

Table of Contents

Model	7
Main	7
Welcome diagram	7
<anonymous>	8
<anonymous>	9
<anonymous>	11
<anonymous>	11
\$diagram://{B076E3B6-C502-474f-B499-CE7CAB361568}	12
\$diagram://{E6899129-FE37-4129-9053-23F6628C8DA2}	12
CIMI Reference Model	12
Legend	13
Abstract	13
Datatype	13
Demographics Entry Point	13
Entry Point	13
Supporting	13
Primitive Types	14
Archetype Entry Points diagram	14
<anonymous>	14
<anonymous>	15
<anonymous>	15
Choice of ...	15
<anonymous>	15
ELEMENT	15
ITEM_GROUP	16
ACTOR	18
PARTY_RELATIONSHIP	19
ROLE	20
CIMI Reference Model	20
CIMI Core Model diagram	21
ARCHETYPED	21
ELEMENT	22
ITEM	23
ITEM_GROUP	24
LINK	25
LOCATABLE	26
PARTICIPATION	28
PARTY	28
<anonymous>	29
CIMI Data Value Types diagram	30
AMOUNT	30
CODED_TEXT	31
COUNT	33
DATA_VALUE	33
DATE	34
DATE_TIME	34
DURATION	35
EHR_URI	35
ENCAPSULATED	36

IDENTIFIER	36
INTERVAL_VALUE	37
MULTIMEDIA	38
ORDERED_VALUE	39
ORDINAL	39
PARSABLE	40
PLAIN_TEXT	41
PROPORTION	41
QUANTIFIED	42
QuantifiedValueStatusEnum	43
QUANTITY	44
TERM_MAPPING	44
TermMappingMatchEnum	46
TEXT	46
TIME	47
URI_VALUE	47
YESNO	48
<anonymous>	48
CIMI Party Model diagram	49
LOCATABLE	49
ACTOR	51
PARTY	51
PARTY_RELATIONSHIP	53
ROLE	53
<anonymous>	54
CIMI Primitive Types diagram	54
<anonymous>	55
<anonymous>	56
Any	56
Array	56
List	56
Boolean	56
Byte	56
Character	56
Integer	57
Real	57
String	57
URI	57
Core	57
ARCHETYPED	57
CORE_LOCATABLE	58
ELEMENT	58
ITEM	59
ITEM_GROUP	60
LINK	61
LOCATABLE	62
PARTICIPATION	64
Data Value Types	66
AMOUNT	66
CODED_TEXT	67
COUNT	68
DATA_VALUE	68
DATE	69
DATE_TIME	70

DURATION	70
EHR_URI	71
ENCAPSULATED	71
IDENTIFIER	72
INTERVAL_VALUE	72
MULTIMEDIA	73
ORDERED_VALUE	74
ORDINAL	75
PARSABLE	76
PLAIN_TEXT	76
PROPORTION	76
QUANTIFIED	77
QUANTITY	78
QuantifiedValueStatusEnum	79
TERM_MAPPING	80
TEXT	81
TIME	82
TermMappingMatchEnum	82
URI_VALUE	83
YESNO	83
Party	85
ACTOR	85
PARTY	86
PARTY_RELATIONSHIP	87
ROLE	88
Primitive Types	89
Any	89
Array	89
List	89
Boolean	89
Byte	89
Character	89
Integer	90
Real	90
String	90
URI	90
Legend	91
Abstract	91
Datatype	91
Demographics Entry Point	91
Entry Point	91
Supporting	91
Primitive Types	91
Mapping CIMI RM 2 HL7 RIM	93
HL7 Entity-Role-Participation-Act diagram	93
CORE_LOCATABLE	93
LINK	94
PARTICIPATION	95
ACTOR	96
PARTY	96
PARTY_RELATIONSHIP	98
ROLE	98
Act	99
ActRelationship	99

Entity	100
Participation	100
Role	100
RoleLink	101
<anonymous>	102
Mapping DCM 2 CIMI RM	103
Mapping DCM 2 CIMI RM diagram	103
ELEMENT	104
ITEM_GROUP	105
LOCATABLE	107
definition	108
<anonymous>	108
<anonymous>	108
Datatype mapping	109
<anonymous>	109
CIMI Datatypes	110
ContainerConcept	112
DCM Datatypes	113
Element	114
LeafConcept	115
Metadata	116
RootConcept	116
preamble	117
ISO631? language	118
datatype	118
id	118
name	119
Model Transformation Map	120
Model Transformation Map diagram	120
<anonymous>	120
<anonymous>	121
ADL Workbench	121
BMM	121
CEM	122
CIMI ADL	122
DCM UML	122
HL7v3 CSP	123
LinkEHR	123
MM Tool	124
Mindmap	124
Mindmap	124
OpenEHR ADL	124
UML AML	124
UML Class	125
UML Class	125
UML Object	126
UML Tool (EA, MDHT, RSA)	126
XML Schema	126

Create Instance Example	127
Tool Chain Use Case	128
Tool Chain diagram	128
CIMI Core	128
CIMI Model Development Process	129
<anonymous>	129
<anonymous>	129
<anonymous>	129
<anonymous>	129
<anonymous>	130
<anonymous>	130
Core TF	130
Modelling Team	131
Publishing Team	131
Clinical Validate Model	131
Develop Model	131
Develop RM	132
Model Metadata	132
Model Structure	133
Model Terminology	133
Publish Model	133
Publish RM	134
Technical Validate Model	134
AML Tests	135
CIMI Patterns (proposed UML representation)	135
CIMI.RM.ClinicalEntryPattern	135
CIMI.RM.ClinicalEntryPattern diagram	135
<anonymous>	136
ClinicalEntry	136
information provider	137
information subject	137
location	137
CIMI.RM.ObservationPattern (mindmap)	139
CIMI.RM.ObservationPattern (mindmap) diagram	139
<anonymous>	139
CIMI.RM.ObservationPattern (Object)	140
CIMI.RM.ObservationPattern (Object) diagram	140
INTERVAL_VALUE	140
<anonymous>	141
UML2.1.2 p.633	141
<anonymous>	142
<anonymous>	143
<anonymous>	143
INTERVAL_VALUE<T -> DATE_TIME>	143
<anonymous>	143
<anonymous>	144
<anonymous>	144
<anonymous>	144
<anonymous>	145
<anonymous>	145
<anonymous>	145

<anonymous>	145
Observation	146
datetime range	146
datetime	147
details	147
duration	147
name	148
observable	148
offset origin datetime	149
offset	149
result	149
status	150
timing	150
CIMI.RM.ObservationPattern (Class)	152
CIMI.RM.ObservationPattern (Class) diagram	152
<anonymous>	152
<anonymous>	152
<anonymous>	153
<anonymous>	153
<anonymous>	153
Clinical Entry	153
Observation	154
observable	154
details	155
name	156
status	156
result	157
timing	157
datetime range	158
Slots in UML Example	159
Slots in UML Example diagram	159
<anonymous>	160
<anonymous>	160
<anonymous>	161
<anonymous>	161
<anonymous>	161
Class1	161
Class1 B	162
Class2	162
Class2 B	162
Class3	162
Class3 B	163
Discharge Summary	163
More Concrete Discharge Summary	163
Slot Filled In	164
Xxx Diagnose	165
Diagnosis "Slot"	165
Closed class	166
Closed class diagram	166
ClosedClass	166
Parameterised Classes	167
Parameterised Classes diagram	167
Class1	167

Model

Package in package "

Model
Version Phase 1.0 Proposed
dks02 created on 12/31/2014. Last modified 12/31/2014

Main

Package in package 'Model'

The CIMI Reference Model is the underlying Reference Model upon which CIMI's clinical models (i.e. archetypes) are defined. This reference model defines a rigorous and stable set of modelling patterns, including a set of structural patterns, complex data types and demographic classes. All CIMI clinical models will be defined by constraining the CIMI reference model. Each example instance of a CIMI clinical model will be an instance of the CIMI reference model, which conforms to the constraints defined by the associated clinical models.

Main
Version 1.0.5 Phase 1.0 Proposed
created on 4/22/2012. Last modified 9/14/2012

Welcome diagram

Custom diagram in package 'Main'

Welcome
Version 1.0
ZelM created on 6/1/2012. Last modified 12/19/2014

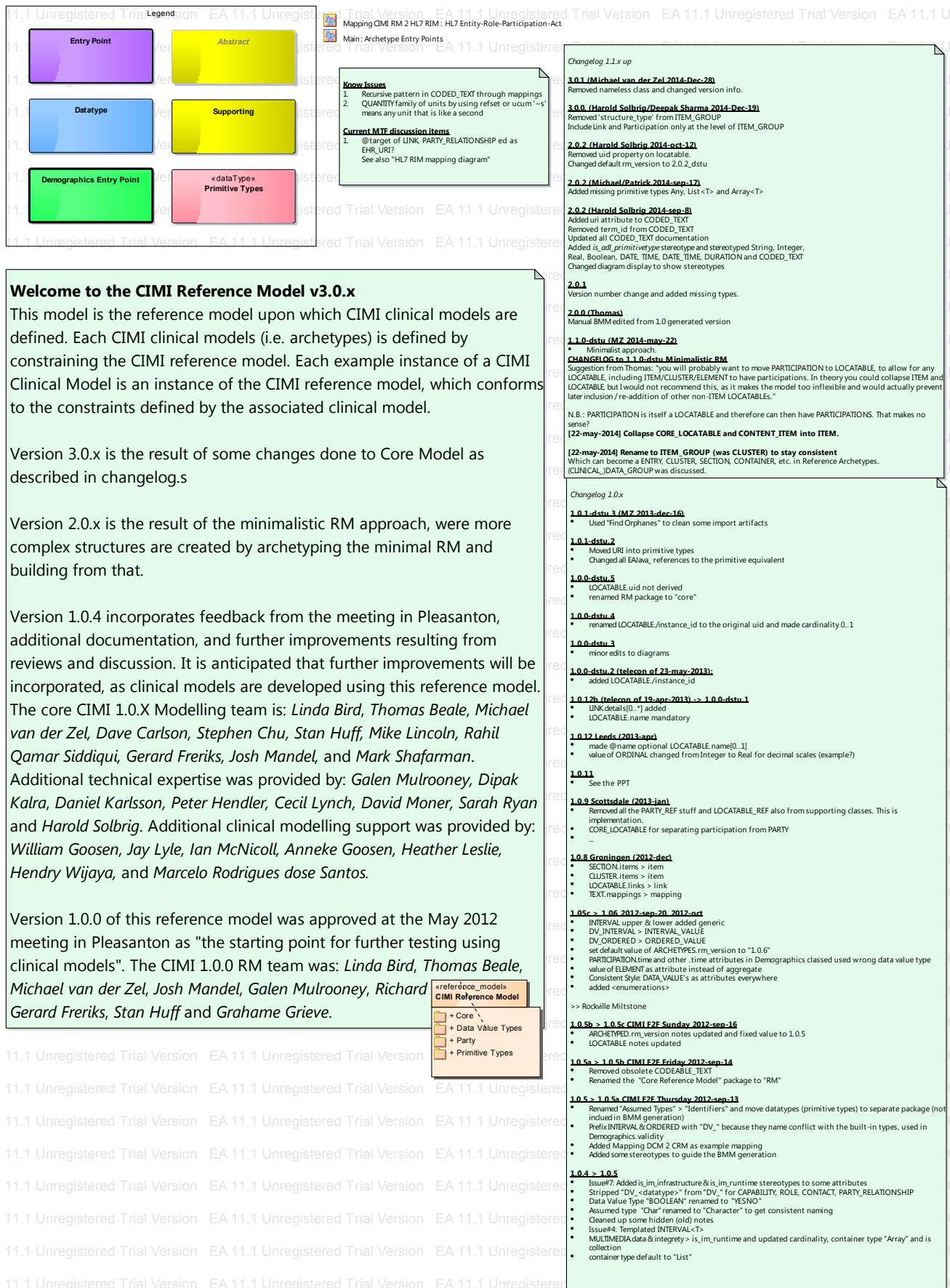


Figure 1: Welcome

<anonymous>

Note in package 'Main'

Welcome to the CIMI Reference Model v3.0.x

This model is the reference model upon which CIMI clinical models are defined. Each CIMI clinical models (i.e. archetypes) is defined by constraining the CIMI reference model. Each example instance of a CIMI Clinical Model is an instance of the CIMI reference model, which conforms to the constraints defined by the associated clinical model.

Version 3.0.x is the result of some changes done to Core Model as described in changelog.s

Version 2.0.x is the result of the minimalistic RM approach, were more complex structures are created by archetyping the minimal RM and building from that.

Version 1.0.4 incorporates feedback from the meeting in Pleasanton, additional documentation, and further improvements resulting from reviews and discussion. It is anticipated that further improvements will be incorporated, as clinical models are developed using this reference model. The core CIMI 1.0.X Modelling team is: *Linda Bird, Thomas Beale, Michael van der Zel, Dave Carlson, Stephen Chu, Stan Huff, Mike Lincoln, Rahil Qamar Siddiqui, Gerard Freriks, Josh Mandel, and Mark Shafarman*. Additional technical expertise was provided by: *Galen Mulrooney, Dipak Kalra, Daniel Karlsson, Peter Hendl, Cecil Lynch, David Moner, Sarah Ryan and Harold Solbrig*. Additional clinical modelling support was provided by: *William Goosen, Jay Lyle, Ian McNicoll, Anneke Goosen, Heather Leslie, Hendry Wijaya, and Marcelo Rodrigues dose Santos*.

Version 1.0.0 of this reference model was approved at the May 2012 meeting in Pleasanton as "the starting point for further testing using clinical models". The CIMI 1.0.0 RM team was: *Linda Bird, Thomas Beale, Michael van der Zel, Josh Mandel, Galen Mulrooney, Richard Kavanagh, Gerard Freriks, Stan Huff and Grahame Grieve*.

<anonymous>
Version 1.0 Phase 1.0 Proposed
ZelM created on 6/1/2012. Last modified 12/19/2014
Extends

<anonymous>

Note in package 'Main'

Changelog 1.0.x

1.0.1-dstu.3 (MZ 2013-dec-16)

- Used "Find Orphanes" to clean some import artifacts

1.0.1-dstu.2

- Moved URI into primitive types
- Changed all EAJava_ references to the primitive equivalent

1.0.0-dstu.5

- LOCATABLE.uid not derived
- renamed RM package to "core"

1.0.0-dstu.4

- renamed LOCATABLE./instance_id to the original uid and made cardinality 0..1

1.0.0-dstu.3

- minor edits to diagrams

1.0.0-dstu.2 (telecon of 23-may-2013):

- added LOCATABLE./instance_id

1.0.12b (telecon of 19-apr-2013) -> 1.0.0-dstu.1

- LINK.details[0..*] added
- LOCATABLE.name mandatory

1.0.12 Leeds (2013-apr)

- made @name optional LOCATABLE.name[0..1]
- value of ORDINAL changed from Integer to Real for decimal scales (example?)

1.0.11

- See the PPT

1.0.9 Scottsdale (2013-jan)

- Removed all the PARTY_REF stuff and LOCATABLE_REF also from supporting classes. This is implementation.
- CORE_LOCATABLE for separating participation from PARTY
- ...

