same fundamental building blocks that the original software applications were created upon.

The Reference Model

Support

ExceptionalValue

Subclasses are used to indicate why a value is missing (Null) or is outside a measurable range.

Inherits From: xs:anyType
Abstract: True

Element Name	ev_name
Datatype	xs:string
minOccurs	1
maxOccurs	1
Note	The class names (complexType names) are abbreviations according to the ISO 21090 Null Flavours. This attribute is the full name.

Element Name	ev_meaning
Datatype	xs:string
minOccurs	1
maxOccurs	1
Note	The descriptive meaning/usage for the class/complexType.

NI

The value is exceptional (missing, incomplete, improper). No information is available as to the reason for being an exceptional value is provided. This is the most general and default value.

Inherits From: mlhim2:ExceptionalValue
Abstract: False

MSK

There is information on this item available but it has not been provided by the sender due to security, privacy or other reasons. There may be an alternate method of obtaining the information.

Inherits From: mlhim2:NI

Abstract: False

NA

No proper value is applicable in this context e.g.,the number of cigarettes smoked per day by a non-smoker subject.

Inherits From: mlhim2:NI
Abstract: False

UNK

A proper value is applicable, but not known.

Inherits From: mlhim2:NI
Abstract: False

INV

The value as represented in the instance is not a member of the set of permitted data values in the constrained value domain of a variable.

Inherits From: mlhim2:NI
Abstract: False

DER

An actual value may exist, but it must be derived from the provided information; usually an expression is provided directly.

Inherits From: mlhim2:INV
Abstract: False

UNC

No attempt has been made to encode the information correctly but the raw source information is represented, usually in free text.

Inherits From: mlhim2:INV
Abstract: False

OTH

The actual value is not a member of the permitted data values in the variable. (e.g., when the value of the variable is not by the coding system)

Inherits From: mlhim2:INV
Abstract: False

ASKU

Information was sought but not found (e.g., patient was asked but did not know).

Inherits From: mlhim2:UNK
Abstract: False

ASKR

Information was sought but refused to be provided (e.g., patient was asked but refused to answer). This element is not part of ISO 21090 but was added in MLHIM2 to provide better coverage of missing values.

Inherits From: mlhim2:UNK
Abstract: False

NASK

This information has not been sought (e.g., patient was not asked).

Inherits From: mlhim2:UNK
Abstract: False

QS

The specific quantity is not known, but is known to non-zero and it is not specified because it makes up the bulk of the material; "Add 10mg of ingredient X, 50mg of ingredient Y and sufficient quantity of water to 100mL."

Inherits From: mlhim2:UNK
Abstract: False

TRC

The content is greater or less than zero but too small to be quantified.

Inherits From: mlhim2:UNK
Abstract: False

NINF

Negative infinity of numbers

Inherits From: mlhim2:OTH
Abstract: False

PINF

Positive infinity of numbers

Inherits From: mlhim2:OTH
Abstract: False

NAV

Information is unavailable at this time but is expected that it will be available later.
Abstract: False
Inherits From: mlhim2:ASKU

ObjectRef

Class describing a reference to another object, which may exist locally or be maintained outside the current namespace, e.g. in another service. Services are usually external, e.g.

available in a LAN (including on the same host) or the internet via Corba, SOAP, or some other distributed protocol. However, in small systems they may be part of the same executable as the data containing the Id.

Inherits From: xs:anyType
Abstract: False

Element Name	uuid
Datatype	xs:token
minOccurs	1
maxOccurs	1
Note	Universally unique id of an object, regardless of where it is stored.

Element Name	namespace
Datatype	xs:normalizedString
minOccurs	1
maxOccurs	1
Note	Namespace to which this identifier belongs in the local system context.

Element Name	obj_type
Datatype	xs:normalizedString
minOccurs	1
maxOccurs	1
Note	Name of the class (concrete or abstract) of object to which this identifier type refers, e.g. “Party”, “Person”, “Guideline” etc. These class names are from the relevant reference model. The type name “anyType” can be used to indicate that any type is accepted (e.g. if the type is unknown).

EntityRef

Identifier for entities in an identity service. There are typically a number of subtypes of the Entity class, including Person, Organization, Device, etc. Abstract supertypes are allowed if the referenced object is of a type not known by the current implementation of this class. For example, if the entity model is changed by the addition of a new subtype, valid EntityRefs can still be constructed to them.

Inherits From: mlhim2:ObjectRef
Abstract: False

Datatypes

DvAny

Serves as a common ancestor of all datatypes in MLHIM models.
Inherits From: xs:anyType
Abstract: True

Element Name	valid_time_begin
Datatype	xs:dateTime
minOccurs	0
maxOccurs	1
Note	Used to tag the earliest datetime that this data element is valid. If present this must be a valid datetime string including timezone.

Element Name	valid_time_end
Datatype	xs:dateTime
minOccurs	0
maxOccurs	1
Note	Used to tag the expiration of the validity of this data item. If present this must be a valid datetime string including timezone

Element Name	ev
Datatype	mlhim2:ExceptionalValue
minOccurs	0
maxOccurs	1
Note	The exceptional value. Often referred to as Null Flavour.

DvChar

Abstract parent of all string datatypes.

Inherits From:mlhim2:DvAny

Abstract: True

DvString

A text item, which may contain any amount of legal characters arranged as e.g. words, sentences etc. as its data value (dv)
Inherits From: mlhim2:DvChar
Abstract: False

Element Name	uuid
Datatype	xs:token
minOccurs	1
maxOccurs	1
Note	UUID in order to track translations.

Element Name	language
Datatype	xs:language
minOccurs	0
maxOccurs	1
Note	Optional indicator of the localised language in which the value is written. Coded IAW IETF RFC 5646. http://tools.ietf.org/html/rfc5646 Only used when the text object is in a different language from the enclosing CCD.

Element Name	dv
Datatype	xs:string
minOccurs	0
maxOccurs	1
Note	Displayable rendition of the item.

DvNormalizedString

Represents a character string that may contain any Unicode character allowed by XML. Certain characters, namely the "less than" symbol (<) and the ampersand (&), must be escaped (using the entities `<` and `&`, respectively) when used in strings in XML instances

Inherits From: mlhim2:DvChar
Abstract: False

Element Name	uuid
Datatype	xs:token
minOccurs	1
maxOccurs	1
Note	UUID in order to track translations.

Element Name	language
Datatype	xs:language
minOccurs	0
maxOccurs	1
Note	Optional indicator of the localised language in which the value is written. Coded IAW IETF RFC 5646. http://tools.ietf.org/html/rfc5646 Only used when the text object is in a different language from the enclosing CCD.

Element Name	dv
Datatype	xs:normalizedString
minOccurs	0
maxOccurs	1
Note	Displayable rendition of the item.

DvToken

Represents a character string that may contain any Unicode character allowed by XML. Certain characters, namely the "less than" symbol and the ampersand, must be escaped (using the entities < and &, respectively) when used in strings in XML instances.

The name token may be slightly confusing because it implies that there may be only one token with no whitespace. In fact, there can be whitespace in a token value. The xsd:token type has a whiteSpace facet of collapse, which means that the processor replaces each carriage return, line feed, and tab by a single space. After this replacement, each group of consecutive spaces is collapsed into one space character, and all leading and trailing spaces are removed. This processing is equivalent to the processing of non-CDATA attribute values in XML 1.0.

Inherits From: mlhim2:DvChar
Abstract: False

Element Name	dv
Datatype	xs:token
minOccurs	0
maxOccurs	1
Note	Displayable rendition of the item.

DvLanguage

A language code IAW RFC 3066, Tags for the Identification of Languages. The three most common formats are:

- For ISO-recognized languages, the format is a two- or three-letter, (usually lowercase) language code that conforms to ISO 639, optionally followed by a

hyphen and a two-letter, (usually uppercase) country code that conforms to ISO 3166. For example, en or en-US.

- For languages registered by the Internet Assigned Numbers Authority (IANA), the format is i-langname, where langname is the registered name. For example, i-navajo.
- For unofficial languages, the format is x-langname, where langname is a name of up to eight characters agreed upon by the two parties sharing the document. For example, x-Newspeak.

Inherits From:mlhim2:DvChar
Abstract: False

Element Name	dv
Datatype	xs:language
minOccurs	0
maxOccurs	1
Note	Displayable rendition of the language code.

DvCodedString

A text item whose dv attribute must be the long name or description from a controlled terminology, the key (i.e. the 'code') of which is the code_string attribute.

Inherits From:mlhim2:DvString
Abstract: False

Element Name	code_string
Datatype	xs:string
minOccurs	0
maxOccurs	1
Note	The key used by the terminology service to identify a concept or coordination of concepts. This string is most likely parsable inside the terminology service, but nothing can be assumed about its syntax outside that context. In the NLM Metathesarus this would be the Concept Unique Identifier (CUI).