1.0.8 Groningen (2012-dec)

- SECTION.items > item
- CLUSTER.items > item
- LOCATABLE.links > link
- TEXT.mappings > mapping

1.05c > 1.06 2012-sep-20, 2012-oct

- INTERVAL upper & lower added generic
- DV_INTERVAL > INTERVAL_VALUE
- DV_ORDERED > ORDERED_VALUE
- set default value of ARCHETYPES.rm_version to "1.0.6"
- PARTICIPATION.time and other .time attributes in Demographics classed used wrong data value type
- value of ELEMENT as attribute instead of aggregate
- Consistent Style: DATA_VALUE's as attributes everywhere
- added <enumerations>

>> Rockville Miltstone

1.0.5b > 1.0.5c CIMI F2F Sunday 2012-sep-16

- ARCHETYPED.rm_version notes updated and fixed value to 1.0.5
- LOCATABLE notes updated

1.0.5a > 1.0.5b CIMI F2F Friday 2012-sep-14

- Removed obsolete CODEABLE_TEXT
- Renamed the "Core Reference Model" package to "RM"

1.0.5 > 1.0.5a CIMI F2F Thursday 2012-sep-13

- Renamed "Assumed Types" > "Identifiers" and move datatypes (primitive types) to separate package (not included in BMM generation)
- Prefix INTERVAL & ORDERED with "DV_" because they name conflict with the built-in types, used in Demographics.validity
- Added Mapping DCM 2 CRM as example mapping
- Added some stereotypes to guide the BMM generation

1.0.4 > 1.0.5

- Issue#7: Added is_im_infrastructure & is_im_runtime stereotypes to some attributes
- Stripped "DV_<datatype>" from "DV_" for CAPABILITY, ROLE, CONTACT, PARTY_RELATIONSHIP
- Data Value Type "BOOLEAN" renamed to "YESNO"
- Assumed type "Char" renamed to "Character" to get consistent naming
- Cleaned up some hidden (old) notes
- Issue#4: Templatized INTERVAL<T>
- MULTIMEDIA.data & integrety > is_im_runtime and updated cardinality, container type "Array" and is collection
- container type default to "List"

<anonymous>

Version 1.0 Phase 1.0 Proposed
ZelM created on 12/2/2012. Last modified 6/3/2014
Extends

<anonymous>

Note in package 'Main'

Known Issues

1. Recursive pattern in CODED_TEXT through mappings
2. QUANTITY family of units by using refset or ucum '˜s' means any unit that is like a second

Current MTF discussion items

1. @target of LINK, PARTY_RELATIONSHIP ed as EHR_URI?
See also "HL7 RIM mapping diagram"

<anonymous>

Version 1.0 Phase 1.0 Proposed
zelm created on 3/22/2013. Last modified 4/20/2013
Extends

<anonymous>

Note in package 'Main'

Changelog 1.1.x up

3.0.1 (Michael van der Zel 2014-Dec-28)

Removed nameless class and changed version info.

3.0.0. (Harold Solbrig/Deepak Sharma 2014-Dec-19)

Removed 'structure_type' from ITEM_GROUP
Include Link and Participation only at the level of ITEM_GROUP

2.0.2 (Harold Solbrig 2014-oct-12)

Removed uid property on locatable.
Changed default rm_version to 2.0.2_dstu

2.0.2 (Michael/Patrick 2014-sep-17)

Added missing primitive types Any, List<T> and Array<T>

2.0.2 (Harold Solbrig 2014-sep-8)

Added uri attribute to CODED_TEXT
Removed term_id from CODED_TEXT
Updated all CODED_TEXT documentation
Added *is_adl_primitive* stereotype and stereotyped String, Integer,
Real, Boolean, DATE, TIME, DATE_TIME, DURATION and CODED_TEXT
Changed diagram display to show stereotypes

2.0.1

Version number change and added missing types.

2.0.0 (Thomas)

Manual BMM edited from 1.0 generated version

1.1.0-dstu (MZ 2014-may-22)

- Minimalist approach.

CHANGELOG to 1.1.0-dstu Minimalistic RM

Suggestion from Thomas: "you will probably want to move PARTICIPATION to LOCATABLE, to allow for any LOCATABLE, including ITEM/CLUSTER/ELEMENT to have participations. In theory you could collapse ITEM and LOCATABLE, but I would not recommend this, as it makes the model too inflexible and would actually prevent later inclusion / re-addition of other non-ITEM LOCATABLEs."

N.B.: PARTICIPATION is itself a LOCATABLE and therefore can then have PARTICIPATIONS. That makes no sense?

[22-may-2014] Collapse CORE_LOCATABLE and CONTENT_ITEM into ITEM.

[22-may-2014] Rename to ITEM_GROUP (was CLUSTER) to stay consistent

Which can become a ENTRY, CLUSTER, SECTION, CONTAINER, etc. in Reference Archetypes.
(CLINICAL_)DATA_GROUP was discussed.

<anonymous>
Version 1.0 Phase 1.0 Proposed
Michael van der Zel created on 6/3/2014. Last modified 12/31/2014
Extends

\$diagram://{B076E3B6-C502-474f-B499-CE7CAB361568}

Text in package 'Main'

Main : Archetype Entry Points

\$diagram://{B076E3B6-C502-474f-B499-CE7CAB361568}
Version 1.0 Phase 1.0 Proposed
ZelM created on 3/24/2013. Last modified 3/24/2013
Extends

\$diagram://{E6899129-FE37-4129-9053-23F6628C8DA2}

Text in package 'Main'

Mapping CIMI RM 2 HL7 RIM : HL7 Entity-Role-Participation-Act

\$diagram://{E6899129-FE37-4129-9053-23F6628C8DA2}
Version 1.0 Phase 1.0 Proposed
ZelM created on 3/24/2013. Last modified 3/24/2013
Extends

CIMI Reference Model

Package «reference_model» in package 'Main'

Welcome to the CIMI Reference Model v2.0.x

This model is the reference model upon which CIMI clinical models are defined. Each CIMI clinical models (i.e. archetypes) is defined by constraining the CIMI reference model. Each example instance of a CIMI Clinical Model is an instance of the CIMI reference model, which conforms to the constraints defined by the associated clinical model.

Version 2.0.x is the result of the minimalistic RM approach, were more complex structures are created by archetyping the minimal RM and building from that.

Version 1.0.4 incorporates feedback from the meeting in Pleasanton, additional documentation, and further improvements resulting from reviews and discussion. It is anticipated that further improvements will be incorporated, as clinical models are developed using this reference model. The core CIMI 1.0.X Modelling team is: *Linda Bird, Thomas Beale, Michael van der Zel, Dave Carlson, Stephen Chu, Stan Huff, Mike Lincoln, Rahil Qamar Siddiqui, Gerard Freriks, Josh Mandel, and Mark Shafarman*. Additional technical expertise was provided by: *Galen Mulrooney, Dipak Kalra, Daniel Karlsson, Peter Handler, Cecil Lynch, David Moner, Sarah Ryan and Harold Solbrig*. Additional clinical modelling support was provided by: *William Goosen, Jay Lyle, Ian McNicoll, Anneke Goosen, Heather Leslie, Hendry Wijaya, and Marcelo Rodrigues dose Santos*.

Version 1.0.0 of this reference model was approved at the May 2012 meeting in Pleasanton as "the starting point for

further testing using clinical models". The CIMI 1.0.0 RM team was: *Linda Bird, Thomas Beale, Michael van der Zel, Josh Mandel, Galen Mulrooney, Richard Kavanagh, Gerard Freriks, Stan Huff and Grahame Grieve.*

CIMI Reference Model
Version 3.0.1 Phase 1.0 dstu
CIMI created on 9/14/2012. Last modified 12/28/2014
Alias RM

Legend

Boundary in package 'Legend'

Legend
Version 1.0 Phase 1.0 Proposed
ZelM created on 6/1/2012. Last modified 6/1/2012
Extends

Abstract

Class in package 'Legend'

Abstract
Version 1.0 Phase 1.0 Proposed
Linda created on 6/1/2012. Last modified 6/1/2012

Datatype

Class in package 'Legend'

Datatype
Version 1.0 Phase 1.0 Proposed
Linda created on 6/1/2012. Last modified 6/1/2012

Demographics Entry Point

Class in package 'Legend'

Demographics Entry Point
Version 1.0 Phase 1.0 Proposed
Linda created on 6/1/2012. Last modified 2/7/2013

Entry Point

Class in package 'Legend'

Entry Point
Version 1.0 Phase 1.0 Proposed
Linda created on 6/1/2012. Last modified 6/1/2012

Supporting

Class in package 'Legend'

Supporting
Version 1.0 Phase 1.0 Proposed
Linda created on 6/1/2012. Last modified 6/1/2012

Primitive Types

DataType in package 'Legend'

= Assumed Types

Primitive Types
Version 1.0 Phase 1.0 Proposed
Linda created on 6/1/2012. Last modified 12/2/2012

Archetype Entry Points diagram

Class diagram in package 'Main'

Archetype Entry Points
Version 1.0
ZelM created on 2/7/2013. Last modified 6/3/2014

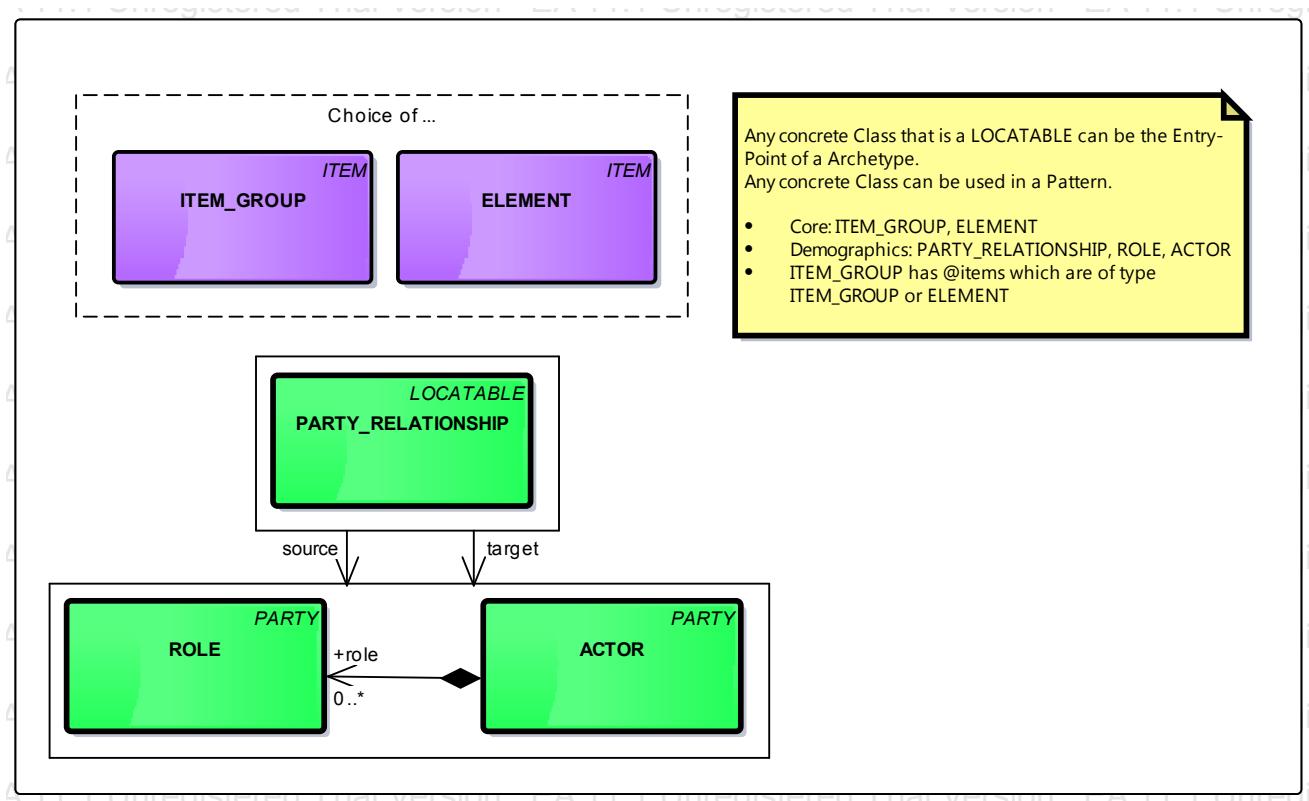


Figure 2: Archetype Entry Points

<anonymous>

Boundary in package 'Main'

<anonymous>
Version 1.0 Phase 1.0 Proposed
ZelM created on 2/7/2013. Last modified 2/7/2013
Extends

<anonymous>

Boundary in package 'Main'

<anonymous>
Version 1.0 Phase 1.0 Proposed
ZelM created on 2/7/2013. Last modified 2/7/2013
Extends

<anonymous>

Boundary in package 'Main'

<anonymous>
Version 1.0 Phase 1.0 Proposed
ZelM created on 2/7/2013. Last modified 2/7/2013
Extends

Choice of ...

Boundary in package 'Main'

Choice of ...
Version 1.0 Phase 1.0 Proposed
ZelM created on 2/7/2013. Last modified 2/7/2013
Extends

<anonymous>

Note in package 'Main'

Any concrete Class that is a LOCATABLE can be the Entry-Point of a Archetype.
Any concrete Class can be used in a Pattern.

- Core: ITEM_GROUP, ELEMENT
- Demographics: PARTY_RELATIONSHIP, ROLE, ACTOR
- ITEM_GROUP has @items which are of type ITEM_GROUP or ELEMENT

<anonymous>
Version 1.0 Phase 1.0 Proposed
ZelM created on 2/7/2013. Last modified 6/3/2014
Extends

ELEMENT

Class in package 'Core'

A type of data ITEM, which does not itself contain ITEMS.

ELEMENT
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 5/22/2014
Extends ITEM

OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Generalization from ELEMENT to ITEM

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

➡ Generalization from name to ELEMENT

[Direction is 'Source -> Destination'.]

➡ Generalization from status to ELEMENT

[Direction is 'Source -> Destination'.]

➡ Generalization from datetime range to ELEMENT

[Direction is 'Source -> Destination'.]

CONNECTORS

↗ Trace «trace» Source -> Destination

From: datatype : Property, Public

To: ELEMENT : Class, Public value

↗ Dependency «instantiate» Source -> Destination

From: LeafConcept : Class, Public

To: ELEMENT : Class, Public

ATTRIBUTES

◆ null_flavor : CODED_TEXT Public

Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)

[Is static False. Containment is Not Specified.]

◆ value : DATA_VALUE Public

Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)

[Is static False. Containment is Not Specified.]

ITEM_GROUP

Class in package 'Core'

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

ITEM_GROUP

Version 1.0 Phase 1.0 Proposed
 Linda created on 4/22/2012. Last modified 12/18/2014
 Extends ITEM

OUTGOING STRUCTURAL RELATIONSHIPS

◀ Generalization from ITEM_GROUP to ITEM

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

⇒ Generalization from observable to ITEM_GROUP

[Direction is 'Source -> Destination'.]

⇒ Generalization from result to ITEM_GROUP

[Direction is 'Source -> Destination'.]

⇒ Generalization from timing to ITEM_GROUP

[Direction is 'Source -> Destination'.]

⇒ Aggregation from PARTICIPATION to ITEM_GROUP

[Direction is 'Destination -> Source'.]

CONNECTORS

↗ Dependency «instantiate» Source -> Destination

From: RootConcept : Class, Public
 To: ITEM_GROUP : Class, Public

↗ Dependency «instantiate» Source -> Destination

From: ContainerConcept : Class, Public
 To: ITEM_GROUP : Class, Public

ASSOCIATIONS

↙ Association (direction: Source -> Destination)

Source: Public source (Class) ITEM_GROUP

Target: Public link (Class) LINK

Cardinality: [0..*]

↙ Association (direction: Source -> Destination)

Source: Private (Class) ITEM_GROUP

Target: Public item (Class) ITEM

Cardinality: [1..*]

↙ Association (direction: Source -> Destination)

Target can be archetype_id or constraint (business rules/query language) on instance.

Target is the constraint on "type of instance" referenced at runtime.

target LOCATABLE "type of instance" is constrained in the AOM. Just like terminology binding.

ASSOCIATIONS

Target is an association to a LOCATABLE.

Source: Public (Class) LINK

Target: Public target (Class) ITEM_GROUP

Cardinality: [1..1]

ACTOR

Class in package 'Party'

An ancestor of all real-world types, including people and organisations. An actor is any real-world entity capable of taking on a role.

GROUP

A group is a real world group of parties which is created by another party (usually an organisation) for some specific purpose. A typical clinical example is that of the specialist care team, e.g. cardiology team. The members of the group usually work together.

AGENT (DEVICE)

A generic concept of any kind of agent, including devices, software systems, but not humans or organisations.

PERSON

The generic description of a person. PERSON provides a dedicated type to which Person archetypes can be targeted.

ORGANISATION

A generic description of an organisation. An organisation is a legally constituted body whose existence (in general) outlives the existence of parties considered to be part of it.

ACTOR

Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013
Extends PARTY

OUTGOING STRUCTURAL RELATIONSHIPS

◀ Generalization from ACTOR to PARTY

[Direction is 'Source -> Destination'.]

CONNECTORS

↗ Dependency Destination -> Source

From: ACTOR : Class, Public

To: Entity : Boundary, Public

ATTRIBUTES

◆ type : CODED_TEXT Public

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) ACTOR

Target: Public role (Class) ROLE

Cardinality: [0..*]

PARTY_RELATIONSHIP

Class in package 'Party'

A generic description of a relationship between the source and target parties.

PARTY_RELATIONSHIP
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013
Extends LOCATABLE

OUTGOING STRUCTURAL RELATIONSHIPS

Generalization from PARTY_RELATIONSHIP to «archetype_parent» LOCATABLE

[Direction is 'Source -> Destination'.]

CONNECTORS

Dependency Destination -> Source

From: PARTY_RELATIONSHIP : Class, Public
To: RoleLink : Boundary, Public

ATTRIBUTES

type : CODED_TEXT Public

The detailed description of the relationship

[Is static False. Containment is Not Specified.]

details : ITEM Public

Multiplicity: ([0..*], Allow duplicates: 0, Is ordered: False)

The detailed description of the relationship.

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) PARTY_RELATIONSHIP

Target: Public target (Class) PARTY

Cardinality: [1..1]

Association (direction: Source -> Destination)

ASSOCIATIONS	
Source: Public source (Class) PARTY	Target: Public relationship (Class) PARTY_RELATIONSHIP Cardinality: [0..*]

ROLE

Class in package 'Party'

A generic description of a role performed by an actor. The role corresponds to a competency of the party. Roles are used to define the responsibilities undertaken by a party for a purpose. Roles should have credentials qualifying the performer to perform the role.

ROLE
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013
Extends PARTY

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Generalization from ROLE to PARTY	[Direction is 'Source -> Destination'.]

CONNECTORS	
↗ Dependency Destination -> Source From: ROLE : Class, Public To: Role : Boundary, Public	

ATTRIBUTES	
◆ type : CODED_TEXT Public	[Is static False. Containment is Not Specified.]

ASSOCIATIONS	
✍ Association (direction: Source -> Destination)	
Source: Public (Class) ACTOR	Target: Public role (Class) ROLE Cardinality: [0..*]

CIMI Reference Model

Package «reference_model» in package 'Main'

Welcome to the CIMI Reference Model v2.0.x

This model is the reference model upon which CIMI clinical models are defined. Each CIMI clinical models (i.e. archetypes) is defined by constraining the CIMI reference model. Each example instance of a CIMI Clinical Model is an

instance of the CIMI reference model, which conforms to the constraints defined by the associated clinical model.

Version 2.0.x is the result of the minimalistic RM approach, where more complex structures are created by archetyping the minimal RM and building from that.

Version 1.0.4 incorporates feedback from the meeting in Pleasanton, additional documentation, and further improvements resulting from reviews and discussion. It is anticipated that further improvements will be incorporated, as clinical models are developed using this reference model. The core CIMI 1.0.X Modelling team is: *Linda Bird, Thomas Beale, Michael van der Zel, Dave Carlson, Stephen Chu, Stan Huff, Mike Lincoln, Rahil Qamar Siddiqui, Gerard Freriks, Josh Mandel, and Mark Shafarman*. Additional technical expertise was provided by: *Galen Mulrooney, Dipak Kalra, Daniel Karlsson, Peter Helder, Cecil Lynch, David Moner, Sarah Ryan and Harold Solbrig*. Additional clinical modelling support was provided by: *William Goosen, Jay Lyle, Ian McNicoll, Anneke Goosen, Heather Leslie, Hendry Wijaya, and Marcelo Rodrigues dose Santos*.

Version 1.0.0 of this reference model was approved at the May 2012 meeting in Pleasanton as "the starting point for further testing using clinical models". The CIMI 1.0.0 RM team was: *Linda Bird, Thomas Beale, Michael van der Zel, Josh Mandel, Galen Mulrooney, Richard Kavanagh, Gerard Freriks, Stan Huff and Grahame Grieve*.

CIMI Reference Model
Version 3.0.1 Phase 1.0 dstu
CIMI created on 9/14/2012. Last modified 12/28/2014
Alias RM

CIMI Core Model diagram

Class diagram in package 'CIMI Reference Model'

The CIMI Core Reference Model diagram shows the main classes in the CIMI reference model upon which clinical models will be defined. These include the concrete classes Item Group and Element.

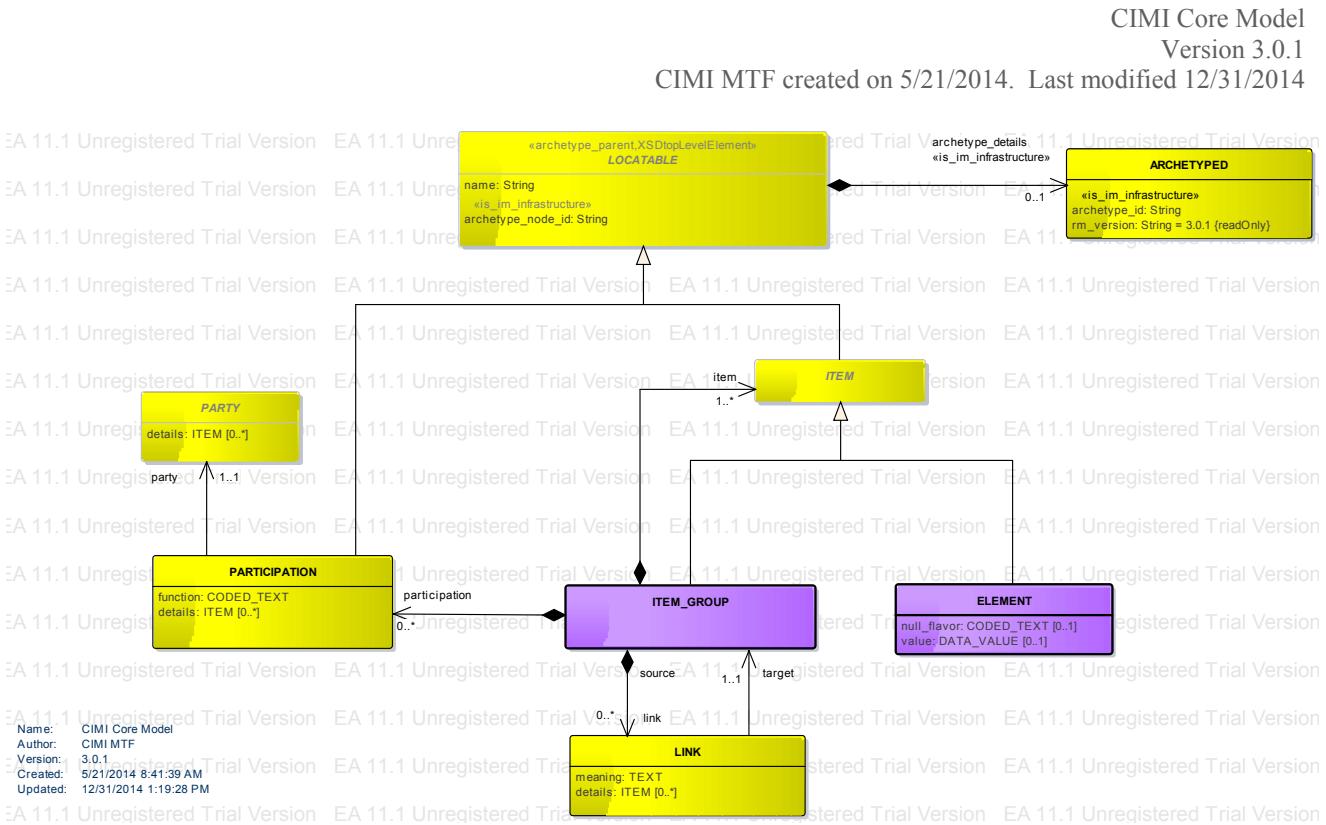


Figure 3: CIMI Core Model

ARCHETYPED

Class in package 'Core'

Archetypes act as the configuration basis for the particular structures of instances defined by the reference model. To enable archetypes to be used to create valid data, key classes in the reference model act as root points for archotyping. These classes have the archetype_details attribute set. An instance of the class ARCHETYPED contains the relevant archetype identification information, allowing archetypes to be matched up with data instances.

ARCHETYPED
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 5/22/2014

ATTRIBUTES

 archetype_id : String Public

Globally unique archetype identifier.

The identifier for archetypes. Ideally these would identify globally unique archetypes. Lexical form:
rm_originator '-' rm_name '-' rm_entity '.' concept_name { '-' specialisation }* .v' number

[Stereotype is «is_im_infrastructure». Is static False. Containment is Not Specified.]

 rm_version : String Public Const = 3.0.1

Version of the CIMI reference model used to create this object.
Expressed in terms of the release version string, e.g. 1.0 , 1.2.4 .

[Stereotype is «is_im_infrastructure». Is static False. Containment is Not Specified.]

ASSOCIATIONS

 Association (direction: Source -> Destination)

Source: Private (Class) LOCATABLE «archetype_parent»

Target: Public «is_im_infrastructure»
archetype_details (Class) ARCHETYPED
Cardinality: [0..1]

ELEMENT*Class in package 'Core'*

A type of data ITEM, which does not itself contain ITEMS.

ELEMENT
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 5/22/2014
Extends ITEM

OUTGOING STRUCTURAL RELATIONSHIPS

 Generalization from ELEMENT to ITEM

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Generalization from name to ELEMENT	[Direction is 'Source -> Destination'.]
⇒ Generalization from status to ELEMENT	[Direction is 'Source -> Destination'.]
⇒ Generalization from datetime range to ELEMENT	[Direction is 'Source -> Destination'.]

CONNECTORS	
↗ Trace «trace» Source -> Destination From: datatype : Property, Public To: ELEMENT : Class, Public value	
↗ Dependency «instantiate» Source -> Destination From: LeafConcept : Class, Public To: ELEMENT : Class, Public	

ATTRIBUTES	
◆ null_flavor : CODED_TEXT Public Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)	[Is static False. Containment is Not Specified.]
◆ value : DATA_VALUE Public Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)	[Is static False. Containment is Not Specified.]

ITEM

Class in package 'Core'

The abstract parent of CLUSTER and ELEMENT representation classes.

ITEM
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 5/22/2014
Extends LOCATABLE

OUTGOING STRUCTURAL RELATIONSHIPS	
◀ Generalization from ITEM to «archetype_parent» LOCATABLE	[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS	
-----------------------------------	--

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Generalization from ELEMENT to ITEM	[Direction is 'Source -> Destination'.]
⇒ Generalization from ITEM_GROUP to ITEM	[Direction is 'Source -> Destination'.]
⇒ Generalization from details to ITEM	[Direction is 'Source -> Destination'.]

ASSOCIATIONS	
Association (direction: Source -> Destination) Source: Private (Class) ITEM_GROUP Target: Public item (Class) ITEM Cardinality: [1..*]	

ITEM_GROUP

Class in package 'Core'

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

ITEM_GROUP
 Version 1.0 Phase 1.0 Proposed
 Linda created on 4/22/2012. Last modified 12/18/2014
 Extends ITEM

OUTGOING STRUCTURAL RELATIONSHIPS	
← Generalization from ITEM_GROUP to ITEM	[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Generalization from observable to ITEM_GROUP	[Direction is 'Source -> Destination'.]
⇒ Generalization from result to ITEM_GROUP	[Direction is 'Source -> Destination'.]
⇒ Generalization from timing to ITEM_GROUP	[Direction is 'Source -> Destination'.]
⇒ Aggregation from PARTICIPATION to ITEM_GROUP	[Direction is 'Destination -> Source'.]

CONNECTORS

 **Dependency** «instantiate» Source -> Destination

From: RootConcept : Class, Public
To: ITEM_GROUP : Class, Public

 **Dependency** «instantiate» Source -> Destination

From: ContainerConcept : Class, Public
To: ITEM_GROUP : Class, Public

ASSOCIATIONS

 Association (direction: Source -> Destination)

Source: Public source (Class) ITEM_GROUP

Target: Public link (Class) LINK

Cardinality: [0..*]

 Association (direction: Source -> Destination)

Source: Private (Class) ITEM_GROUP

Target: Public item (Class) ITEM

Cardinality: [1..*]

 Association (direction: Source -> Destination)

Target can be archetype_id or constraint (business rules/query language) on instance.

Target is the constraint on "type of instance" referenced at runtime.

target LOCATABLE "type of instance" is constrained in the AOM. Just like terminology binding.

Target is an association to a LOCATABLE.

Source: Public (Class) LINK

Target: Public target (Class) ITEM_GROUP

Cardinality: [1..1]

LINK

Class in package 'Core'

The LINK type defines a logical relationship between two items, such as two ENTRYs or an ENTRY and a COMPOSITION. Links can be used across compositions, and across EHRs. Links can potentially be used between interior (i.e. non archetype root) nodes, although this probably should be prevented in archetypes. Multiple LINKs can be attached to the root object of any archetyped structure to give the effect of a 1->N link. 1:1 and 1:N relationships between archetyped content elements (e.g. ENTRYs) can be expressed by using one, or more than one, respectively, LINKs. Chains of links can be used to see problem threads or other logical groupings of items. Links should be used between archetyped structures only, i.e. between objects representing complete domain concepts, because relationships between sub-elements of whole concepts are not necessarily meaningful. Sensible /*links only exist between whole ITEM_GROUPS and/or PARTICIPATIONS.

LINK

Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 12/18/2014

CONNECTORS

CONNECTORS **Dependency** Destination -> Source

From: LINK : Class, Public

To: ActRelationship : Boundary, Public

ATTRIBUTES meaning : TEXT Public

Used to describe the relationship between the source and the target of the link (usually in clinical terms) - such as the relationship between test results and an order, follow-up to a consultation and so on. The meaning of each link falls under one of the following categories: generic, documenting and reporting, organisational, clinical , circumstantial , and view management .

[Is static False. Containment is .]

 details : ITEM Public

Multiplicity: ([0..*], Allow duplicates: 0, Is ordered: False)

[Is static False. Containment is Not Specified.]