Element Name	terminology_name
Datatype	xs:string
minOccurs	0
maxOccurs	1
Note	Full Source Name from NLM Metathesarus; or similar.

Element Name	terminology_abbrev
Datatype	xs:string
minOccurs	0
maxOccurs	1
Note	Version Source Abbreviation (VSAB) from NLM Metathesarus; or similar

Element Name	uuid
Datatype	xs:token
minOccurs	1
maxOccurs	1
Note	UUID in order to track translations.

Element Name	language
Datatype	xs:language
minOccurs	0
maxOccurs	1
Note	Optional indicator of the localised language in which the value is written. Coded IAW IETF RFC 5646. http://tools.ietf.org/html/rfc5646 Only used when the text object is in a different language from the enclosing CCD.

DvIdentifier

Type for representing identifiers of real-world entities. Typical identifiers include: drivers license number, social security number, veterans affairs number, prescription id, order id, and so on.

Inherits From: mlhim2:DvString

Abstract: False

Element Name	issuer
Datatype	xs:string
minOccurs	0
maxOccurs	1
Note	Authority which issues the kind of id used in the id field of this object.

Element Name	assigner
Datatype	xs:string
minOccurs	0
maxOccurs	1
Note	Organization that assigned the id to the item or person being identified.

Element Name	name
Datatype	xs:string
minOccurs	0
maxOccurs	1
Note	The identifier common name, such as “Driver's License” or “SSN”.

Element Name	dv
Datatype	xs:string
minOccurs	0
maxOccurs	1
Note	The data value for this identifier.

Element Name	uuid
Datatype	xs:token
minOccurs	1
maxOccurs	1
Note	UUID in order to track translations.

Element Name	language
Datatype	xs:language
minOccurs	0
maxOccurs	1
Note	Optional indicator of the localised language in which the value is written. Coded IAW IETF RFC 5646. http://tools.ietf.org/html/rfc5646 Only used when the text object is in a different language from the enclosing CCD.

DvBoolean

Items which are truly boolean data, such as true/false or yes/no answers. Use for such data, it is important to devise the meanings (usually questions in subjective data) carefully, so that the only allowed results are in fact true or false.

Potential MisUse: The DvBoolean class should not be used as a replacement for naively modelled enumerated types such as male/female etc. Such values should be coded, and in any case the enumeration often has more than two values.

Inherits From: mlhim2:DvAny
Abstract: False

Element Name	dv
Datatype	xs:boolean
minOccurs	0
maxOccurs	1
Note	A string indicating a boolean type result according to the rules described below. Boolean True values are represented by any one of these strings: ["True","1","T"]. Boolean False values are represented by any one of these strings: ["False","0","F"]. Note that the strings are not case-sensitive; i.e. "T" == "t" and "TRUE" == "t". The empty string; "" represents a Void or Null value, in which case the ev attribute cannot be empty.

DvURI

A reference to an object which conforms to the Universal Resource Identifier (URI) standard, as defined by W3C RFC 2936. See "Universal Resource Identifiers in WWW" by Tim Berners-Lee at http://www.ietf.org/rfc/rfc2396.txt. This is a World-Wide Web RFC for global identification of resources. See http://www.w3.org/Addressing for a starting point on URIs. See http://www.ietf.org/rfc/rfc2806.txt for new URI types like telephone, fax and modem numbers. Enables external resources to be referenced from within the content of the EHR. A number of functions return the logical subparts of the URI string.

Inherits From: mlhim2:DvAny
Abstract: False

Element Name	dv
Datatype	xs:anyURI
minOccurs	0
maxOccurs	1
Note	Value of URI as a String.

DvParagraph

A logical composite text value consisting of a series of DvStrings, i.e. plain or coded text to form a larger tract of prose, which may be interpreted for display purposes as a paragraph. DvParagraph is the standard way for constructing longer text items in summaries, reports and so on.

Element Name	items
Datatype	mlhim2:DvString
minOccurs	0
maxOccurs	1
Note	List of Items making up the paragraph, each of which is a text item. The items should be displayed in sequential order from smallest index number to highest index number.

DvEncapsulated

Abstract class defining the common meta-data of all types of encapsulated data.

Inherits From: mlhim2:DvAny
Abstract: True

Element Name	size
Datatype	xs:integer
minOccurs	0
maxOccurs	1
Note	Original size in bytes of unencoded encapsulated data. I.e. encodings such as base64, hexadecimal etc do not change the value of this attribute.

Element Name	charset
Datatype	xs:string
minOccurs	0
maxOccurs	1
Note	Name of character encoding scheme in which this value is encoded. Unicode is the default assumption in MLHIM, with UTF-8 being the assumed encoding. This attribute allows for variations from these assumptions.

Element Name	language
Datatype	xs:language
minOccurs	0
maxOccurs	1
Note	Optional indicator of the localised language in which the value is written. Coded IAW IETF RFC 5646. http://tools.ietf.org/html/rfc5646 Only used when the text object is in a different language from the enclosing CCD.

DvParsable

Encapsulated data expressed as a parsable String. The internal model of the data item is not described in the MLHIM model in common with other encapsulated types, but in this case, the form of the data is assumed to be plaintext, rather than compressed or other types of large binary data.

Inherits From: mlhim2:DvEncapsulated

Abstract: False

Element Name	dv
Datatype	xs:string
minOccurs	0
maxOccurs	1
Note	The string, which may validly be empty in some syntaxes

Element Name	formalism
Datatype	xs:string
minOccurs	0
maxOccurs	1
Note	Name of the formalism, e.g. "MAG 1.0", “GLIF 1.0”, “PROforma” etc.

DvMedia

A specialisation of DvEncapsulated for audiovisual and biosignal types. Includes further metadata relating to multimedia types which are not applicable to other subtypes of DvEncapsulated.

Inherits From: mlhim2:DvEncapsulated
Abstract: True

Element Name	mime_type
Datatype	xs:normalizedString
minOccurs	0
maxOccurs	1
Note	Content type as defined in RFC2045 and RFC 2046

Element Name	compression_type
Datatype	xs:normalizedString
minOccurs	0
maxOccurs	1
Note	Compression type, a coded value from the MLHIM "Integrity check" code set. Void means no compression.

Element Name	hash_result
Datatype	xs:normalizedString
minOccurs	0
maxOccurs	1
Note	integrity checksum

Element Name	hash_function
Datatype	xs:normalizedString
minOccurs	0
maxOccurs	1
Note	Type of integrity check.

Element Name	uri
Datatype	xs:anyURI
minOccurs	0
maxOccurs	1
Note	URI reference to electronic information stored outside the record as a file, database entry etc, if supplied as a reference.

Element Name	alt_text
Datatype	xs:normalizedString
minOccurs	0
maxOccurs	1
Note	Text to display in lieu of multimedia display/replay

DvMediaMultipart

Inherits From: mlhim2:DvMedia

Abstract: False

Element Name	dv
Datatype	xs:anyType
minOccurs	0
maxOccurs	1
Note	The contents of the item.

DvMediaApp

Inherits From: mlhim2:DvMedia

Abstract: False

Element Name	dv
Datatype	xs:anyType
minOccurs	0
maxOccurs	1
Note	The contents of the item.

DvMediaText

Inherits From: mlhim2:DvMedia

Abstract: False

Element Name	dv
Datatype	xs:string
minOccurs	0
maxOccurs	1
Note	The contents of the item.

DvMediaVideo

Inherits From: mlhim2:DvMedia

Abstract: False

Element Name	dv
Datatype	xs:base64Binary
minOccurs	0
maxOccurs	1
Note	The contents of the item.

DvMediaAudio

Inherits From: mlhim2:DvMedia

Abstract: False

Element Name	dv
Datatype	xs:base64Binary
minOccurs	0
maxOccurs	1
Note	The contents of the item.

DvMediaImage

Inherits From: mlhim2:DvMedia

Abstract: False

Element Name	dv
Datatype	xs:base64Binary
minOccurs	0
maxOccurs	1
Note	The contents of the item.

DvMediaMsg

Inherits From: mlhim2:DvMedia

Abstract: False

Element Name	dv
Datatype	xs:base64Binary
minOccurs	0
maxOccurs	1
Note	The contents of the item.

DvMediaModel

Inherits From: mlhim2:DvMedia

Abstract: False

Element Name	dv
Datatype	xs:base64Binary
minOccurs	0
maxOccurs	1
Note	The contents of the item.

DvInterval

Generic class defining an interval (i.e. range) of a comparable type. An interval is a contiguous subrange of a comparable base type. Used to define intervals of dates, times, quantities (whose units match) and so on.