ASSOCIATIONS Association (direction: Source -> Destination)

Target can be archetype_id or constraint (business rules/query language) on instance.

Target is the constraint on "type of instance" referenced at runtime.

target LOCATABLE "type of instance" is constrained in the AOM. Just like terminology binding.

Target is an association to a LOCATABLE.

Source: Public (Class) LINK

Target: Public target (Class) ITEM_GROUP

Cardinality: [1..1]

 Association (direction: Source -> Destination)

Source: Public source (Class) ITEM_GROUP

Target: Public link (Class) LINK

Cardinality: [0..*]

LOCATABLE*Class «archetype_parent» in package 'Core'*

The root class of all information model classes that can be archetyped. Most classes in the CIMI reference model inherit from the LOCATABLE class, which defines the idea of locatability in an archetyped structure. LOCATABLE defines a runtime name and an archetype_node_id.

LOCATABLE
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 5/22/2014

INCOMING STRUCTURAL RELATIONSHIPS

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Generalization from PARTICIPATION to «archetype_parent» LOCATABLE	[Direction is 'Source -> Destination'.]
⇒ Generalization from PARTY to «archetype_parent» LOCATABLE	[Direction is 'Source -> Destination'.]
⇒ Generalization from ITEM to «archetype_parent» LOCATABLE	[Direction is 'Source -> Destination'.]
⇒ Generalization from PARTY_RELATIONSHIP to «archetype_parent» LOCATABLE	[Direction is 'Source -> Destination'.]

CONNECTORS	
↗ Trace «trace» Source -> Destination From: id : Property, Public To: LOCATABLE : Class, Public	
↗ Trace «trace» Source -> Destination From: name : Property, Public To: LOCATABLE : Class, Public	

ATTRIBUTES	
◆ archetype_node_id : String Public	
The design-time archetype id of this node taken from its generating archetype. This archetype node id is used to build archetype paths. It is always in the form of an at code, e.g. at0005 . This value enables a "standardised" name for this node to be generated, by referring to the generating archetype local ontology. At an archetype root point, the value of this attribute is always the stringified form of the archetype_id found in the archetype_details object.	[Stereotype is «is_im_infrastructure». Is static False. Containment is Not Specified.]
◆ name : String Public	
The runtime name of this fragment, used to build runtime paths. This is the term provided via a clinical application or batch process to name this EHR construct. Its retention in the EHR faithfully preserves the original label by which this entry was known to end users.	[Is static False. Containment is Not Specified.]

ASSOCIATIONS	
↙ Association (direction: Source -> Destination) Source: Private (Class) LOCATABLE «archetype_parent»	Target: Public «is_im_infrastructure» archetype_details (Class) ARCHETYPED Cardinality: [0..1]

PARTICIPATION

Class in package 'Core'

Used to represent any participation of a Party (e.g. any Actor or Role) in some activity, 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.

PARTICIPATION
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 5/22/2014
Extends LOCATABLE

OUTGOING STRUCTURAL RELATIONSHIPS

◀ Generalization from PARTICIPATION to «archetype_parent» LOCATABLE

[Direction is 'Source -> Destination'.]

◀ Aggregation from PARTICIPATION to ITEM_GROUP

[Direction is 'Destination -> Source'.]

CONNECTORS

↗ Dependency Destination -> Source

From: PARTICIPATION : Class, Public

To: Participation : Boundary, Public

ATTRIBUTES

◆ function : CODED_TEXT Public

The function of the Party in this participation. Please note that a given party might participate in more than one way in a particular activity.

[Is static False. Containment is Not Specified.]

◆ details : ITEM Public

Multiplicity: ([0..*], Allow duplicates: 0, Is ordered: False)

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

↙ Association (direction: Source -> Destination)

Source: Public (Class) PARTICIPATION

Target: Public party (Class) PARTY

Cardinality: [1..1]

PARTY

Class in package 'Party'

An ancestor of all party types, including real world entities and their roles. A party is any entity which can participate in

an activity. The name attribute inherited from LOCATABLE is used to indicate the actual type of the party (note that the actual names, i.e. identities of parties are indicated in the identities attribute, not the name attribute).

PARTY
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 5/22/2014
Extends LOCATABLE

OUTGOING STRUCTURAL RELATIONSHIPS

◀ Generalization from PARTY to «archetype_parent» LOCATABLE

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

⇒ Generalization from ROLE to PARTY

[Direction is 'Source -> Destination'.]

⇒ Generalization from ACTOR to PARTY

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ details : ITEM Public

Multiplicity: ([0..*], Allow duplicates: 0, Is ordered: False)

All other details for this party. These details are usually archetyped.

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

✓ Association (direction: Source -> Destination)

Source: Public source (Class) PARTY

Target: Public relationship (Class)

PARTY_RELATIONSHIP

Cardinality: [0..*]

✓ Association (direction: Source -> Destination)

Source: Public (Class) PARTICIPATION

Target: Public party (Class) PARTY

Cardinality: [1..1]

✓ Association (direction: Source -> Destination)

Source: Public (Class) PARTY_RELATIONSHIP

Target: Public target (Class) PARTY

Cardinality: [1..1]

<anonymous>

Text in package 'CIMI Reference Model'

<anonymous>
Version 1.0 Phase 1.0 Proposed
zelm created on 5/21/2014. Last modified 12/31/2014
Extends

CIMI Data Value Types diagram

Class diagram in package 'CIMI Reference Model'

The CIMI Data Value Types diagram shows the set of data types, which underly all other models. These data types provides both general and clinically specific types required for all kinds of health information.

CIMI Data Value Types
Version 3.0.1

CIMI MTF created on 3/22/2013. Last modified 12/31/2014

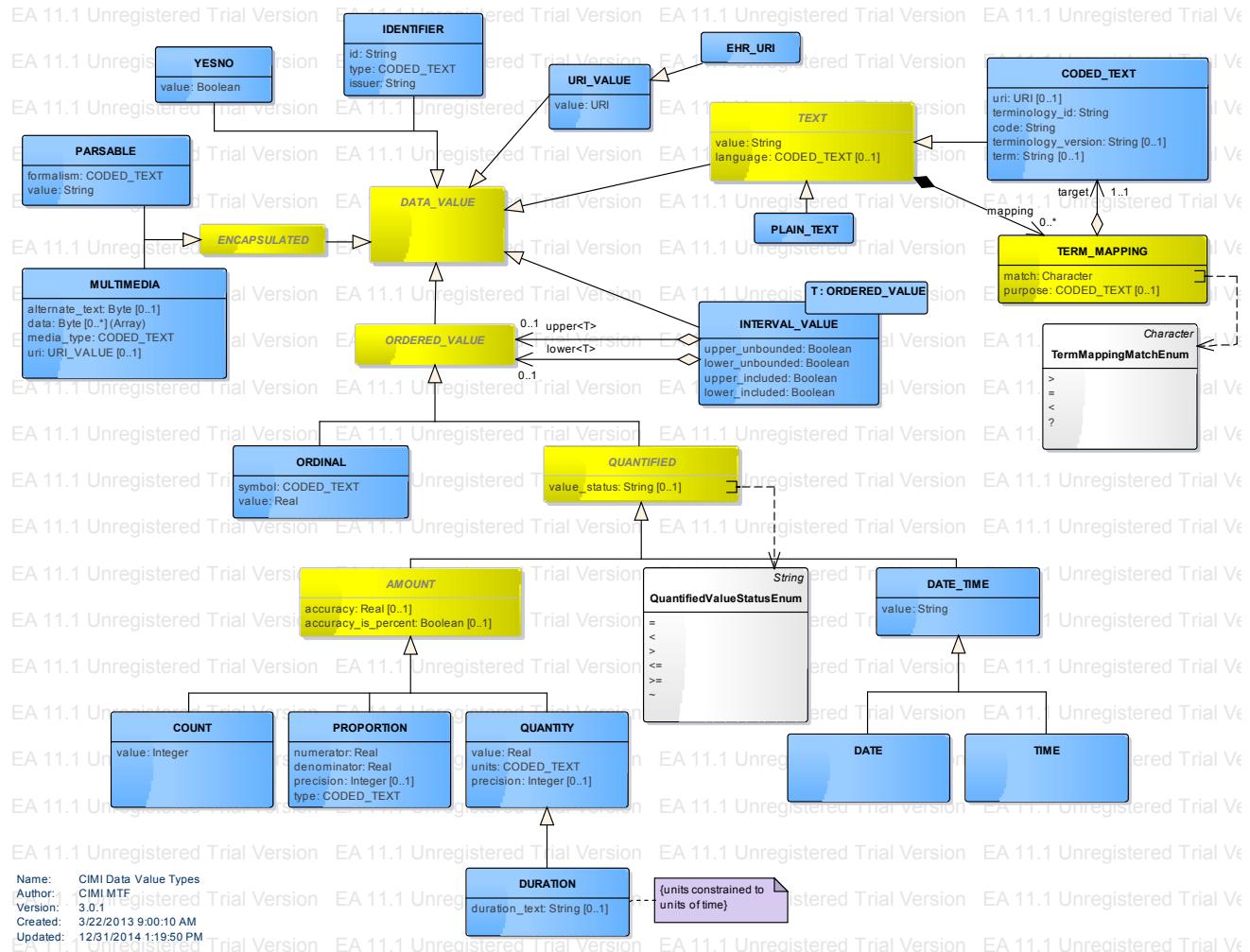


Figure 4: CIMI Data Value Types

AMOUNT

Class in package 'Data Value Types'

Abstract class defining the concept of relative quantified amounts'. For relative quantities, the '+' and '-' operators are

defined (unlike descendants of ABSOLUTE_QUANTITY, such as the date/time types).

AMOUNT
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013
Extends QUANTIFIED

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from AMOUNT to QUANTIFIED

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

⇒ Generalization from COUNT to AMOUNT

[Direction is 'Source -> Destination'.]

⇒ Generalization from QUANTITY to AMOUNT

[Direction is 'Source -> Destination'.]

⇒ Generalization from PROPORTION to AMOUNT

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ accuracy : Real Public

Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)

Accuracy of the measurement, expressed either

as a half-range percent value (i.e. accuracy_is_percent = True) or a half-range quantity. A value of 0 means that accuracy was not recorded.

[Is static False. Containment is Not Specified.]

◆ accuracy_is_percent : Boolean Public

Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)

If True, this indicates that when this object was created, accuracy was recorded as a percent value; if False, as an absolute quantity value.

[Is static False. Containment is Not Specified.]

CODED_TEXT

Class «is_adl_primitivetype» in package 'Data Value Types'

A reference to a class, category or individual that is described in an external terminology. Every CODED_TEXT instance must either have a *uri*, a *code* or both. When a *uri* is present in a CODED_TEXT instance, it will be treated as the instance identity — any two CODED_TEXT instances reference the same concept if they have the same *uri*, and the remaining fields will be ignored. If a *uri* is not included in a CODED_TEXT instance, the instance identity is the *terminology_id* / *code* pair. Any two CODED_TEXT instances reference the same concept if they (a) both have no *uri* and (b) have the same *terminology_id* and *code*. The *terminology_version* and *term* attributes are strictly informative

and play no role in determining the concept referent.

CODED_TEXT
 Version 1.0 Phase 1.0 Proposed
 Linda created on 4/22/2012. Last modified 9/8/2014
 Extends TEXT

OUTGOING STRUCTURAL RELATIONSHIPS

◀ Generalization from «is_adl_primitivetype» CODED_TEXT to TEXT

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ uri : URI Public

Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)

A URI that uniquely identifies the referenced concept. Examples: <http://snomed.info/id/74400008>, <http://loinc.org/id/2951-2>, <http://id.loc.gov/vocabulary/iso639-1/de>

[Is static False. Containment is Not Specified.]

◆ terminology_id : String Public

"A locally unique identifier for the namespace from which code was derived. Examples: SNOMED_CT, LOINC, ISO639-1
 [Is static False. Containment is Not Specified.]

◆ code : String Public

A code that uniquely identifies the referenced concept within the context of terminology_id. Examples: 74400008, 2951-2, de
 [Is static False. Containment is Not Specified.]

◆ terminology_version : String Public

Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)

"The URI of a terminology or terminology version from which the meaning of the code was determined for the purposes of this record. Examples: <http://snomed.info/sct/900000000000207008>, <http://snomed.info/sct/900000000000207008/version/20130731>, <http://loinc.org/>, <http://loinc.org/246>, <http://id.loc.gov/vocabulary/iso639-1>

[Is static False. Containment is Not Specified.]

◆ term : String Public

Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)

A human readable string that conveys the intended meaning of the code. Examples: 'Appendicitis', 'Appendicitis (Finding)', 'inflamación aguda del apéndice', 'Serum Sodium', 'Plasma Serum Sodium', 'German', 'Deutsch'

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

↙ Association (direction: Source -> Destination)

Source: Private (Class) TERM_MAPPING

Target: Public target (Class) CODED_TEXT

«is_adl_primitivetype»

Cardinality: [1..1]

COUNT

Class in package 'Data Value Types'

A countable quantity. Used for countable types, such as number of pregnancies, number of steps (taken by a physiotherapy patient), and number of cigarettes smoked in a day. Not to be used for amounts of physical entities (which have units).

COUNT

Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013
Extends AMOUNT

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from COUNT to AMOUNT

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ value : Integer Public

[Is static False. Containment is Not Specified.]

DATA_VALUE

Class in package 'Data Value Types'

Serves as a common ancestor of all data value types in the CIMI model.

DATA_VALUE

Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 9/13/2012

INCOMING STRUCTURAL RELATIONSHIPS

⇒ Generalization from INTERVAL_VALUE to DATA_VALUE

[Direction is 'Source -> Destination'.]

⇒ Generalization from ENCAPSULATED to DATA_VALUE

[Direction is 'Source -> Destination'.]

⇒ Generalization from YESNO to DATA_VALUE

[Direction is 'Source -> Destination'.]

⇒ Generalization from URI_VALUE to DATA_VALUE

[Direction is 'Source -> Destination'.]

⇒ Generalization from TEXT to DATA_VALUE

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Generalization from IDENTIFIER to DATA_VALUE	[Direction is 'Source -> Destination'.]
⇒ Generalization from ORDERED_VALUE to DATA_VALUE	[Direction is 'Source -> Destination'.]

DATE

Class «is_adl_primitivetype» in package 'Data Value Types'

Represents an absolute point in time, as measured on the Gregorian calendar, and specified only to the day. Semantics defined by ISO 8601. Used for recording dates in real world time. The partial form is used for approximate birth dates, dates of death, etc.

value constrained to: The ISO 8601 date string.

DATE
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 9/8/2014
Extends DATE_TIME

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Generalization from «is_adl_primitivetype» DATE to «is_adl_primitivetype» DATE_TIME	[Direction is 'Source -> Destination'.]

DATE_TIME

Class «is_adl_primitivetype» in package 'Data Value Types'

Represents an absolute point in time, specified to the second. Semantics defined by ISO 8601. Used for recording a precise point in real world time, and for approximate time stamps.

value constrained to: The ISO8601 date/time string.

DATE_TIME
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 9/8/2014
Extends QUANTIFIED

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Generalization from «is_adl_primitivetype» DATE_TIME to QUANTIFIED	[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS	
-----------------------------------	--

INCOMING STRUCTURAL RELATIONSHIPS

⇒ Generalization from «is_adl_primitivetype» DATE to «is_adl_primitivetype» DATE_TIME
 [Direction is 'Source -> Destination'.]