Inherits From: mlhim2:DvAny

Abstract: False

Element Name	lower
Datatype	mlhim2:DvOrdered
minOccurs	0
maxOccurs	1
Note	Lower boundary.

Element Name	upper
Datatype	mlhim2:DvOrdered
minOccurs	0
maxOccurs	1
Note	Upper boundary.

Element Name	lower_included
Datatype	xs:boolean
minOccurs	1
maxOccurs	1
Note	Is the lower boundary included in the interval?

Element Name	upper_included
Datatype	xs:boolean
minOccurs	1
maxOccurs	1
Note	Is the upper boundary included in the interval?

Element Name	lower_unbounded
Datatype	xs:boolean
minOccurs	1
maxOccurs	1
Note	If True, there is no lower boundary

Element Name	upper_unbounded
Datatype	xs:boolean
minOccurs	1
maxOccurs	1
Note	If True, there is no upper boundary

ReferenceRange

Defines a named range to be associated with any Ordered datum. Each such range is particular to the patient and context, e.g. sex, age, and any other factor which affects ranges. May be used to represent normal, therapeutic, dangerous, critical etc ranges.

Inherits From: mlhim2:DvAny
Abstract: False

Element Name	definition
Datatype	mlhim2:DvString
minOccurs	1
maxOccurs	1
Note	Term whose value indicates the meaning of this range, e.g. “normal”, “critical”, “therapeutic”etc.

Element Name	data_range
Datatype	mlhim2:DvInterval
minOccurs	1
maxOccurs	1
Note	The data range for this meaning.

DvOrdered

Abstract class defining the concept of ordered values, which includes ordinals as well as true quantities. It defines the functions less than and is_strictly_comparable_to, the latter of which must evaluate to True for instances being compared with the less than function, or used as limits in the DvInterval class. Data value types which are to be used as limits must inherit from this class, and implement the function is_strictly_comparable_to to ensure that instances compare meaningfully. For example, instances of DvQuantity can only be compared if they measure the same kind of physical quantity.

Inherits From: mlhim2:DvAny
Abstract: True

Element Name	normal_range
Datatype	mlhim2:DvInterval
minOccurs	0
maxOccurs	1
Note	Optional normal range.

Element Name	other_reference_ranges
Datatype	mlhim2:ReferenceRange
minOccurs	0
maxOccurs	1
Note	List of ReferenceRanges. Optional tagged other reference ranges for this value in its particular measurement context.

Element Name	normal_status
Datatype	xs:string
minOccurs	0
maxOccurs	1
Note	Optional normal status indicator of value with respect to normal range for this value. Often included by lab, even if the normal range itself is not included. Coded by ordinals in series HHH, HH, H, (nothing), L, LL, LLL, etc.

DvOrdinal

Models rankings and scores, e.g. pain, Apgar values, etc, where there is a) implied ordering, b) no implication that the distance between each value is constant, and c) the total number of values is finite. Note that although the term 'ordinal' in mathematics means natural numbers only, here any integer is allowed, since negative and zero values are often used by medical professionals for values around a neutral point.

Examples of sets of ordinal values:

-3, -2, -1, 0, 1, 2, 3 -- reflex response values

0, 1, 2 -- Apgar values

Used for recording any clinical datum which is customarily recorded using symbolic values. Example: the results on a urinalysis strip, e.g. {neg, trace, +,++, +++} are used for leucocytes, protein, nitrites etc; for non-haemolysed blood {neg, trace, moderate}; for haemolysed blood {neg, trace,small, moderate, large}.

Inherits From: mlhim2:DvOrdered

Abstract:False

Element Name	dv
Datatype	xs:integer
minOccurs	0
maxOccurs	1
Note	Value in ordered enumeration of values. Any integer value can be used.

Element Name	symbol
Datatype	xs:string
minOccurs	0
maxOccurs	1
Note	Coded textual representation of this value in the enumeration, which may be strings made from "+" symbols, or other enumerations of terms such as "mild", "moderate", "severe", or even the same number series as the values, e.g. "1", "2", "3".

DvQuantified

Abstract class defining the concept of true quantified values, i.e. values which are not only ordered, but which have a precise magnitude.

Inherits From: mlhim2:DvOrdered
Abstract: True

Element Name	magnitude
Datatype	xs:decimal
minOccurs	0
maxOccurs	1
Note	Numeric value of the quantity in canonical (i.e. single value) form. Implemented as constant, function or attribute in subtypes as appropriate. The type Ordered_numeric is mapped to the available appropriate type in each implementation technology.

Element Name	magnitude_status
Datatype	xs:string
minOccurs	0
maxOccurs	1
Note	Optional status of magnitude with values: • "=" : magnitude is a point value • "<" : value is < magnitude • ">" : value is > magnitude • "<=" : value is <= magnitude • ">=" : value is >= magnitude • "~" : value is approximately magnitude If not present, meaning is "=".

Element Name	error
Datatype	xs:int
minOccurs	1
maxOccurs	1
Note	Error margin of measurement, indicating error in the recording method or instrument (+/- %). Implemented in subtypes. A logical value of 0 indicates 100% accuracy, i.e. no error.

Element Name	accuracy
Datatype	xs:decimal
minOccurs	1
maxOccurs	1
Note	Accuracy of the value in the magnitude attribute. 0% to +/- 100% A value of 0 means that the accuracy is unknown.

DvQuantity

Quantitified type representing "scientific" quantities, i.e. quantities expressed as a magnitude and units. Units were inspired by the Unified Code for Units of Measure (UCUM), developed by Gunther Schadow and Clement J. McDonald of The Regenstrief Institute. Can also be used for time durations, where it is more convenient to treat these as simply a number of individual seconds, minutes, hours, days, months, years, etc.

Inherits From: mlhim2:DvQuantified

Abstract: False

Element Name	units
Datatype	xs:normalizedString
minOccurs	1
maxOccurs	1
Note	Units expressed as a string in UCUM unit syntax, e.g. "kg/m2", “mm[Hg]", "ms-1", "km/h". Implemented accordingly in subtypes.

DvCount

Countable quantities. Used for countable types such as pregnancies and steps (taken by a physiotherapy patient), number of cigarettes smoked in a day, etc.

Misuse: Not used for amounts of physical entities (which all have units)

Inherits From: mlhim2:DvQuantified
Abstract: False

Element Name	count
Datatype	xs:integer
minOccurs	1
maxOccurs	1
Note	Number of items counted.

DvRatio

Models a ratio of values, i.e. where the numerator and denominator are both pure numbers.

Inherits From: mlhim2:DvQuantified

Abstract: True

Element Name	numerator
Datatype	xs:decimal
minOccurs	1
maxOccurs	1
Note	numerator of ratio

Element Name	denominator
Datatype	xs:decimal
minOccurs	1
maxOccurs	1
Note	denominator of ratio

DvRate

Models a ratio of values, i.e. where the numerator and denominator are both pure numbers, and the numerator is not contained (it is not a subset of the denominator). Example 1: Numerator = Number of episodes of seizures; Denominator = Number of days Example 2 = Number of hospital admissions; Denominator = Number of bed-days

Inherits From: mlhim2:DvRatio

Abstract: False

Element Name	rate_type
Datatype	xs:normalizedString
minOccurs	1
maxOccurs	1
Note	Indicates semantic type of coefficient: pk_coefficient = coefficient type. Numerator and denominator may be any value. pk_unitary = Denominator must be 1. pk_per10^n = Denominator is 10^2, numerator is understood as a real number divided by an exponent of 10 (10^n). pk_fraction = Numerator and denominator are real numbers, allowing rational and irrational fractions, and the presentation method uses a slash, e.g. “1/2”; if the numerator is greater than the denominator, e.g. n=3, d=2, the presentation is “1 1/2”.

DvProportion

Models a ratio of values, i.e. where the numerator and denominator are both pure numbers. The proportion_kind property is used to control the type attribute to be one of a defined set. Used for recording titers (e.g. 1:128), concentration ratios, e.g. Na:K (unitary denominator),

albumin:creatinine ratio, and percentages, e.g. red cell distirbution width (RDW). Should not be used to represent things like blood pressure which are often written using a '/' character, giving the misleading impression that the item is a ratio, when in fact it is a structured value. Similarly, visual acuity, often written as (e.g.) "6/24" in clinical notes is not a ratio but an ordinal (which includes non-numeric symbols like CF = count fingers etc). Should not be used for formulations.

Inherits From: mlhim2:DvRatio

Abstract: False

Element Name	proportion_type
Datatype	xs:normalizedString
minOccurs	1
maxOccurs	1
Note	Indicates semantic type of proportion: pk_proportion = proportion type. Numerator and denominator may be any value. pk_unitary = Denominator must be 1. pk_per10^n = Denominator is 10^2, numerator is understood as a real number divided by an exponent of 10 (10^n). pk_fraction = Numerator and denominator are real numbers, allowing rational and irrational fractions, and the presentation method uses a slash, e.g. "1/2"; if the numerator is greater than the denominator, e.g. n=3, d=2, the presentation is "1 1/2".