⇒ Generalization from «is_adl_primitivetype» TIME to «is_adl_primitivetype» DATE_TIME
 [Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ value : String Public
 [Is static False. Containment is Not Specified.]

DURATION

Class «is_adl_primitivetype» in package 'Data Value Types'

Represents a period of time with respect to a notional point in time, which is not specified. A sign may be used to indicate the duration is backwards in time rather than forwards.

DURATION
 Version 1.0 Phase 1.0 Proposed
 Linda created on 4/22/2012. Last modified 9/8/2014
 Extends QUANTITY

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from «is_adl_primitivetype» DURATION to QUANTITY
 [Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ duration_text : String Public
 Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)

A string representing an ISO8601 duration.

[Is static False. Containment is Not Specified.]

OPERATIONS

◆ is_strictly_comparable_to (other :) : Public
 [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

EHR_URI

Class in package 'Data Value Types'

A URI which has the scheme name ehr, and which can only reference elements in EHRs. Used to reference elements in an EHR, which may be the current one, or another.

EHR_URI
 Version 1.0 Phase 1.0 Proposed
 Linda created on 4/22/2012. Last modified 4/10/2013
 Extends URI_VALUE

OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Generalization from EHR_URI to URI_VALUE

[Direction is 'Source -> Destination'.]

ENCAPSULATED

Class in package 'Data Value Types'

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

ENCAPSULATED
 Version 1.0 Phase 1.0 Proposed
 Linda created on 4/22/2012. Last modified 5/18/2012
 Extends DATA_VALUE

OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Generalization from ENCAPSULATED to DATA_VALUE

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

➡ Generalization from PARSABLE to ENCAPSULATED

[Direction is 'Source -> Destination'.]

➡ Generalization from MULTIMEDIA to ENCAPSULATED

[Direction is 'Source -> Destination'.]

IDENTIFIER

Class in package 'Data Value Types'

A type for representing identifiers of real-world entities. Typical identifiers include drivers licence number, social security number, veterans affairs number, prescription id, order id, and so on. IDENTIFIER is used to represent any identifier of a real thing, issued by some authority or agency. IDENTIFIER is not used to express identifiers generated by the infrastructure to refer to information items; the types OBJECT_ID and OBJECT_REF and subtypes are defined for this purpose.

IDENTIFIER
 Version 1.0 Phase 1.0 Proposed
 Linda created on 4/22/2012. Last modified 4/10/2013
 Extends DATA_VALUE

OUTGOING STRUCTURAL RELATIONSHIPS

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from IDENTIFIER to DATA_VALUE

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ id : String Public

The identifier value. Often structured, according to the definition of the issuing authority's rules.

[Is static False. Containment is Not Specified.]

◆ type : CODED_TEXT Public

The identifier type, such as prescription id, or Social Security Number.

[Is static False. Containment is Not Specified.]

◆ issuer : String Public

Authority which issues the kind of id used in the id field of this object.

[Is static False. Containment is Not Specified.]

INTERVAL_VALUE

Class in package 'Data Value Types'

Generic class defining an interval (i.e. range) of a comparable type. An interval is used to define intervals of dates, times, quantities and so on. The type parameter, T, must be a descendant of the type ORDERED, which is necessary (but not sufficient) for instances to be compared (strictly_comparable is also needed). Without the INTERVAL class, quite a few more classes would be needed to express logical intervals, namely interval versions of all the date/time classes, and of quantity classes. Further, it allows the semantics of intervals to be stated in one place unequivocally, including the conditions for strict comparison.

INTERVAL_VALUE

Version 1.0 Phase 1.0 Proposed

Linda created on 5/1/2012. Last modified 4/13/2013

Extends DATA_VALUE

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from INTERVAL_VALUE to DATA_VALUE

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

⇒ Realization «bind» from <anonymous> to INTERVAL_VALUE

[Name is <T -> DATE_TIME>. Direction is 'Source -> Destination'.]

⇒ Aggregation from ORDERED_VALUE to INTERVAL_VALUE

[Direction is 'Destination -> Source'.]

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Generalization from INTERVAL_VALUE<T -> DATE_TIME> to INTERVAL_VALUE	[Direction is 'Source -> Destination'.]
⇒ Aggregation from ORDERED_VALUE to INTERVAL_VALUE	[Direction is 'Destination -> Source'.]

ATTRIBUTES	
◆ upper_unbounded : Boolean Public	[Is static False. Containment is Not Specified.]
◆ lower_unbounded : Boolean Public	[Is static False. Containment is Not Specified.]
◆ upper_included : Boolean Public	[Is static False. Containment is Not Specified.]
◆ lower_included : Boolean Public	[Is static False. Containment is Not Specified.]

MULTIMEDIA

Class in package 'Data Value Types'

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

MULTIMEDIA
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013
Extends ENCAPSULATED

OUTGOING STRUCTURAL RELATIONSHIPS	
◀ Generalization from MULTIMEDIA to ENCAPSULATED	[Direction is 'Source -> Destination'.]

ATTRIBUTES	
◆ alternate_text : Byte Public Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)	Text to display in lieu of multimedia display/replay. [Is static False. Containment is Not Specified.]
◆ data : Byte Public Multiplicity: ([0..*], Allow duplicates: 0, Is ordered: False)	

ATTRIBUTES	
The actual data that represents the multimedia item. If the multimedia item has a uri, then this is the data found at this uri.	[Is static False. Containment is Not Specified.]
◆ media_type : CODED_TEXT Public Data media type coded from the IANA MIME types code set.	[Is static False. Containment is .]
◆ uri : URI_VALUE Public Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False) URI reference to electronic information stored outside the record as a file, database entry etc, if supplied as a reference.	[Is static False. Containment is .]

ORDERED_VALUE

Class in package 'Data Value Types'

An abstract class defining the concept of ordered values, which includes ordinals as well as true quantities.

ORDERED_VALUE

Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 9/20/2012
Extends DATA_VALUE

OUTGOING STRUCTURAL RELATIONSHIPS	
Aggregation from ORDERED_VALUE to INTERVAL_VALUE	[Direction is 'Destination -> Source'.]
Aggregation from ORDERED_VALUE to INTERVAL_VALUE	[Direction is 'Destination -> Source'.]
Generalization from ORDERED_VALUE to DATA_VALUE	[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Generalization from ORDINAL to ORDERED_VALUE	[Direction is 'Source -> Destination'.]
⇒ Generalization from QUANTIFIED to ORDERED_VALUE	[Direction is 'Source -> Destination'.]

ORDINAL

Class in package 'Data Value Types'

Used to represent model rankings and scores, e.g. pain, Apgar values, 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

This class is 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, nitrates etc; for non-haemolysed blood {neg, trace, moderate}; for haemolysed blood small, moderate, large}.

ORDINAL

Version 1.0 Phase 1.0 Proposed

Linda created on 4/22/2012. Last modified 4/10/2013

Extends ORDERED_VALUE

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from ORDINAL to ORDERED_VALUE

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ symbol : CODED_TEXT Public

The 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 .

[Is static False. Containment is Not Specified.]

◆ value : Real Public

Value in ordered enumeration of values. Any integer value can be used.

[Is static False. Containment is Not Specified.]

PARSABLE

Class in package 'Data Value Types'

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

PARSABLE

Version 1.0 Phase 1.0 Proposed

Linda created on 4/22/2012. Last modified 4/10/2013

Extends ENCAPSULATED

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from PARSABLE to ENCAPSULATED

[Direction is 'Source -> Destination'.]

ATTRIBUTES
<p>◆ formalism : CODED_TEXT Public The name of the formalism, e.g. GLIF 1.0, Proforma. [Is static False. Containment is Not Specified.]</p>
<p>◆ value : String Public The string which can be parsed according to the given formalism. The value may validly be empty in some syntaxes. [Is static False. Containment is Not Specified.]</p>

PLAIN_TEXT

Class in package 'Data Value Types'

A string of characters, written in a particular language, without any associated coding.

PLAIN_TEXT
Version 1.0 Phase 1.0 Proposed
Linda created on 5/12/2012. Last modified 6/6/2012
Extends TEXT

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from PLAIN_TEXT to TEXT
[Direction is 'Source -> Destination'.]

PROPORTION

Class in package 'Data Value Types'

Models a ratio of values, i.e. where the numerator and denominator are both pure numbers. 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 distribution 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. E.g. visual acuity 6/24 is not a ratio. Should not be used for formulations.

PROPORTION
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013
Extends AMOUNT

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from PROPORTION to AMOUNT
[Direction is 'Source -> Destination'.]

ATTRIBUTES

 numerator : Real Public	The numerator of the ratio. [Is static False. Containment is Not Specified.]
 denominator : Real Public	The denominator of the ratio. [Is static False. Containment is Not Specified.]
 precision : Integer Public Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)	The precision to which the numerator and denominator values of the proportion are expressed, in terms of the number of decimal places. The value 0 implies an integral quantity. The value -1 implies no limit, i.e. any number of decimal places. [Is static False. Containment is Not Specified.]
 type : CODED_TEXT Public	Indicates the semantic type of the proportion. Valid values include: ratio, unitary, percent, fraction, integer_fraction [Is static False. Containment is Not Specified.]

QUANTIFIED

Class in package 'Data Value Types'

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

QUANTIFIED
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013
Extends ORDERED_VALUE

OUTGOING STRUCTURAL RELATIONSHIPS	
 Generalization from QUANTIFIED to ORDERED_VALUE	[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS	
 Generalization from AMOUNT to QUANTIFIED	[Direction is 'Source -> Destination'.]
 Generalization from «is_adl_primitivetype» DATE_TIME to QUANTIFIED	[Direction is 'Source -> Destination'.]

CONNECTORS	

CONNECTORS **Dependency** Source -> Destination

From: QUANTIFIED : Class, Public
 To: QuantifiedValueStatusEnum : Enumeration, Public

ATTRIBUTES **value_status** : String Public

Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)

The optional status of the magnitude with possible 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 =.

[Is static False. Containment is Not Specified.]

QuantifiedValueStatusEnum

Enumeration «enumeration» in package 'Data Value Types'

QuantifiedValueStatusEnum
 Version 1.0 Phase 1.0 Proposed
 ZelM created on 10/5/2012. Last modified 3/22/2013
 Extends String

CONNECTORS **Dependency** Source -> Destination

From: QUANTIFIED : Class, Public
 To: QuantifiedValueStatusEnum : Enumeration, Public

ATTRIBUTES = : Public

[Stereotype is «enum». Is static False. Containment is Not Specified.]

 < : Public

[Stereotype is «enum». Is static False. Containment is Not Specified.]

 > : Public

[Stereotype is «enum». Is static False. Containment is Not Specified.]

 <= : Public

[Stereotype is «enum». Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ >= : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ ~ : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]

QUANTITY

Class in package 'Data Value Types'

A type representing a quantity, i.e. a measure which includes both a numeric magnitude and a coded units. The quantity's units may be represented in any code system, including the Unified Code for Units of Measure (UCUM), and SNOMED CT. Quantities can also be used for time durations, where the units represents a temporal measure (e.g. seconds, minutes, hours, days, months, years).

QUANTITY
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013
Extends AMOUNT

OUTGOING STRUCTURAL RELATIONSHIPS	
◀ Generalization from QUANTITY to AMOUNT	[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Generalization from «is_adl_primitivetype» DURATION to QUANTITY	[Direction is 'Source -> Destination'.]

ATTRIBUTES	
◆ value : Real Public	
The numeric size of the quantity.	[Is static False. Containment is Not Specified.]
◆ units : CODED_TEXT Public	[Is static False. Containment is Not Specified.]
◆ precision : Integer Public Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)	
The precision to which the value of the quantity is expressed, as a number of significant figures.	[Is static False. Containment is Not Specified.]

TERM_MAPPING

Class in package 'Data Value Types'

Represents a coded term mapped to a CODEABLE_TEXT, and the relative match of the target term with respect to the mapped item. Plain or coded text items may appear for which one or more mappings in alternative terminologies are required.

Used for adding classification terms (e.g. adding ICD classifiers to SNOMED CT descriptive terms), or mapping into equivalents in other terminologies (e.g. across nursing vocabularies).

TERM_MAPPING
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Aggregation from TERM_MAPPING to TEXT

[Direction is 'Destination -> Source'.]

CONNECTORS

↗ Dependency Source -> Destination

From: TERM_MAPPING : Class, Public

To: TermMappingMatchEnum : Enumeration, Public

ATTRIBUTES

◆ match : Character Public

The relative accuracy with which the target term matches with the respective mapped text item. Valid values include:

>: the mapping is to a broader term, e.g. original text = arbovirus infection , target = viral infection

=: the mapping is equivalent to the original item

<: the mapping is to a narrower term, e.g. original text = diabetes , mapping = diabetes mellitus .

? : the kind of mapping is unknown.

The first three values are taken from the ISO standards 2788 (Guide to Establishment and development of monolingual thesauri) and 5964 (Guide to Establishment and development of multilingual thesauri).

[Is static False. Containment is Not Specified.]

◆ purpose : CODED_TEXT Public

Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)

The purpose of the mapping e.g. automated data mining , billing, interoperability.

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

↙ Association (direction: Source -> Destination)

Source: Private (Class) TERM_MAPPING

Target: Public target (Class) CODED_TEXT
«is_adl_primitivetype»
Cardinality: [1..1]

TermMappingMatchEnum

Enumeration «enumeration» in package 'Data Value Types'

TermMappingMatchEnum
Version 1.0 Phase 1.0 Proposed
ZelM created on 10/5/2012. Last modified 3/22/2013
Extends Character

CONNECTORS

↗ **Dependency** Source -> Destination
 From: TERM_MAPPING : Class, Public
 To: TermMappingMatchEnum : Enumeration, Public

ATTRIBUTES

◊ > : Public
 [Stereotype is «enum». Is static False. Containment is Not Specified.]

◊ = : Public
 [Stereotype is «enum». Is static False. Containment is Not Specified.]

◊ < : Public
 [Stereotype is «enum». Is static False. Containment is Not Specified.]

◊ ? : Public
 [Stereotype is «enum». Is static False. Containment is Not Specified.]

TEXT

Class in package 'Data Value Types'

A text item, which may contain any amount of legal characters. TEXT may either be coded, codeable or plain.

A text item, which may either be a plain string value (with optional term mappings), or text that has been coded according to a given terminology.

TEXT
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 7/12/2013
Extends DATA_VALUE

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from TEXT to DATA_VALUE
 [Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Generalization from «is_adl_primitivetype» CODED_TEXT to TEXT	[Direction is 'Source -> Destination'.]
⇒ Aggregation from TERM_MAPPING to TEXT	[Direction is 'Destination -> Source'.]
⇒ Generalization from PLAIN_TEXT to TEXT	[Direction is 'Source -> Destination'.]

ATTRIBUTES	
◆ value : String Public	[Is static False. Containment is Not Specified.]
◆ language : CODED_TEXT Public Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)	An optional indicator of the localised language in which the value is written. Only used when either the text object is in a different language from the enclosing ENTRY, or the text object is being used outside of an ENTRY or other enclosing structure which indicates the language. [Is static False. Containment is .]

TIME

Class «is_adl_primitivetype» in package 'Data Value Types'

Represents an absolute point in time from an origin usually interpreted as meaning the start of the current day, specified to the second. Semantics defined by ISO 8601. Used for recording real world times, rather than scientifically measured fine amounts of time. The partial form is used for approximate times of events and substance administrations.

value constrained to: The ISO 8601 time string.

TIME
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 9/8/2014
Extends DATE_TIME

OUTGOING STRUCTURAL RELATIONSHIPS	
⇐ Generalization from «is_adl_primitivetype» TIME to «is_adl_primitivetype» DATE_TIME	[Direction is 'Source -> Destination'.]

URI_VALUE

Class in package 'Data Value Types'

A data type used for referring to information resources. The URI type allows data values which are references to objects on the world wide web to be created. Its specialisation, EHR_URI, enables any element to be identified in the same way

as other objects on the web. The EHR_URI type is convenient, because it is a string, like any other URI, and is therefore easily transportable and processable. Because it has its own scheme space (i.e. ehr), instances can be globally unique, as long as EHR identification is globally unique. EHR_URIs are used to express all runtime paths in the EHR.

URI_VALUE
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 7/12/2013
Extends DATA_VALUE

OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Generalization from URI_VALUE to DATA_VALUE

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

➡ Generalization from EHR_URI to URI_VALUE

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ value : URI Public

[Is static False. Containment is Not Specified.]

YESNO

Class in package 'Data Value Types'

Values which represent boolean data, such as true/false or yes/no. 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. The YESNO class should not be used as a replacement for naively modelled enumerated types such as male/female etc. Such values should be coded.

YESNO
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013
Extends DATA_VALUE

OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Generalization from YESNO to DATA_VALUE

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ value : Boolean Public

Boolean value of this item. Actual values may be language or implementation dependent.

[Is static False. Containment is Not Specified.]

<anonymous>

Text in package 'CIMI Reference Model'

<anonymous>
Version 1.0 Phase 1.0 Proposed
zelm created on 3/22/2013. Last modified 3/22/2013
Extends

CIMI Party Model diagram

Class diagram in package 'CIMI Reference Model'

The CIMI Party Model diagram shows the generic concepts of PARTY, ROLE and related details such as contacts and addresses.

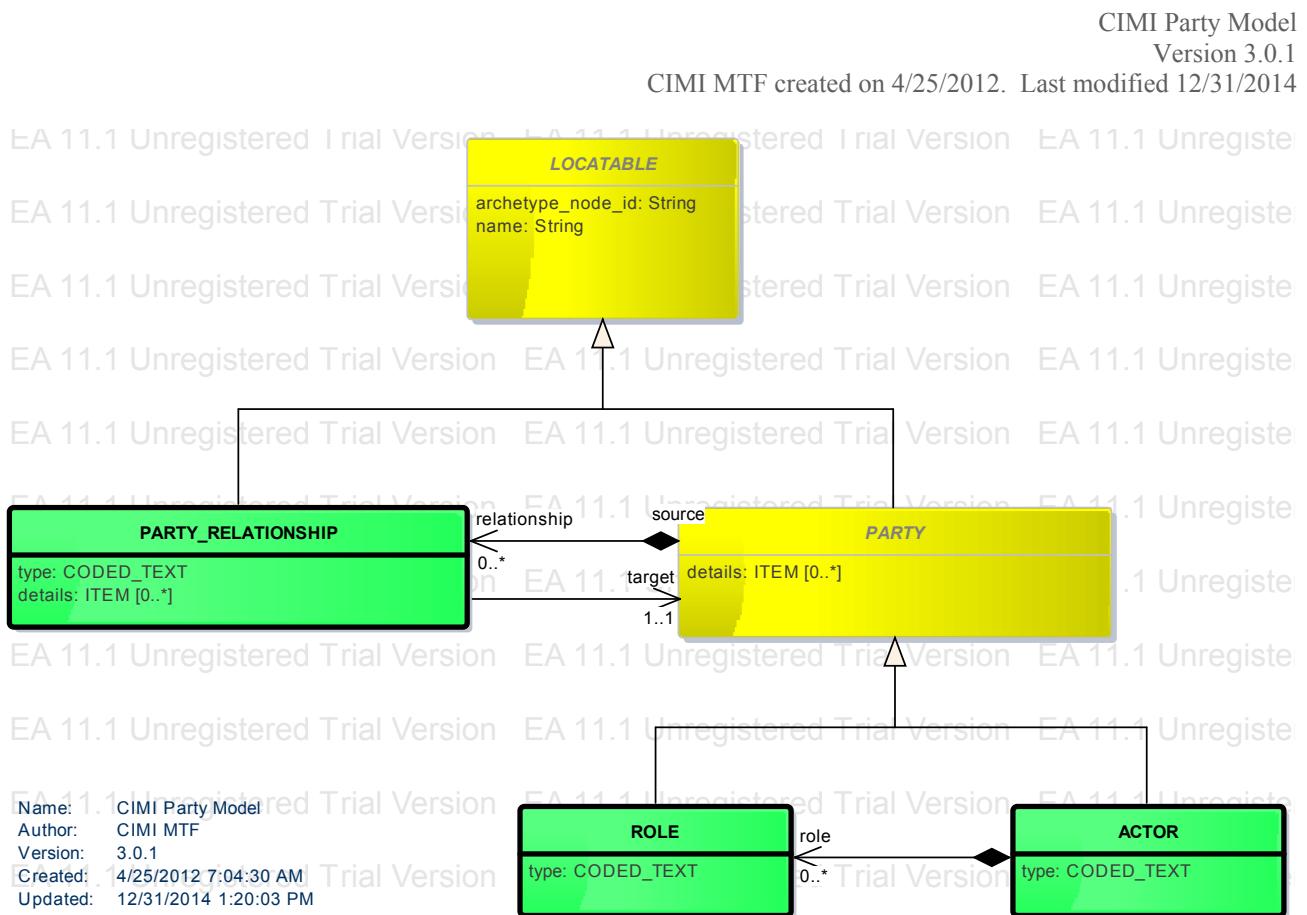


Figure 5: CIMI Party Model

LOCATABLE

Class «archetype_parent» in package 'Core'

The root class of all information model classes that can be archetyped. Most classes in the CIMI reference model inherit from the LOCATABLE class, which defines the idea of locatability in an archetyped structure. LOCATABLE defines a runtime name and an archetype_node_id.

LOCATABLE
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 5/22/2014

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Generalization from PARTICIPATION to «archetype_parent» LOCATABLE	[Direction is 'Source -> Destination'.]
⇒ Generalization from PARTY to «archetype_parent» LOCATABLE	[Direction is 'Source -> Destination'.]
⇒ Generalization from ITEM to «archetype_parent» LOCATABLE	[Direction is 'Source -> Destination'.]
⇒ Generalization from PARTY_RELATIONSHIP to «archetype_parent» LOCATABLE	[Direction is 'Source -> Destination'.]

CONNECTORS	
↗ Trace «trace» Source -> Destination From: id : Property, Public To: LOCATABLE : Class, Public	
↗ Trace «trace» Source -> Destination From: name : Property, Public To: LOCATABLE : Class, Public	

ATTRIBUTES	
◆ archetype_node_id : String Public	
The design-time archetype id of this node taken from its generating archetype. This archetype node id is used to build archetype paths. It is always in the form of an at code, e.g. at0005 . This value enables a "standardised" name for this node to be generated, by referring to the generating archetype local ontology. At an archetype root point, the value of this attribute is always the stringified form of the archetype_id found in the archetype_details object.	[Stereotype is «is_im_infrastructure». Is static False. Containment is Not Specified.]
◆ name : String Public	
The runtime name of this fragment, used to build runtime paths. This is the term provided via a clinical application or batch process to name this EHR construct. Its retention in the EHR faithfully preserves the original label by which this entry was known to end users.	[Is static False. Containment is Not Specified.]

ASSOCIATIONS	
↙ Association (direction: Source -> Destination) Source: Private (Class) LOCATABLE «archetype_parent»	Target: Public «is_im_infrastructure» archetype_details (Class) ARCHETYPED Cardinality: [0..1]

ACTOR

Class in package 'Party'

An ancestor of all real-world types, including people and organisations. An actor is any real-world entity capable of taking on a role.

GROUP

A group is a real world group of parties which is created by another party (usually an organisation) for some specific purpose. A typical clinical example is that of the specialist care team, e.g. cardiology team. The members of the group usually work together.

AGENT (DEVICE)

A generic concept of any kind of agent, including devices, software systems, but not humans or organisations.

PERSON

The generic description of a person. PERSON provides a dedicated type to which Person archetypes can be targeted.

ORGANISATION

A generic description of an organisation. An organisation is a legally constituted body whose existence (in general) outlives the existence of parties considered to be part of it.

ACTOR

Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013
Extends PARTY

OUTGOING STRUCTURAL RELATIONSHIPS

◀ Generalization from ACTOR to PARTY

[Direction is 'Source -> Destination'.]

CONNECTORS

↗ Dependency Destination -> Source

From: ACTOR : Class, Public
To: Entity : Boundary, Public

ATTRIBUTES

◆ type : CODED_TEXT Public

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

↙ Association (direction: Source -> Destination)

Source: Public (Class) ACTOR

Target: Public role (Class) ROLE
Cardinality: [0..*]

PARTY

Class in package 'Party'

An ancestor of all party types, including real world entities and their roles. A party is any entity which can participate in an activity. The name attribute inherited from LOCATABLE is used to indicate the actual type of the party (note that the actual names, i.e. identities of parties are indicated in the identities attribute, not the name attribute).

PARTY
 Version 1.0 Phase 1.0 Proposed
 Linda created on 4/22/2012. Last modified 5/22/2014
 Extends LOCATABLE

OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Generalization from PARTY to «archetype_parent» LOCATABLE

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

➡ Generalization from ROLE to PARTY

[Direction is 'Source -> Destination'.]

➡ Generalization from ACTOR to PARTY

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ details : ITEM Public

Multiplicity: ([0..*], Allow duplicates: 0, Is ordered: False)

All other details for this party. These details are usually archetyped.

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

✓ Association (direction: Source -> Destination)

Source: Public source (Class) PARTY

Target: Public relationship (Class)

PARTY_RELATIONSHIP

Cardinality: [0..*]

✓ Association (direction: Source -> Destination)

Source: Public (Class) PARTICIPATION

Target: Public party (Class) PARTY

Cardinality: [1..1]

✓ Association (direction: Source -> Destination)

Source: Public (Class) PARTY_RELATIONSHIP

Target: Public target (Class) PARTY

Cardinality: [1..1]

PARTY_RELATIONSHIP

Class in package 'Party'

A generic description of a relationship between the source and target parties.

PARTY_RELATIONSHIP
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013
Extends LOCATABLE

OUTGOING STRUCTURAL RELATIONSHIPS

- ◀ Generalization from PARTY_RELATIONSHIP to «archetype_parent» LOCATABLE

[Direction is 'Source -> Destination'.]

CONNECTORS

- ↗ Dependency Destination -> Source

From: PARTY_RELATIONSHIP : Class, Public
To: RoleLink : Boundary, Public

ATTRIBUTES

- ◆ type : CODED_TEXT Public

The detailed description of the relationship

[Is static False. Containment is Not Specified.]

- ◆ details : ITEM Public

Multiplicity: ([0..*], Allow duplicates: 0, Is ordered: False)

The detailed description of the relationship.

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

- ↙ Association (direction: Source -> Destination)

Source: Public (Class) PARTY_RELATIONSHIP

Target: Public target (Class) PARTY
Cardinality: [1..1]

- ↙ Association (direction: Source -> Destination)

Source: Public source (Class) PARTY

Target: Public relationship (Class)
PARTY_RELATIONSHIP
Cardinality: [0..*]

ROLE

Class in package 'Party'

A generic description of a role performed by an actor. The role corresponds to a competency of the party. Roles are used to define the responsibilities undertaken by a party for a purpose. Roles should have credentials qualifying the performer to perform the role.

ROLE
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013
Extends PARTY

OUTGOING STRUCTURAL RELATIONSHIPS

◀ Generalization from ROLE to PARTY

[Direction is 'Source -> Destination'.]

CONNECTORS

↗ Dependency Destination -> Source

From: ROLE : Class, Public
To: Role : Boundary, Public

ATTRIBUTES

◆ type : CODED_TEXT Public

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

↙ Association (direction: Source -> Destination)

Source: Public (Class) ACTOR

Target: Public role (Class) ROLE
Cardinality: [0..*]

<anonymous>

Text in package 'CIMI Reference Model'

<anonymous>
Version 1.0 Phase 1.0 Proposed
zelm created on 3/22/2013. Last modified 3/22/2013
Extends

CIMI Primitive Types diagram

Class diagram in package 'CIMI Reference Model'

The CIMI Primitive Types diagram shows the basic types that are assumed in external type systems.

CIMI Primitive Types
Version 3.0.1
Linda created on 5/1/2012. Last modified 12/31/2014

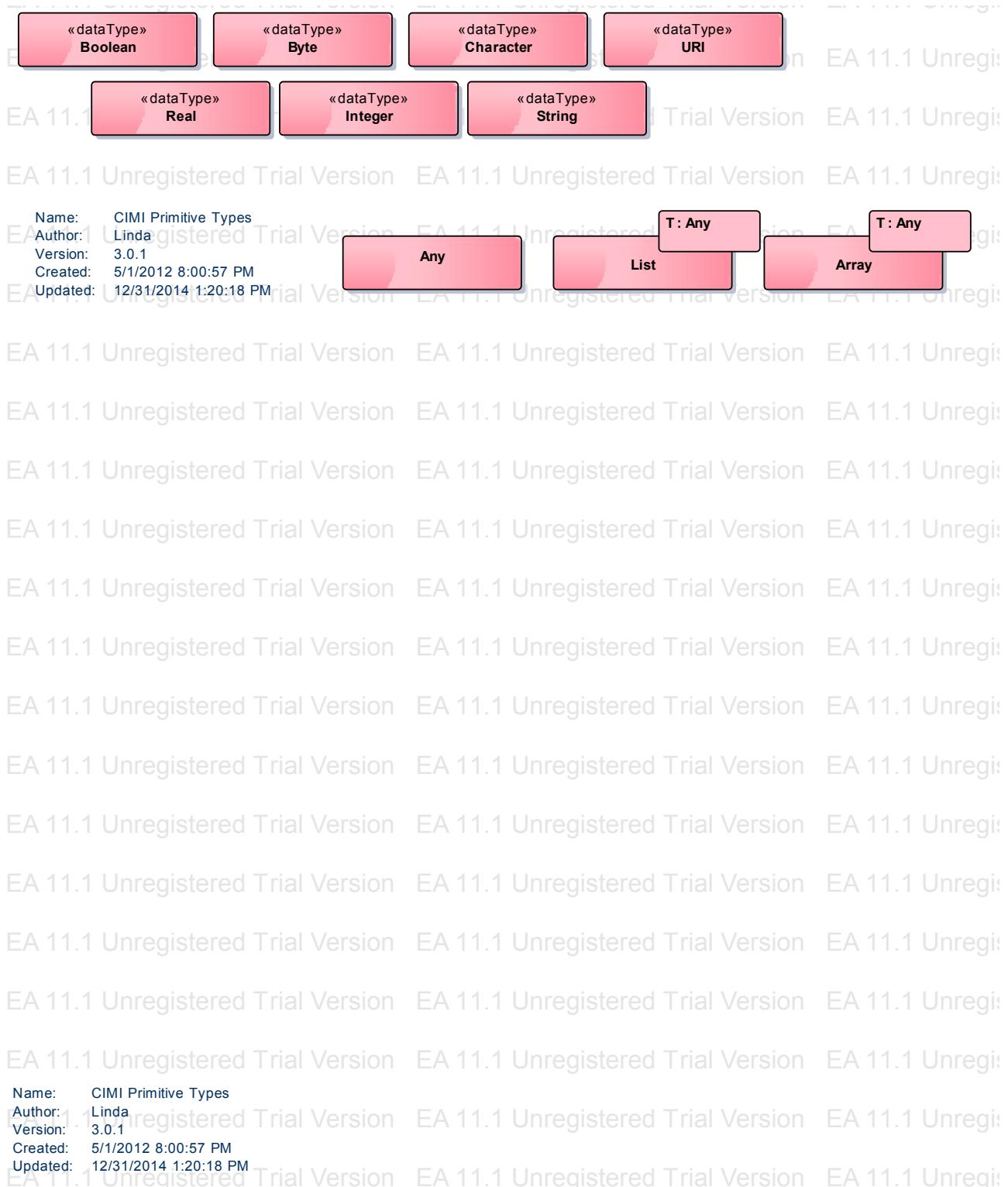


Figure 6: CIMI Primitive Types

<anonymous>

Text in package 'CIMI Reference Model'