DvTemporal

Abstract base class for all temporal types.

Inherits From: mlhim2:DvAny

Abstract: True

DvDateTime

Represents an absolute point in time, specified to the second. Used for recording a precise point in real world time, and for approximate time stamps, e.g. the origin of a History in an Observation which is only partially known. All dates and times are assumed to be in the "current era"; somewhere between 0001-01-01T00:00:00Z and 9999-12-31T23:59:59Z AD.

Inherits From: mlhim2:DvTemporal

Abstract: False

Element Name	dv
Datatype	xs:dateTime
minOccurs	0
maxOccurs	1
Note	ISO8601:2004 date/time string including UTC offset. [date]T[time]Z

DvDate

Represents an absolute point in time, specified to the day.

Inherits From: mlhim2:DvTemporal
Abstract: False

Element Name	dv
Datatype	xs:date
minOccurs	0
maxOccurs	1
Note	ISO8601:2004 date string

DvTime

Represents an absolute point in time, specified to the second.

Inherits From: mlhim2:DvTemporal
Abstract: False

Element Name	dv
Datatype	xs:time
minOccurs	0
maxOccurs	1
Note	ISO8601:2004 time string

DvDuration

Durations are a component of time intervals and define the amount of intervening time in a time interval. They should only be used as part of a time interval as prescribed by the standard. Durations are represented by the format P[n]Y[n]M[n]DT[n]H[n]M[n]S or P[n]W as shown to the right. In these representations, the [n] is replaced by the value for each of the date and time elements that follow the [n]. Leading zeros are not required, but the maximum number of digits for each element should be agreed to by the communicating parties. The capital letters 'P', 'Y', 'M', 'W', 'D', 'T', 'H', 'M', and 'S' are designators for each of the date and time elements and are not replaced.

* P is the duration designator (historically called "period") placed at the start of the duration representation.

* Y is the year designator that follows the value for the number of years.

* M is the month designator that follows the value for the number of months.

* W is the week designator that follows the value for the number of weeks.

* D is the day designator that follows the value for the number of days.

* T is the time designator that precedes the time components of the representation.

* H is the hour designator that follows the value for the number of hours.

* M is the minute designator that follows the value for the number of minutes.

* S is the second designator that follows the value for the number of seconds.

For example, "P3Y6M4DT12H30M5S" represents a duration of "three years, six months, four days, twelve hours, thirty minutes, and five seconds". Date and time elements including their designator may be omitted if their value is zero, and lower order elements may also be omitted for reduced precision. For example, "P23DT23H" and "P4Y" are both acceptable duration representations. To resolve ambiguity, "P1M" is a one-month duration and "PT1M" is a one-minute duration (note the time designator, T, that precedes the time value). The smallest value used may also have a decimal fraction, as in "P0.5Y" to indicate half a year. This decimal fraction may be specified with either a comma or a full stop, as in "P0,5Y" or "P0.5Y". The standard does not prohibit date and time values in a duration representation from exceeding their "carry-over points" except as noted below. Thus, "PT36H" could be used as well as "P1DT12H" for representing the same duration. Alternately, a format for duration based on combined date and time representations may be used by agreement between the communicating parties either in the basic format PYYYYMMDDThhmmss or in the extended format P[YYYY]-[MM]-[DD]T[hh]:[mm]:[ss]. For example, the first duration shown above would be "P0003-06-04T12:30:05". However, individual date and time values cannot exceed their moduli (e.g. a value of 13 for the month or 25 for the hour would not be permissible).

Inherits From: mlhim2:DvTemporal

Abstract: False

Element Name	dv
Datatype	xs:duration
minOccurs	0
maxOccurs	1
Note	The duration in the form of a string.

DvNumeric

Abstract base for all numeric types in MLHIM.

Inherits From: mlhim2:DvAny

Abstract: True

DvDecimal

Represents a decimal number of arbitrary precision. Schema processors vary in the number of significant digits they support, but a conforming processor must support a minimum of 18 significant digits. The format of `xsd:decimal` is a sequence of digits optionally preceded by a sign ("+" or "-") and optionally containing a period. The value may start or end with a period. If the fractional part is 0 then the period and trailing zeros may be omitted. Leading and trailing zeros are permitted, but they are not considered significant. That is, the decimal values 3.0 and 3.0000 are considered equal.

Inherits From: mlhim2:DvNumeric
Abstract: False

Element Name	dv
Datatype	xs:decimal
minOccurs	0
maxOccurs	1
Note	Data value for this class.

DvUnsignedLong

Represents an integer between 0 and 18446744073709551615. An `xsd:unsignedLong` is a sequence of digits, optionally preceded by a + sign. Leading zeros are permitted, but decimal points are not.

Inherits From: mlhim2:DvNumeric
Abstract: False

Element Name	dv
Datatype	xs:unsignedLong
minOccurs	0
maxOccurs	1
Note	Data value for this class.

DvShort

Represents an integer between -32768 and 32767. An `xsd:short` is a sequence of digits,

optionally preceded by a + or - sign. Leading zeros are permitted, but decimal points are not.

Inherits From: mlhim2:DvNumeric
Abstract: False

Element Name	dv
Datatype	xs:short
minOccurs	0
maxOccurs	1
Note	Data value for this class.

DvPositiveInteger

Represents an arbitrarily large positive integer. An `xsd:positiveInteger` is a sequence of digits, optionally preceded by a + sign. Leading zeros are permitted, but decimal points are not.

Inherits From: mlhim2:DvNumeric
Abstract: False

Element Name	dv
Datatype	xs:positiveInteger
minOccurs	0
maxOccurs	1
Note	Data value for this class.

DvByte

Represents an integer between -128 and 127. An `xsd:byte` is a sequence of digits, optionally preceded by a + or - sign. Leading zeros are permitted, but decimal points are not.

Inherits From: mlhim2:DvNumeric
Abstract: False

Element Name	dv
Datatype	xs:byte
minOccurs	0
maxOccurs	1
Note	Data value for this class.

DvNonPositiveInteger

Represents an arbitrarily large non-positive integer. An `xsd:nonPositiveInteger` is a sequence of digits, optionally preceded by a - sign. Leading zeros are permitted, but decimal points are not.

Inherits From: mlhim2:DvNumeric
Abstract: False

Element Name	dv
Datatype	xs:nonPositiveInteger
minOccurs	0
maxOccurs	1
Note	Data value for this class.

DvNonNegativeInteger

Represents an arbitrarily large non-negative integer. An `xsd:nonNegativeInteger` is a sequence of digits, optionally preceded by a + sign. Leading zeros are permitted, but decimal points are not.

Inherits From: mlhim2:DvNumeric
Abstract: False

Element Name	dv
Datatype	xs:nonNegativeInteger
minOccurs	0
maxOccurs	1
Note	Data value for this class.

DvInt

Represents an integer between -2147483648 and 2147483647. An `xsd:int` is a sequence of digits, optionally preceded by a + or - sign. Leading zeros are permitted, but decimal points are not.

Inherits From: mlhim2:DvNumeric
Abstract: False

Element Name	dv
Datatype	xs:int
minOccurs	0
maxOccurs	1
Note	Data value for this class.

DvNegativeInteger

Represents an arbitrarily large negative integer. An `xsd:negativeInteger` is a sequence of digits, preceded by a - sign. Leading zeros are permitted, but decimal points are not.

Inherits From: mlhim2:DvNumeric
Abstract: False

Element Name	dv
Datatype	xs:negativeInteger
minOccurs	0
maxOccurs	1
Note	Data value for this class.

DvInteger

Represents an arbitrarily large integer, from which twelve other built-in integer types are derived (directly or indirectly). An `xsd:integer` is a sequence of digits, optionally preceded by a + or - sign. Leading zeros are permitted, but decimal points are not.

Inherits From: mlhim2:DvNumeric
Abstract: False

Element Name	dv
Datatype	xs:integer
minOccurs	0
maxOccurs	1
Note	Data value for this class.

DvUnsignedInt

Represents an integer between 0 and 4294967295. An `xsd:unsignedInt` is a sequence of digits, optionally preceded by a + sign. Leading zeros are permitted, but decimal points are not.

Inherits From: mlhim2:DvNumeric
Abstract: False

Element Name	dv
Datatype	xs:unsignedInt
minOccurs	0
maxOccurs	1
Note	Data value for this class.

DvUnsignedShort

Represents an integer between -32768 and 32767. An `xsd:short` is a sequence of digits, optionally preceded by a + or - sign. Leading zeros are permitted, but decimal points are not.