<anonymous>
Version 1.0 Phase 1.0 Proposed
mrf7578 created on 9/8/2014. Last modified 9/8/2014
Extends

<anonymous>*Text in package 'CIMI Reference Model'*

<anonymous>
Version 1.0 Phase 1.0 Proposed
Zelm created on 3/22/2013. Last modified 3/22/2013
Extends

Any*Class in package 'Primitive Types'*

Any
Version 1.0 Phase 1.0 Proposed
Michael van der Zel created on 9/17/2014. Last modified 9/17/2014

Array*Class in package 'Primitive Types'*

Array
Version 1.0 Phase 1.0 Proposed
Linda created on 9/17/2014. Last modified 9/17/2014

List*Class in package 'Primitive Types'*

List
Version 1.0 Phase 1.0 Proposed
Linda created on 9/17/2014. Last modified 9/17/2014

Boolean*DataType in package 'Primitive Types'*

Boolean
Version 1.0 Phase 1.0 Proposed
ZelM created on 5/30/2013. Last modified 9/17/2014

Byte*DataType in package 'Primitive Types'*

Byte
Version 1.0 Phase 1.0 Proposed
ZelM created on 5/30/2013. Last modified 9/17/2014

Character*DataType in package 'Primitive Types'*

Character
Version 1.0 Phase 1.0 Proposed
ZelM created on 5/30/2013. Last modified 9/17/2014

Integer

DataType in package 'Primitive Types'

Integer
Version 1.0 Phase 1.0 Proposed
ZelM created on 5/30/2013. Last modified 9/17/2014

Real

DataType in package 'Primitive Types'

Real
Version 1.0 Phase 1.0 Proposed
ZelM created on 5/30/2013. Last modified 9/17/2014

String

DataType in package 'Primitive Types'

String
Version 1.0 Phase 1.0 Proposed
ZelM created on 5/30/2013. Last modified 9/17/2014

URI

DataType in package 'Primitive Types'

URI
Version 1.0 Phase 1.0 Proposed
Linda created on 7/12/2013. Last modified 9/17/2014

Core

Package «archetype_rm_closure» in package 'CIMI Reference Model'

The Core Reference Model package includes the main classes in the CIMI reference model upon which clinical models will be defined. These include the concrete classes Composition, Section, Entry, Cluster and Element.

Core
Version 1.0 Phase 1.0 Proposed
Linda created on 5/1/2012. Last modified 6/6/2013
Alias Core Reference Model

ARCHETYPED

Class in package 'Core'

Archetypes act as the configuration basis for the particular structures of instances defined by the reference model. To enable archetypes to be used to create valid data, key classes in the reference model act as root points for archotyping. These classes have the archetype_details attribute set. An instance of the class ARCHETYPED contains the relevant archetype identification information, allowing archetypes to be matched up with data instances.

ARCHETYPED
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 5/22/2014

ATTRIBUTES

◆ archetype_id : String Public

Globally unique archetype identifier.

The identifier for archetypes. Ideally these would identify globally unique archetypes. Lexical form:
rm_originator '-' rm_name '-' rm_entity '.' concept_name { '-' specialisation }* .v' number

[Stereotype is «is_im_infrastructure». Is static False. Containment is Not Specified.]

◆ rm_version : String Public Const = 3.0.1

Version of the CIMI reference model used to create this object.
Expressed in terms of the release version string, e.g. 1.0 , 1.2.4 .

[Stereotype is «is_im_infrastructure». Is static False. Containment is Not Specified.]

ASSOCIATIONS

↙ Association (direction: Source -> Destination)

Source: Private (Class) LOCATABLE «archetype_parent»

Target: Public «is_im_infrastructure»
archetype_details (Class) ARCHETYPED
Cardinality: [0..1]

CORE_LOCATABLE*Class in package 'Core'*

CORE_LOCATABLE
Version 1.0 Phase 1.0 Proposed
ZelM created on 1/20/2013. Last modified 5/23/2013

CONNECTORS

↗ Dependency Destination -> Source
From: CORE_LOCATABLE : Class, Public
To: Act : Boundary, Public

ELEMENT

Class in package 'Core'

A type of data ITEM, which does not itself contain ITEMS.

ELEMENT
 Version 1.0 Phase 1.0 Proposed
 Linda created on 4/22/2012. Last modified 5/22/2014
 Extends ITEM

OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Generalization from ELEMENT to ITEM

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

➡ Generalization from name to ELEMENT

[Direction is 'Source -> Destination'.]

➡ Generalization from status to ELEMENT

[Direction is 'Source -> Destination'.]

➡ Generalization from datetime range to ELEMENT

[Direction is 'Source -> Destination'.]

CONNECTORS

↗ Trace «trace» Source -> Destination

From: datatype : Property, Public

To: ELEMENT : Class, Public value

↗ Dependency «instantiate» Source -> Destination

From: LeafConcept : Class, Public

To: ELEMENT : Class, Public

ATTRIBUTES

◆ null_flavor : CODED_TEXT Public

Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)

[Is static False. Containment is Not Specified.]

◆ value : DATA_VALUE Public

Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)

[Is static False. Containment is Not Specified.]

ITEM*Class in package 'Core'*

The abstract parent of CLUSTER and ELEMENT representation classes.

ITEM

Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 5/22/2014
Extends LOCATABLE

OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Generalization from ITEM to «archetype_parent» LOCATABLE

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

➡ Generalization from ELEMENT to ITEM

[Direction is 'Source -> Destination'.]

➡ Generalization from ITEM_GROUP to ITEM

[Direction is 'Source -> Destination'.]

➡ Generalization from details to ITEM

[Direction is 'Source -> Destination'.]

ASSOCIATIONS

✍ Association (direction: Source -> Destination)

Source: Private (Class) ITEM_GROUP

Target: Public item (Class) ITEM

Cardinality: [1..*]

ITEM_GROUP

Class in package 'Core'

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

ITEM_GROUP

Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 12/18/2014
Extends ITEM

OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Generalization from ITEM_GROUP to ITEM

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

➡ Generalization from observable to ITEM_GROUP

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Generalization from result to ITEM_GROUP	[Direction is 'Source -> Destination'.]
⇒ Generalization from timing to ITEM_GROUP	[Direction is 'Source -> Destination'.]
⇒ Aggregation from PARTICIPATION to ITEM_GROUP	[Direction is 'Destination -> Source'.]

CONNECTORS	
↗ Dependency «instantiate» Source -> Destination From: RootConcept : Class, Public To: ITEM_GROUP : Class, Public	
↗ Dependency «instantiate» Source -> Destination From: ContainerConcept : Class, Public To: ITEM_GROUP : Class, Public	

ASSOCIATIONS	
↙ Association (direction: Source -> Destination) Source: Public source (Class) ITEM_GROUP	Target: Public link (Class) LINK Cardinality: [0..*]
↙ Association (direction: Source -> Destination) Source: Private (Class) ITEM_GROUP	Target: Public item (Class) ITEM Cardinality: [1..*]
↙ Association (direction: Source -> Destination) Target can be archetype_id or constraint (business rules/query language) on instance. Target is the constraint on "type of instance" referenced at runtime. target LOCATABLE "type of instance" is constrained in the AOM. Just like terminology binding. Target is an association to a LOCATABLE.	
Source: Public (Class) LINK	Target: Public target (Class) ITEM_GROUP Cardinality: [1..1]

LINK

Class in package 'Core'

The LINK type defines a logical relationship between two items, such as two ENTRYs or an ENTRY and a

COMPOSITION. Links can be used across compositions, and across EHRs. Links can potentially be used between interior (i.e. non archetype root) nodes, although this probably should be prevented in archetypes. Multiple LINKs can be attached to the root object of any archetyped structure to give the effect of a 1->N link. 1:1 and 1:N relationships between archetyped content elements (e.g. ENTRYs) can be expressed by using one, or more than one, respectively, LINKs. Chains of links can be used to see problem threads or other logical groupings of items. Links should be used between archetyped structures only, i.e. between objects representing complete domain concepts, because relationships between sub-elements of whole concepts are not necessarily meaningful. Sensible /*links only exist between whole ITEM_GROUPS and/or PARTICIPATIONS.

LINK

Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 12/18/2014

CONNECTORS **Dependency** Destination -> Source

From: LINK : Class, Public
To: ActRelationship : Boundary, Public

ATTRIBUTES

meaning : TEXT Public

Used to describe the relationship between the source and the target of the link (usually in clinical terms) - such as the relationship between test results and an order, follow-up to a consultation and so on. The meaning of each link falls under one of the following categories: generic, documenting and reporting, organisational, clinical , circumstantial , and view management .

[Is static False. Containment is .]

details : ITEM Public

Multiplicity: ([0..*], Allow duplicates: 0, Is ordered: False)

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Target can be archetype_id or constraint (business rules/query language) on instance.

Target is the constraint on "type of instance" referenced at runtime.

target LOCATABLE "type of instance" is constrained in the AOM. Just like terminology binding.

Target is an association to a LOCATABLE.

Source: Public (Class) LINK

Target: Public target (Class) ITEM_GROUP

Cardinality: [1..1]

Association (direction: Source -> Destination)

Source: Public source (Class) ITEM_GROUP

Target: Public link (Class) LINK

Cardinality: [0..*]

LOCATABLE

Class «archetype_parent» in package 'Core'

The root class of all information model classes that can be archetyped. Most classes in the CIMI reference model inherit from the LOCATABLE class, which defines the idea of locatability in an archetyped structure. LOCATABLE defines a runtime name and an archetype_node_id.

LOCATABLE
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 5/22/2014

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Generalization from PARTICIPATION to «archetype_parent» LOCATABLE [Direction is 'Source -> Destination'.]
- ⇒ Generalization from PARTY to «archetype_parent» LOCATABLE [Direction is 'Source -> Destination'.]
- ⇒ Generalization from ITEM to «archetype_parent» LOCATABLE [Direction is 'Source -> Destination'.]
- ⇒ Generalization from PARTY_RELATIONSHIP to «archetype_parent» LOCATABLE [Direction is 'Source -> Destination'.]

CONNECTORS

- ↗ Trace «trace» Source -> Destination

From: id : Property, Public
To: LOCATABLE : Class, Public
- ↗ Trace «trace» Source -> Destination

From: name : Property, Public
To: LOCATABLE : Class, Public

ATTRIBUTES

- ◊ archetype_node_id : String Public

The design-time archetype id of this node taken from its generating archetype. This archetype node id is used to build archetype paths. It is always in the form of an at code, e.g. at0005. This value enables a "standardised" name for this node to be generated, by referring to the generating archetype local ontology. At an archetype root point, the value of this attribute is always the stringified form of the archetype_id found in the archetype_details object.

[Stereotype is «is_im_infrastructure». Is static False. Containment is Not Specified.]

- ◊ name : String Public

The runtime name of this fragment, used to build runtime paths. This is the term provided via a clinical application or batch process to name this EHR construct. Its retention in the EHR faithfully preserves the original label by which this entry was known to end users.

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Private (Class) LOCATABLE «archetype_parent»

Target: Public «is_im_infrastructure»
archetype_details (Class) ARCHETYPED

Cardinality: [0..1]

PARTICIPATION

Class in package 'Core'

Used to represent any participation of a Party (e.g. any Actor or Role) in some activity, 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.

PARTICIPATION
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 5/22/2014
Extends LOCATABLE

OUTGOING STRUCTURAL RELATIONSHIPS

Generalization from PARTICIPATION to «archetype_parent» LOCATABLE

[Direction is 'Source -> Destination'.]

Aggregation from PARTICIPATION to ITEM_GROUP

[Direction is 'Destination -> Source'.]

CONNECTORS

Dependency Destination -> Source

From: PARTICIPATION : Class, Public

To: Participation : Boundary, Public

ATTRIBUTES

function : CODED_TEXT Public

The function of the Party in this participation. Please note that a given party might participate in more than one way in a particular activity.

[Is static False. Containment is Not Specified.]

details : ITEM Public

Multiplicity: ([0..*], Allow duplicates: 0, Is ordered: False)

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

ASSOCIATIONS	
Source: Public (Class) PARTICIPATION	Target: Public party (Class) PARTY Cardinality: [1..1]

Data Value Types

Package in package 'CIMI Reference Model'

The Data Value Type package includes a set of clearly defined data types, which underly all other models. These data types provides both general and clinically specific types required for all kinds of health information. The following categories of data types are defined in the data types reference model.

Text: plain text, codeable text, code phrase

Quantities: any ordered type including ordinal values (used for representing symbolic ordered values such as +, ++, +++), measured quantities with values and units.

Date/times: date, time, date-time types, and partial date/time types.

Encapsulated data: multimedia, parsable content.

Basic types: boolean, URI, identifier.

Data Value Types
Version 1.0 Phase 1.0 Proposed
Linda created on 5/1/2012. Last modified 2/28/2013

AMOUNT

Class in package 'Data Value Types'

Abstract class defining the concept of relative quantified amounts'. For relative quantities, the '+' and '-' operators are defined (unlike descendants of ABSOLUTE_QUANTITY, such as the date/time types).

AMOUNT
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013
Extends QUANTIFIED

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from AMOUNT to QUANTIFIED

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

⇒ Generalization from COUNT to AMOUNT

[Direction is 'Source -> Destination'.]

⇒ Generalization from QUANTITY to AMOUNT

[Direction is 'Source -> Destination'.]

⇒ Generalization from PROPORTION to AMOUNT

[Direction is 'Source -> Destination'.]

ATTRIBUTES

accuracy : Real Public

Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)

Accuracy of the measurement, expressed either as a half-range percent value (i.e. accuracy_is_percent = True) or a half-range quantity. A value of 0 means that accuracy was

ATTRIBUTES
<p>not recorded.</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>accuracy_is_percent : Boolean Public Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)</p> <p>If True, this indicates that when this object was created, accuracy was recorded as a percent value; if False, as an absolute quantity value.</p> <p>[Is static False. Containment is Not Specified.]</p>

CODED_TEXT

Class «is_adl_primitivetype» in package 'Data Value Types'

A reference to a class, category or individual that is described in an external terminology. Every CODED_TEXT instance must either have a *uri*, a *code* or both. When a *uri* is present in a CODED_TEXT instance, it will be treated as the instance identity — any two CODED_TEXT instances reference the same concept if they have the same *uri*, and the remaining fields will be ignored. If a *uri* is not included in a CODED_TEXT instance, the instance identity is the *terminology_id* / *code* pair. Any two CODED_TEXT instances reference the same concept if they (a) both have no *uri* and (b) have the same *terminology_id* and *code*. The *terminology_version* and *term* attributes are strictly informative and play no role in determining the concept referent.