Inherits From: mlhim2:DvNumeric
Abstract: False

Element Name	dv
Datatype	xs:unsignedShort
minOccurs	0
maxOccurs	1
Note	Data value for this class.

DvLong

Represents an integer between -9223372036854775808 and 9223372036854775807. An `xsd:long` is a sequence of digits, optionally preceded by a + or - sign. Leading zeros are permitted, but decimal points are not.

Inherits From: mlhim2:DvNumeric
Abstract: False

Element Name	dv
Datatype	xs:long
minOccurs	0
maxOccurs	1
Note	Data value for this class.

DvUnsignedByte

Represents an integer between 0 and 255. An `xsd:unsignedByte` is a sequence of digits, optionally preceded by a + sign. Leading zeros are permitted, but decimal points are not.

Inherits From: mlhim2:DvNumeric
Abstract: False

Element Name	dv
Datatype	xs:unsignedByte
minOccurs	0
maxOccurs	1
Note	Data value for this class.

Common

PartyProxy

Abstract concept of a proxy description of a party, including an optional link to data for this party in a demographic or other identity management system. Sub-typed into PartyIdentified

and PartySelf.

Inherits From: mlhim2:DvAny
Abstract: True

Element Name	external_ref
Datatype	mlhim2:EntityRef
minOccurs	0
maxOccurs	1
Note	Optional reference to more detailed demographic or identification information for this party, in an external system.

PartySelf

Party proxy representing the subject of the record. Used to indicate that the party is the owner of the record. May or may not have external_ref set.

Inherits From: mlhim2:PartyProxy
Abstract: False

PartyIdentified

Proxy data for an identified party other than the subject of the record, minimally consisting of human-readable identifier(s), such as name, formal (and possibly computable) identifiers such as NHS number, and an optional link to external data.

There must be at least one of name, identifier or external_ref present.Used to describe parties where only identifiers may be known, and there is no entry at all in the demographic system (or even no demographic system). Typically for health care providers, e.g. name and provider number of an institution. Should not be used to include patient identifying information.

Inherits From: mlhim2:PartyProxy
Abstract: False

Element Name	name
Datatype	mlhim2:DvString
minOccurs	0
maxOccurs	1
Note	Optional human-readable name (in String form).

Element Name	identifiers
Datatype	mlhim2:DvIdentifier
minOccurs	0
maxOccurs	unbounded
Note	List of DvIdentifiers - One or more formal identifiers (possibly computable).

PartyRelated

Proxy type for identifying a party and its relationship to the subject of the record. Use where the relationship between the party and the subject of the record must be known.

Inherits From: mlhim2:PartyIdentified
Abstract: False

Element Name	relationship
Datatype	mlhim2:Relationship
minOccurs	1
maxOccurs	1
Note	Relationship of subject of this Entry to the subject of the record. May be coded. If it is the patient, coded as “self”.

FeederAuditDetails

Audit details for any system in a feeder system chain. Audit details here means the general notion of who/where/when the information item to which the audit is attached was created. None of the attributes is defined as mandatory, however, in different scenarios, various combinations of attributes will usually be mandatory. This can be controlled by specifying feeder audit details in legacy archetypes.

Inherits From: xs:anyType
Abstract: False

Element Name	system_id
Datatype	mlhim2:DvIdentifier
minOccurs	1
maxOccurs	1
Note	Identifier of the system which handled the information item.

Element Name	version_id
Datatype	mlhim2:DvNormalizedString
minOccurs	0
maxOccurs	1
Note	Any identifier used in the system such as “interim”, “final”, or numeric versions if available.

Element Name	provider
Datatype	mlhim2:PartyIdentified
minOccurs	0
maxOccurs	1
Note	Optional provider(s) who created, committed, forwarded or otherwise handled the item.

Element Name	location
Datatype	mlhim2:Location
minOccurs	0
maxOccurs	1
Note	Identifier of the particular site/facility within an organization which handled the item. For computability, this identifier needs to be e.g. a PKI identifier which can be included in the identifier list of the PartyIdentified object.

Element Name	time
Datatype	mlhim2:DvTemporal
minOccurs	0
maxOccurs	1
Note	Time of handling the item. For an originating system, this will be time of creation, for an intermediate feeder system, this will be a time of accession or other time of handling, where available.

Element Name	subject
Datatype	mlhim2:PartyProxy
minOccurs	0
maxOccurs	unbounded
Note	Subject(s) of the received information item.

FeederAudit

Audit and other meta-data for systems in the feeder chain.
Inherits From: xs:anyType
Abstract: False

Element Name	originating_system_audit
Datatype	mlhim2:FeederAuditDetails
minOccurs	1
maxOccurs	1
Note	Any audit information for the information item from the originating system.

Element Name	originating_system_item_ids
Datatype	mlhim2:DvIdentifier
minOccurs	1
maxOccurs	unbounded
Note	Identifiers used for the item in the originating system, e.g. filler and placer ids.

Element Name	feeder_system_audit
Datatype	mlhim2:FeederAuditDetails
minOccurs	0
maxOccurs	1
Note	Any audit information for the information item from the feeder system, if different from the originating system.

Element Name	feeder_system_ids
Datatype	mlhim2:DvIdentifier
minOccurs	0
maxOccurs	unbounded
Note	Identifiers used for the item in the feeder system, where the feeder system is distinct from the originating system.

Element Name	original_content
Datatype	mlhim2:DvEncapsulated
minOccurs	1
maxOccurs	1
Note	Optional inline inclusion of or reference to original content corresponding to the MLHIM content at this node. Typically a URI reference to a document or message in a persistent store associated with the EHR.

Locatable

Root class of all information model classes that can be expressed in a constraint model.

Inherits From: mlhim2:DvAny
Abstract: True

Element Name	feeder_audit
Datatype	mlhim2:FeederAudit
minOccurs	0
maxOccurs	1
Note	Audit trail from non-MLHIM system of original commit of information forming the content of this node, or from a conversion gateway which has synthesized this node.

Element Name	name
Datatype	mlhim2:DvNormalizedString
minOccurs	1
maxOccurs	1
Note	This is the term provided at design time to name this EHR construct. Its retention in the EHR faithfully preserves the original label by which this entry was known to end users. When created in English; it shall consist only of lowercase letters a-z, digits 0-9 and use the _ (underscore) as a separator. Other languages should follow similar naming rules in order to not conflict with implementations. These names are defined as part of the ontology and may be translated.

Element Name	uuid
Datatype	mlhim2:DvToken
minOccurs	1
maxOccurs	1
Note	UUID for each Locatable structure. These UUIDs are used to construct runtime paths.

Element Name	parent
Datatype	mlhim2:DvToken
minOccurs	0
maxOccurs	1
Note	Parent of this node in compositional hierarchy. Usually a UUID of another Locatable.

Folder

A container for other items representing a logical grouping. In implementations this may also inherit from library components providing the required functionality.

Inherits From: mlhim2:Locatable
Abstract: False

Element Name	items
Datatype	mlhim2:Composition
minOccurs	0
maxOccurs	unbounded
Note	A list of UUIDs of Compositions logically in this folder.

Attestation

Record an attestation by a party of item(s) of record content. The type of attestation is recorded by the reason attribute, which may be coded.

Inherits From: mlhim2:Locatable
Abstract: False

Element Name	attested_view
Datatype	mlhim2:DvMedia
minOccurs	0
maxOccurs	1
Note	Optional visual representation of content attested e.g. screen image.

Element Name	proof
Datatype	mlhim2:DvParsable
minOccurs	1
maxOccurs	1
Note	Proof of attestation such as an GPG signature.

Element Name	reason
Datatype	mlhim2:DvCodedString
minOccurs	1
maxOccurs	1
Note	Reason of this attestation. Optionally coded by the MLHIM Terminology group “attestation reason”; includes values like “authorization”, “witness” etc.

Element Name	commiter
Datatype	mlhim2:PartyProxy
minOccurs	1
maxOccurs	1
Note	Identity and optional reference into identity management service, of user who committed the item.

Element Name	time_commited
Datatype	mlhim2:DvTemporal
minOccurs	1
maxOccurs	1
Note	Time of committal of the item.

Element Name	is_pending
Datatype	mlhim2:DvBoolean
minOccurs	1
maxOccurs	1
Note	True if this attestation is outstanding; False means it has been completed.

Participation

Model of a participation of a Party (any Actor or Role) in an activity. Used to represent any participation of a Party in some activity, which is not explicitly in the model, e.g. assisting nurse. Can be used to record past or future participations. Should not be used in place of more permanent relationships between demographic entities.

Inherits From: xs:anyType
Abstract: False

Element Name	performer
Datatype	mlhim2:PartyProxy
minOccurs	1
maxOccurs	1
Note	The id and possibly demographic system link of the party participating in the activity.

Element Name	function
Datatype	mlhim2:DvCodedString
minOccurs	1
maxOccurs	1
Note	The function of the Party in this participation (note that a given party might participate in more than one way in a particular activity). This attribute should be coded.

Element Name	mode
Datatype	mlhim2:DvCodedString
minOccurs	1
maxOccurs	1
Note	The mode of the performer / activity interaction, e.g. present, by telephone, by email etc.

Element Name	time
Datatype	mlhim2:DvTemporal
minOccurs	1
maxOccurs	1
Note	The time interval during which the participation took place, if it is used in an observational context (i.e. recording facts about the past); or the intended time interval of the participation when used in future contexts, such as EHR Instructions.

Link

The Link type defines a logical relationship between two items, such as two Entry types or an Entry and a Composition.

Links can be used across compositions, and across EHRs. Links can potentially be used between interior nodes, although this probably should be prevented in CCDs. Multiple Links can be attached to the root object of any structure to give the effect of a 1->N link. 1:1 and 1:N relationships between content elements can be expressed by using one, or more than one. Chains of links can be used to see "problem threads" or other logical groupings of items. Links should be between structures only, i.e. between objects representing complete domain concepts because relationships between sub-elements of whole concepts are not necessarily meaningful, and may be downright confusing.

Inherits From: xs:anyType
Abstract: False

Element Name	relation
Datatype	mlhim2:Relationship
minOccurs	1
maxOccurs	1
Note	The relation attribute is used to indicate a clinical or domain-level meaning for the kind of link. This attribute should describe the relationship with the target object. Normally this relationship is found in the semantic relations found in the NLM Semantic Network.

Element Name	target
Datatype	mlhim2:DvToken
minOccurs	1
maxOccurs	1
Note	The target uuid of this Link.

Structures

DataStructure

Abstract parent class of all data structure types.

Inherits From: mlhim2:Locatable
Abstract: True

ItemStructure

Abstract parent class of all item structure types.

Inherits From: mlhim2:DataStructure
Abstract: True

Event

Defines the abstract notion of a single event in a series. This class is generic, allowing types to be generated which are locked to particular spatial types. Subtypes express point or interval data.

Inherits From: mlhim2:Locatable
Abstract: True

Element Name	time
Datatype	mlhim2:DvTemporal
minOccurs	1
maxOccurs	1
Note	Time of this event. If the width is non-zero, it is the time point of the trailing edge of the event.

Element Name	data
Datatype	mlhim2:DvAny
minOccurs	1
maxOccurs	1
Note	The data of this event.

Element Name	state
Datatype	mlhim2:ItemStructure
minOccurs	0
maxOccurs	1
Note	Optional state data for this event.

PointEvent

Defines a single point event in a series.

Inherits From: mlhim2:Event
Abstract: False

IntervalEvent

Defines a single interval event in a series.

Inherits From: mlhim2:Event
Abstract: False

Element Name	width
Datatype	mlhim2:DvDuration
minOccurs	1
maxOccurs	1
Note	Length of the interval during which the state was true.

Element Name	math_function
Datatype	mlhim2:DvCodedString
minOccurs	1
maxOccurs	1
Note	Mathematical function of the data of this event, e.g. “maximum”, “mean” etc.

Element Name	sample_count
Datatype	mlhim2:DvCount
minOccurs	1
maxOccurs	1
Note	Count of original samples to which this event corresponds.

Item

The abstract parent of Slot, Cluster and Element representation classes.

Inherits From: mlhim2:Locatable
Abstract: True

Slot

A structure allowing the inclusion of one CCD inside a CCD. An unbounded list of allowable CCDs to choose from should be available at runtime.

Inherits From: mlhim2:Item
Abstract: False

Element Name	ccd
Datatype	mlhim2:DvToken
minOccurs	1
maxOccurs	1
Note	The CCD ID selected at run-time.

Element Name	allowed_ccds
Datatype	mlhim2:DvToken
minOccurs	1
maxOccurs	unbounded
Note	A list of allowed CCD IDs of which one is selected at runtime.

Element

The leaf variant of Item, to which a DvAny instance is attached.

Inherits From: mlhim2:Item
Abstract: False

Element Name	dv
Datatype	mlhim2:DvAny
minOccurs	1
maxOccurs	1
Note	Data value of this leaf

Cluster

The grouping variant of Item, which may contain further instances of Item, in an ordered list.

Inherits From: mlhim2:Item
Abstract: False

Element Name	items
Datatype	mlhim2:Item
minOccurs	1
maxOccurs	unbounded
Note	List of Items.

History

Root object of a linear history, i.e. time series structure. For a periodic series of events, period will be set, and the time of each Event in the History must correspond; i.e. the Event.offset must be a multiple of period for each Event. Missing events in a period History are however allowed.

Inherits From: mlhim2:DataStructure
Abstract: False

Element Name	origin
Datatype	mlhim2:DvTemporal
minOccurs	1
maxOccurs	1
Note	Time origin of this event history. The first event is not necessarily at the origin point.

Element Name	events
Datatype	mlhim2:Event
minOccurs	1
maxOccurs	unbounded
Note	The events in the series.

Element Name	period
Datatype	mlhim2:DvDuration
minOccurs	0
maxOccurs	1
Note	Period between samples in this segment if periodic.

Element Name	duration
Datatype	mlhim2:DvDuration
minOccurs	0
maxOccurs	1
Note	Duration of the entire History; either corresponds to the duration of all the events, and/or the duration represented by the summary, if it exists.

Element Name	summary
Datatype	mlhim2:ItemStructure
minOccurs	0
maxOccurs	1
Note	Optional summary data expressing e.g. text or image which summarises entire History.

ItemSingle

Used to represent any data which is logically a single value, such as a person's height or weight.

Inherits From: mlhim2:ItemStructure
Abstract: False

Element Name	item
Datatype	mlhim2:Item
minOccurs	1
maxOccurs	1
Note	Element or Slot for this Item.

ItemList

Logical list data structure, where each item has a value and can be referred to by a name and a positional index in the list. The list may be empty.

Inherits From: mlhim2:ItemStructure
Abstract: False

Element Name	items
Datatype	mlhim2:Item
minOccurs	0
maxOccurs	unbounded
Note	List of Elements, Clusters or Slots for this Item.

ItemTable

Logical relational database style table data structure, in which columns are named and ordered with respect to each other. Implemented using Cluster-per-row encoding. Each row Cluster must have an identical number of Elements, each of which in turn must have identical names and value types in the corresponding postions in each row. Some columns may be designated 'key' columns, containing key data for each row, in the manner of relational tables. This allows row-naming, where each row represents a body site, a blood antigen etc.

All values in a column have the same data type.

Inherits From: mlhim2:ItemStructure
Abstract: False

Element Name	rows
Datatype	mlhim2:Cluster
minOccurs	0
maxOccurs	unbounded
Note	List of Clusters for this Item.

ItemTree

Logical tree data structure. The tree may be empty. Used to represent data which are logically a tree such as audiology results, microbiology results, biochemistry results.

Inherits From: mlhim2:ItemStructure
Abstract: False

Element Name	items
Datatype	mlhim2:ItemStructure
minOccurs	1
maxOccurs	unbounded
Note	List of structures to define the root of the tree.

Content

Activity

Defines a single activity within an Instruction, such as a medication administration.

Inherits From: mlhim2:Locatable
Abstract: False

Element Name	description
Datatype	mlhim2:ItemStructure
minOccurs	1
maxOccurs	1
Note	Description of the activity, in the form of a structure.

Element Name	timing
Datatype	mlhim2:DvParsable
minOccurs	1
maxOccurs	1
Note	Timing of the activity, in the form of a parsable string, such as an HL7 GTS or ISO8601 string.

Element Name	actions
Datatype	mlhim2:DvToken
minOccurs	1
maxOccurs	unbounded
Note	List of CCD UUIDs (in order) to be executed for this activity.

EventContext

Documents the context information of a healthcare event involving the subject of care and the health system. The context information recorded here is independent of the attributes recorded in the version audit, which document the "system interaction" context, i.e. the context of a user interacting with the health record system. Healthcare events include patient contacts, and any other business activity, such as pathology investigations which take place on behalf of the patient.

Inherits From: mlhim2:Locatable
Abstract: False

Element Name	healthcare_facility
Datatype	mlhim2:Organization
minOccurs	0
maxOccurs	1
Note	The health care facility under whose care the event took place. This is the most specific workgroup or delivery unit within a care delivery enterprise that has an official identifier in the health system, and can be used to ensure medico-legal accountability.

Element Name	start_time
Datatype	mlhim2:DvTemporal
minOccurs	1
maxOccurs	1
Note	Start time of the clinical session or other kind of event during which a provider performs a service of any kind for the patient.

Element Name	end_time
Datatype	mlhim2:DvTemporal
minOccurs	0
maxOccurs	1
Note	Optional end time of the clinical session.

Element Name	participation
Datatype	mlhim2:Participation
minOccurs	0
maxOccurs	1
Note	Parties involved in the healthcare event.

Element Name	location
Datatype	mlhim2:Location
minOccurs	0
maxOccurs	1
Note	The actual location where the session occurred, e.g. “microbiol lab 2”, “home”, “ward A3” and so on.

Element Name	setting
Datatype	mlhim2:DvCodedString
minOccurs	0
maxOccurs	1
Note	The setting in which the clinical session took place.

ContentItem

Abstract ancestor of all concrete content types.

Inherits From: mlhim2:Locatable
Abstract: True

Element Name	links
Datatype	mlhim2:Link
minOccurs	0
maxOccurs	unbounded
Note	A list of links to other content items.

Element Name	attestation
Datatype	mlhim2:Attestation
minOccurs	0
maxOccurs	1
Note	Sign off on completeness and accuracy of the content.

Section

Represents a heading in a heading structure, or "section tree". Created according to structures for typical headings such as SOAP, physical examination, but also pathology result heading structures. Should not be used instead of Entry hierarchical structures.

Element Name	items
Datatype	mlhim2:ContentItem
minOccurs	0
maxOccurs	1
Note	Ordered list of content items under this section, which may include more Section or Entry types.

Entry

The abstract parent of all ENTRY subtypes. An ENTRY is the root of a logical item of "hard" clinical information created in the "clinical statement" context, within a clinical session. There can be numerous such contexts in a clinical session. Observations and other Entry types only ever document information captured/created in the event documented by the enclosing Composition. An Entry is also the minimal unit of information any query should return, since a whole Entry (including sub-parts) records spatial structure, timing information, and contextual information, as well as the subject and generator of the information.

Inherits From: mlhim2:ContentItem
Abstract: True

Element Name	language
Datatype	mlhim2:DvLanguage
minOccurs	1
maxOccurs	1
Note	Mandatory indicator of the localised language in which this Entry is written.

Element Name	encoding
Datatype	mlhim2:DvCodedString
minOccurs	0
maxOccurs	1
Note	Name of character set in which text values in this Entry are encoded.

Element Name	subject
Datatype	mlhim2:PartyProxy
minOccurs	1
maxOccurs	1
Note	Id of human subject of this Entry, e.g.: organ donor, foetus, family member, another clinically relevant person.

Element Name	provider
Datatype	mlhim2:PartyProxy
minOccurs	0
maxOccurs	1
Note	Optional identification of provider of the information in this Entry, which might be: the patient, a patient agent, e.g. parent, guardian, the clinician, a device or software. Generally only used when the recorder needs to make it explicit. Otherwise, Composition composer and other participants are assumed.

Element Name	other_participations
Datatype	mlhim2:Participation
minOccurs	0
maxOccurs	unbounded
Note	Other participations at Entry level.

Element Name	workflow_id
Datatype	mlhim2:ObjectRef
minOccurs	
maxOccurs	
Note	Identifier of externally held workflow engine data for this workflow execution, for this subject of care.

AdminEntry

Entry subtype for administrative information, i.e. information about setting up the clinical process, but not itself clinically relevant. CCDs will define contained information. Used for admistrative details of admission, episode, ward location, discharge, appointment (if not stored in a practice management or appointments system). Not used for any clinically significant information.

Inherits From: mlhim2:Entry
Abstract: False

Element Name	data
Datatype	mlhim2:ItemStructure
minOccurs	1
maxOccurs	1
Note	The data of the Entry.

CareEntry

The abstract parent of all clinical Entry subtypes. Defines protocol and guideline attributes for all clinical Entry subtypes.

Inherits From: mlhim2:Entry
Abstract: True

Element Name	protocol
Datatype	mlhim2:ItemStructure
minOccurs	0
maxOccurs	1
Note	Description of the method (i.e. how) the information in this entry was arrived at. For Observations, this is a description of the method or instrument used. For Evaluations, how the evaluation was arrived at. For Instructions, how to execute the Instruction. This may take the form of references to guidelines, including manually followed and executable; knowledge references such as a paper in Medline; clinical reasons within a larger care process.<

Element Name	guideline_id
Datatype	mlhim2:ObjectRef
minOccurs	0
maxOccurs	1
Note	Optional external identifier of guideline creating this action if relevant.

Observation

Entry subtype for all clinical data in the past or present, i.e. which (by the time it is recorded) has already occurred. Observation data is expressed using the class History, which guarantees that it is situated in time. Observation is used for all notionally objective (i.e. measured in some way) observations of phenomena, and patient-reported phenomena, e.g. pain. Not used for recording opinion or future statements of any kind, including instructions, intentions, plans etc.

Inherits From: mlhim2:CareEntry
Abstract: False

Element Name	data
Datatype	mlhim2:History
minOccurs	1
maxOccurs	1
Note	The data of this observation, in the form of a history of values which may be of any complexity.

Element Name	state
Datatype	mlhim2:History
minOccurs	0
maxOccurs	1
Note	Optional recording of the state of subject of this observation during the observation process, in the form of a separate history of values which may be of any complexity. State may also be recorded within the History of the data attribute.

Evaluation

Used for all kinds of statements which evaluate other information, such as interpretations of obvservations, diagnoses, differential diagnoses, hypotheses, risk assessments, goals and plans.

Inherits From: mlhim2:CareEntry
Abstract: False

Element Name	data
Datatype	mlhim2:ItemStructure"
minOccurs	1
maxOccurs	1
Note	The data from the evaluation.

Action

Used to record a clinical action that has been performed, which may have been adhoc, or due to the execution of an Activity in an Instruction workflow. Every Action corresponds to a careflow step of some kind or another.

Inherits From: mlhim2:CareEntry
Abstract: False

Element Name	time
Datatype	mlhim2:DvTemporal
minOccurs	0
maxOccurs	1
Note	Point in time at which this action completed.<

Element Name	description
Datatype	mlhim2:ItemStructure
minOccurs	1
maxOccurs	1
Note	Description of the activity to be performed, in the form of a defined structure.

Element Name	ism_transition
Datatype	mlhim2:DvCodedString
minOccurs	0
maxOccurs	1
Note	Details of transition in the Instruction state machine caused by this Action.

Element Name	instruction_details
Datatype	mlhim2:DvCodedString
minOccurs	0
maxOccurs	1
Note	Details to of the Instruction that caused this Action to be performed, if there was one.

Instruction

Used for any actionable statement such as medication and therapeutic orders, monitoring, recall and review. Enough details must be provided for the specification to be directly executed by an actor, either human or machine.

Inherits From: mlhim2:CareEntry
Abstract: False

Element Name	expiry_time
Datatype	mlhim2:DvTemporal
minOccurs	0
maxOccurs	1
Note	Optional expiry date/time to assist determination of when an Instruction can be assumed to have expired. This helps prevent false listing of Instructions as Active when they clearly must have been terminated in some way or other.

Element Name	narrative
Datatype	mlhim2:DvString
minOccurs	1
maxOccurs	1
Note	Mandatory human-readable version of what the Instruction is about.

Element Name	wf_definition
Datatype	mlhim2:DvParsable
minOccurs	0
maxOccurs	1
Note	Optional workflow engine executable expression of the Instruction.

Element Name	activities
Datatype	mlhim2:Activity
minOccurs	0
maxOccurs	unbounded
Note	List of all activities in Instruction.

Composition

One version in a VersionedComposition. A composition is considered the unit of modification of the record, the unit of transmission in record extracts, and the unit of attestation by authorising clinicians. In this latter sense, it may be considered equivalent to a signed document.

Inherits From: mlhim2:Locatable
Abstract: False

Element Name	attestation
Datatype	mlhim2:Attestation
minOccurs	1
maxOccurs	1
Note	The signoff of this Composition.

Element Name	links
Datatype	mlhim2:Link
minOccurs	0
maxOccurs	unbounded
Note	A list of links to other relevant content items.

Element Name	original
Datatype	mlhim2:DvToken
minOccurs	0
maxOccurs	1
Note	The UUID of the original composition if this is a new version. Void if this is an original composition. This attribute along with the predecessor form the versioning function for compositions.

Element Name	predecessor
Datatype	mlhim2:DvToken
minOccurs	0
maxOccurs	1
Note	The UUID of the preceeding composition if this is a new version. Void if this is an original composition.

Element Name	created
Datatype	mlhim2:DvTemporal
minOccurs	1
maxOccurs	1
Note	DateTime created. Since Compositions are never modified, only copied and edited.

Element Name	composer
Datatype	mlhim2:PartyProxy
minOccurs	1
maxOccurs	1
Note	The person primarily responsible for the content of the Composition (but not necessarily its committal into the EHR system). This is the identifier which should appear on the screen. It may or may not be the person who entered the data. When it is the patient, the special “self” instance of PartyProxy will be used.

Element Name	territory
Datatype	mlhim2:Location
minOccurs	0
maxOccurs	1
Note	Name of territory in which this Composition was written.

Element Name	category
Datatype	mlhim2:DvCodedString
minOccurs	0
maxOccurs	1
Note	Indicates what broad category this Composition is belongs to, e.g. “persistent” - of longitudinal validity, “event”, “process”.

Element Name	language
Datatype	mlhim2:DvLanguage
minOccurs	1
maxOccurs	1
Note	Mandatory indicator of the localized language in which this Composition is written. The language of an Entry if different from the Composition is indicated in Entry.language.

Element Name	context
Datatype	mlhim2:EventContext
minOccurs	1
maxOccurs	1
Note	The clinical session context of this Composition, i.e. the contextual attributes of the clinical session.

Element Name	content
Datatype	mlhim2:Locatable
minOccurs	1
maxOccurs	1
Note	The content of this Composition. Any Locatable content items or structures may be added here.

Entity

Role

A role defines some activity that an entity may participate in.

Inherits From: mlhim2:Locatable

Abstract: False

Element Name	details
Datatype	mlhim2:ItemStructure
minOccurs	1
maxOccurs	1
Note	The details are a structural segment that may be used to create machine processable capabilities. For example in a workflow or authentication service.

Element Name	description
Datatype	mlhim2:DvString
minOccurs	1
maxOccurs	1
Note	A human readable description for this role.

Element Name	is_primary
Datatype	mlhim2:DvBoolean
minOccurs	1
maxOccurs	1
Note	A binary attribute indicating if this is the primary role in the list of roles.

Location

A location should be meaningful to human readers as well as contain machine processable components. See ISO 19115 The details attribute can be constrained to describe specific address details or any other physical world location information.

Inherits From: mlhim2:Locatable
Abstract: False

Element Name	details
Datatype	mlhim2:ItemStructure
minOccurs	1
maxOccurs	1
Note	The details are a structural segment that may be used to create machine processable capabilities. For example in a GIS it may contain longitude/latitude/altitude information.

Element Name	description
Datatype	mlhim2:DvString
minOccurs	1
maxOccurs	1
Note	A human readable description for this location.

Element Name	is_primary
Datatype	mlhim2:DvBoolean
minOccurs	1
maxOccurs	1
Note	A binary attribute indicating if this is the primary location in the list of locations.

Relationship

A relationship exists in a source and points to a target. Sources and Targets can be persons, places, or things. Relationships are locatable objects and they may only reference locatable

objects by uuid.

Inherits From: mlhim2:Locatable
Abstract: False

Element Name	targets
Datatype	mlhim2:DvToken
minOccurs	1
maxOccurs	unbounded
Note	The List of target UUIDs. Relationships can only be established between instances.

Party

Any type of real world living entity.

Inherits From: mlhim2:Locatable
Abstract: False

Element Name	valid_time_begin
Datatype	mlhim2:DvDateTime
minOccurs	0
maxOccurs	1
Note	If present this must be a valid datetime string including timezone.

Element Name	valid_time_end
Datatype	mlhim2:DvDateTime
minOccurs	0
maxOccurs	1
Note	If present this must be a valid datetime string including timezone.

Element Name	ev
Datatype	mlhim2:ExceptionalValue
minOccurs	0
maxOccurs	1
Note	The exceptional value. Often referred to as Null Flavour. <

Element Name	identifiers
Datatype	mlhim2:DvIdentifier
minOccurs	0
maxOccurs	unbounded
Note	A List of identifiers used to relate this entity to the real world.

Element Name	roles
Datatype	mlhim2:Role
minOccurs	0
maxOccurs	unbounded
Note	A List of roles that this entity may participate.

Element Name	locations
Datatype	mlhim2:Location
minOccurs	0
maxOccurs	unbounded
Note	A list of locations for this entity.

Element Name	relationships
Datatype	mlhim2:Relationship
minOccurs	0
maxOccurs	unbounded
Note	A list of relationships for this entity.

Device

Any one instance of an inanimate object.

Inherits From: mlhim2:Party
Abstract: False

Element Name	details
Datatype	mlhim2:ItemStructure
minOccurs	1
maxOccurs	1
Note	Structural details about the device

Organization

A legally defined entity allowed to operate as one unit of identity.

Inherits From: mlhim2:Party
Abstract: False

Element Name	details
Datatype	mlhim2:ItemStructure
minOccurs	1
maxOccurs	1
Note	Structural details about the organization

Group

A collection of persons existing as a conceptual entity for some period of time.

Inherits From: mlhim2:Party
Abstract: False

Element Name	members
Datatype	mlhim2:Party
minOccurs	0
maxOccurs	unbounded
Note	A Set of Partys composing the Group. Logically, only Party subclasses of the same type should be collected in a group.

Person

A singular human being.

Inherits From: mlhim2:Party
Abstract: False

Element Name	details
Datatype	mlhim2:ItemStructure
minOccurs	1
maxOccurs	1
Note	Structural details about the person<

NonHuman

Any nonhuman living animal or plant.

Inherits From: mlhim2:Party
Abstract: False

Element Name	details
Datatype	mlhim2:ItemStructure
minOccurs	1
maxOccurs	1
Note	Structural details about the non-human

Constraint

OntologyEntry

A set of optional entries for a specific node or nodes in this CCD.

Inherits From: xs:anyType

Element Name	contents
Datatype	xs:string
minOccurs	1
maxOccurs	unbounded
Note	A list of strings for this entry.

Ontology

The pre-coordinated ontology for this CCD. The contents of the ontology are created during design time by the CCD author.

The ontology provides the means to create a set of options to restrict data in the context of the CCD.

Inherits From: xs:anyType
Abstract: False

Element Name	names
Datatype	xs:string
minOccurs	0
maxOccurs	unbounded
Note	Contains the list of 'name' attributes in the CCD;in the original language. They are connected to the name points in the CCD via a uuid. This allows translations using the uuid as the GETTEXT msgid.

Element Name	internalVocabularies
Datatype	xs:string
minOccurs	0
maxOccurs	unbounded
Note	Internal vocabularies useful for options such as DvString.dv The DvString.dv attribute will contain the uuid link to the named element. This element will contain the options.

Element Name	lookupTables
Datatype	xs:string
minOccurs	0
maxOccurs	unbounded
Note	Lookup tables used for data points in the CCD. Such as DvText.dv options. These tables may be in a variety of formats such as CSV. Therefore the implementation will have to handle how this table is handled.

Element Name	terminologies
Datatype	xs:string
minOccurs	0
maxOccurs	unbounded
Note	References to terminology entries.

Element Name	generalEntries
Datatype	mlhim2:OntologyEntry
minOccurs	0
maxOccurs	unbounded
Note	A list of general Ontology Entries that do not fit into one of the predefined categories.

CCD

Concept Constraint Definition

Inherits From: xs:anyType
Abstract: False

Element Name	definition
Datatype	mlhim2:Locatable
minOccurs	1
maxOccurs	1
Note	Contains one Locatable entry as the root term for this CCD.

Element Name	ontology
Datatype	mlhim2:Ontology
minOccurs	1
maxOccurs	1
Note	Provides the semantic context for the data described by the definition.

Element Name	meta-data
Datatype	mlhim2:MetaDataSet
minOccurs	1
maxOccurs	unbounded
Note	Contains one or more named meta-data sets.

MLHIM2

Multi-Level Health Information Modelling Reference Model - Use this schema when creating bindings to other languages. It has includes for all reference model schemas.

Constraint Definitions

A Constraint Definition Designer (CDD) has been developed using the mind mapping software, XMind. It provides domain experts a copy/paste method of building up the structures to define a certain concept.

Healthcare Knowledge Component Repository

Managing CCDs

An open source content management system; Healthcare Knowledge Component Repository (HKCR) is being deployed to provide an easy path for the development and distribution of CCDs on a global basis. See the HKCR documentation for more information.

OSHIP

The Open Source Health Information Platform (OSHIP) is a generic acronym for all implementations of the the MLHIM reference model in all programming languages. The basic concept is to supply a common information model for all healthcare applications; irregardless of the implementation language.

OSHIPpy

The reference model has been generated using PyXB. It is expected that implementers using Plone, Grok, Django, Web2py, etc. will create apps based on the reference model.

OSHIPjava

The reference model has been implemented as an Eclipse module.

OSHIPrb

OSHIPcpp

OSHIPlua