CODED_TEXT
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 9/8/2014
Extends TEXT

OUTGOING STRUCTURAL RELATIONSHIPS
<p>↳ Generalization from «is_adl_primitivetype» CODED_TEXT to TEXT</p> <p>[Direction is 'Source -> Destination'.]</p>

ATTRIBUTES
<p>uri : URI Public Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)</p> <p>A URI that uniquely identifies the referenced concept. Examples: http://snomed.info/id/74400008, http://loinc.org/id/2951-2, http://id.loc.gov/vocabulary/iso639-1/de</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>terminology_id : String Public</p> <p>"A locally unique identifier for the namespace from which code was derived. Examples: SNOMED_CT, LOINC, ISO639-1</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>code : String Public</p> <p>A code that uniquely identifies the referenced concept within the context of terminology_id. Examples: 74400008, 2951-2, de</p> <p>[Is static False. Containment is Not Specified.]</p>

ATTRIBUTES
<p>◆ terminology_version : String Public Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)</p> <p>"The URI of a terminology or terminology version from which the meaning of the code was determined for the purposes of this record. Examples: http://snomed.info/sct/900000000000207008, http://snomed.info/sct/900000000000207008/version/20130731, http://loinc.org/, http://loinc.org/246, http://id.loc.gov/vocabulary/iso639-1</p> <p>[Is static False. Containment is Not Specified.]</p>
<p>◆ term : String Public Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)</p> <p>A human readable string that conveys the intended meaning of the code. Examples: 'Appendicitis', 'Appendicitis (Finding)', 'inflamación aguda del apéndice', 'Serum Sodium', 'Plasma Serum Sodium', 'German', 'Deutsch'</p> <p>[Is static False. Containment is Not Specified.]</p>

ASSOCIATIONS
<p>✍ Association (direction: Source -> Destination)</p> <p>Source: Private (Class) TERM_MAPPING Target: Public target (Class) CODED_TEXT «is_adl_primitivetype» Cardinality: [1..1]</p>

COUNT

Class in package 'Data Value Types'

A countable quantity. Used for countable types, such as number of pregnancies, number of steps (taken by a physiotherapy patient), and number of cigarettes smoked in a day. Not to be used for amounts of physical entities (which have units).

COUNT
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013
Extends AMOUNT

OUTGOING STRUCTURAL RELATIONSHIPS
<p>◀ Generalization from COUNT to AMOUNT</p> <p>[Direction is 'Source -> Destination'.]</p>

ATTRIBUTES
<p>◆ value : Integer Public</p> <p>[Is static False. Containment is Not Specified.]</p>

DATA_VALUE

Class in package 'Data Value Types'

Serves as a common ancestor of all data value types in the CIMI model.

DATA_VALUE

Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 9/13/2012

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Generalization from INTERVAL_VALUE to DATA_VALUE	[Direction is 'Source -> Destination'.]
⇒ Generalization from ENCAPSULATED to DATA_VALUE	[Direction is 'Source -> Destination'.]
⇒ Generalization from YESNO to DATA_VALUE	[Direction is 'Source -> Destination'.]
⇒ Generalization from URI_VALUE to DATA_VALUE	[Direction is 'Source -> Destination'.]
⇒ Generalization from TEXT to DATA_VALUE	[Direction is 'Source -> Destination'.]
⇒ Generalization from IDENTIFIER to DATA_VALUE	[Direction is 'Source -> Destination'.]
⇒ Generalization from ORDERED_VALUE to DATA_VALUE	[Direction is 'Source -> Destination'.]

DATE

Class «is_adl_primitivetype» in package 'Data Value Types'

Represents an absolute point in time, as measured on the Gregorian calendar, and specified only to the day. Semantics defined by ISO 8601. Used for recording dates in real world time. The partial form is used for approximate birth dates, dates of death, etc.

value constrained to: The ISO 8601 date string.

DATE

Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 9/8/2014
Extends DATE_TIME

OUTGOING STRUCTURAL RELATIONSHIPS	
← Generalization from «is_adl_primitivetype» DATE to «is_adl_primitivetype» DATE_TIME	[Direction is 'Source -> Destination'.]

DATE_TIME

Class «is_adl_primitivetype» in package 'Data Value Types'

Represents an absolute point in time, specified to the second. Semantics defined by ISO 8601. Used for recording a precise point in real world time, and for approximate time stamps.

value constrained to: The ISO8601 date/time string.

DATE_TIME

Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 9/8/2014
Extends QUANTIFIED

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from «is_adl_primitivetype» DATE_TIME to QUANTIFIED

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

⇒ Generalization from «is_adl_primitivetype» DATE to «is_adl_primitivetype» DATE_TIME

[Direction is 'Source -> Destination'.]

⇒ Generalization from «is_adl_primitivetype» TIME to «is_adl_primitivetype» DATE_TIME

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ value : String Public

[Is static False. Containment is Not Specified.]

DURATION

Class «is_adl_primitivetype» in package 'Data Value Types'

Represents a period of time with respect to a notional point in time, which is not specified. A sign may be used to indicate the duration is backwards in time rather than forwards.

DURATION

Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 9/8/2014
Extends QUANTITY

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from «is_adl_primitivetype» DURATION to QUANTITY

[Direction is 'Source -> Destination'.]

ATTRIBUTES

duration_text : String Public
 Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)

A string representing an ISO8601 duration.

[Is static False. Containment is Not Specified.]

OPERATIONS

is_strictly_comparable_to (other :) : Public
 [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

EHR_URI

Class in package 'Data Value Types'

A URI which has the scheme name ehr, and which can only reference elements in EHRs. Used to reference elements in an EHR, which may be the current one, or another.

EHR_URI
 Version 1.0 Phase 1.0 Proposed
 Linda created on 4/22/2012. Last modified 4/10/2013
 Extends URL_VALUE

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from EHR_URI to URI_VALUE

[Direction is 'Source -> Destination'.]

ENCAPSULATED

Class in package 'Data Value Types'

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

ENCAPSULATED
 Version 1.0 Phase 1.0 Proposed
 Linda created on 4/22/2012. Last modified 5/18/2012
 Extends DATA_VALUE

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from ENCAPSULATED to DATA_VALUE

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

⇒ Generalization from PARSABLE to ENCAPSULATED

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

⇒ Generalization from MULTIMEDIA to ENCAPSULATED

[Direction is 'Source -> Destination'.]

IDENTIFIER

Class in package 'Data Value Types'

A type for representing identifiers of real-world entities. Typical identifiers include drivers licence number, social security number, veterans affairs number, prescription id, order id, and so on. IDENTIFIER is used to represent any identifier of a real thing, issued by some authority or agency. IDENTIFIER is not used to express identifiers generated by the infrastructure to refer to information items; the types OBJECT_ID and OBJECT_REF and subtypes are defined for this purpose.

IDENTIFIER

Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013
Extends DATA_VALUE

OUTGOING STRUCTURAL RELATIONSHIPS

← Generalization from IDENTIFIER to DATA_VALUE

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ id : String Public

The identifier value. Often structured, according to the definition of the issuing authority's rules.

[Is static False. Containment is Not Specified.]

◆ type : CODED_TEXT Public

The identifier type, such as prescription id, or Social Security Number.

[Is static False. Containment is Not Specified.]

◆ issuer : String Public

Authority which issues the kind of id used in the id field of this object.

[Is static False. Containment is Not Specified.]

INTERVAL_VALUE

Class in package 'Data Value Types'

Generic class defining an interval (i.e. range) of a comparable type. An interval is used to define intervals of dates, times, quantities and so on. The type parameter, T, must be a descendant of the type ORDERED, which is necessary (but not sufficient) for instances to be compared (strictly_comparable is also needed). Without the INTERVAL class, quite a few more classes would be needed to express logical intervals, namely interval versions of all the date/time classes, and of quantity classes. Further, it allows the semantics of intervals to be stated in one place unequivocally, including the conditions for strict comparison.

INTERVAL_VALUE
Version 1.0 Phase 1.0 Proposed
Linda created on 5/1/2012. Last modified 4/13/2013
Extends DATA_VALUE

OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Generalization from INTERVAL_VALUE to DATA_VALUE

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

➡ Realization «bind» from <anonymous> to INTERVAL_VALUE

[Name is <T -> DATE_TIME>. Direction is 'Source -> Destination'.]

➡ Aggregation from ORDERED_VALUE to INTERVAL_VALUE

[Direction is 'Destination -> Source'.]

➡ Generalization from INTERVAL_VALUE<T -> DATE_TIME> to INTERVAL_VALUE

[Direction is 'Source -> Destination'.]

➡ Aggregation from ORDERED_VALUE to INTERVAL_VALUE

[Direction is 'Destination -> Source'.]

ATTRIBUTES

◆ upper_unbounded : Boolean Public

[Is static False. Containment is Not Specified.]

◆ lower_unbounded : Boolean Public

[Is static False. Containment is Not Specified.]

◆ upper_included : Boolean Public

[Is static False. Containment is Not Specified.]

◆ lower_included : Boolean Public

[Is static False. Containment is Not Specified.]

MULTIMEDIA

Class in package 'Data Value Types'

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

MULTIMEDIA
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013

Extends ENCAPSULATED

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from MULTIMEDIA to ENCAPSULATED

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ alternate_text : Byte Public

Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)

Text to display in lieu of multimedia display/replay.

[Is static False. Containment is Not Specified.]

◆ data : Byte Public

Multiplicity: ([0..*], Allow duplicates: 0, Is ordered: False)

The actual data that represents the multimedia item. If the multimedia item has a uri, then this is the data found at this uri.

[Is static False. Containment is Not Specified.]

◆ media_type : CODED_TEXT Public

Data media type coded from the IANA MIME types code set.

[Is static False. Containment is .]

◆ uri : URI_VALUE Public

Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)

URI reference to electronic information stored outside the record as a file, database entry etc, if supplied as a reference.

[Is static False. Containment is .]

ORDERED_VALUE*Class in package 'Data Value Types'*

An abstract class defining the concept of ordered values, which includes ordinals as well as true quantities.

ORDERED_VALUE
 Version 1.0 Phase 1.0 Proposed
 Linda created on 4/22/2012. Last modified 9/20/2012
 Extends DATA_VALUE

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Aggregation from ORDERED_VALUE to INTERVAL_VALUE

[Direction is 'Destination -> Source'.]

↳ Aggregation from ORDERED_VALUE to INTERVAL_VALUE

[Direction is 'Destination -> Source'.]

OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Generalization from ORDERED_VALUE to DATA_VALUE

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

➡ Generalization from ORDINAL to ORDERED_VALUE

[Direction is 'Source -> Destination'.]

➡ Generalization from QUANTIFIED to ORDERED_VALUE

[Direction is 'Source -> Destination'.]

ORDINAL

Class in package 'Data Value Types'

Used to represent model rankings and scores, e.g. pain, Apgar values, 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

This class is 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 small, moderate, large}.

ORDINAL

Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013

Extends ORDERED_VALUE

OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Generalization from ORDINAL to ORDERED_VALUE

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ symbol : CODED_TEXT Public

The 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.

[Is static False. Containment is Not Specified.]

◆ value : Real Public

Value in ordered enumeration of values. Any integer value can be used.

[Is static False. Containment is Not Specified.]

PARSABLE

Class in package 'Data Value Types'

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

PARSABLE
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013
Extends ENCAPSULATED

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from PARSABLE to ENCAPSULATED

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ formalism : CODED_TEXT Public

The name of the formalism, e.g. GLIF 1.0, Proforma.

[Is static False. Containment is Not Specified.]

◆ value : String Public

The string which can be parsed according to the given formalism. The value may validly be empty in some syntaxes.

[Is static False. Containment is Not Specified.]

PLAIN_TEXT

Class in package 'Data Value Types'

A string of characters, written in a particular language, without any associated coding.

PLAIN_TEXT
Version 1.0 Phase 1.0 Proposed
Linda created on 5/12/2012. Last modified 6/6/2012
Extends TEXT

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from PLAIN_TEXT to TEXT

[Direction is 'Source -> Destination'.]

PROPORTION

Class in package 'Data Value Types'

Models a ratio of values, i.e. where the numerator and denominator are both pure numbers. 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. E.g. visual acuity 6/24 is not a ratio. Should not be used for formulations.

PROPORTION

Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013
Extends AMOUNT

OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Generalization from PROPORTION to AMOUNT

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ numerator : Real Public

The numerator of the ratio.

[Is static False. Containment is Not Specified.]

◆ denominator : Real Public

The denominator of the ratio.

[Is static False. Containment is Not Specified.]

◆ precision : Integer Public

Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)

The precision to which the numerator and denominator values of the proportion are expressed, in terms of the number of decimal places. The value 0 implies an integral quantity. The value -1 implies no limit, i.e. any number of decimal places.

[Is static False. Containment is Not Specified.]

◆ type : CODED_TEXT Public

Indicates the semantic type of the proportion. Valid values include: ratio, unitary, percent, fraction, integer_fraction

[Is static False. Containment is Not Specified.]

QUANTIFIED

Class in package 'Data Value Types'

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

QUANTIFIED

Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013
Extends ORDERED_VALUE

OUTGOING STRUCTURAL RELATIONSHIPS

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from QUANTIFIED to ORDERED_VALUE

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

⇒ Generalization from AMOUNT to QUANTIFIED

[Direction is 'Source -> Destination'.]

⇒ Generalization from «is_adl_primitivetype» DATE_TIME to QUANTIFIED

[Direction is 'Source -> Destination'.]

CONNECTORS

↗ Dependency Source -> Destination

From: QUANTIFIED : Class, Public

To: QuantifiedValueStatusEnum : Enumeration, Public

ATTRIBUTES

◆ value_status : String Public

Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)

The optional status of the magnitude with possible 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 =.

[Is static False. Containment is Not Specified.]

QUANTITY

Class in package 'Data Value Types'

A type representing a quantity, i.e. a measure which includes both a numeric magnitude and a coded units. The quantity's units may be represented in any code system, including the Unified Code for Units of Measure (UCUM), and SNOMED CT. Quantities can also be used for time durations, where the units represents a temporal measure (e.g. seconds, minutes, hours, days, months, years).

QUANTITY

Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013
Extends AMOUNT

OUTGOING STRUCTURAL RELATIONSHIPS

OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Generalization from QUANTITY to AMOUNT

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

➡ Generalization from «is_adl_primitivetype» DURATION to QUANTITY

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ value : Real Public

The numeric size of the quantity.

[Is static False. Containment is Not Specified.]

◆ units : CODED_TEXT Public

[Is static False. Containment is Not Specified.]

◆ precision : Integer Public

Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)

The precision to which the value of the quantity is expressed, as a number of significant figures.

[Is static False. Containment is Not Specified.]

QuantifiedValueStatusEnum

Enumeration «enumeration» in package 'Data Value Types'

QuantifiedValueStatusEnum
Version 1.0 Phase 1.0 Proposed
ZelM created on 10/5/2012. Last modified 3/22/2013
Extends String

CONNECTORS

↗ Dependency Source -> Destination

From: QUANTIFIED : Class, Public

To: QuantifiedValueStatusEnum : Enumeration, Public

ATTRIBUTES

◆ = : Public

[Stereotype is «enum». Is static False. Containment is Not Specified.]

◆ < : Public

[Stereotype is «enum». Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ > : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ <= : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ >= : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ ~ : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]

TERM_MAPPING

Class in package 'Data Value Types'

Represents a coded term mapped to a CODEABLE_TEXT, and the relative match of the target term with respect to the mapped item. Plain or coded text items may appear for which one or more mappings in alternative terminologies are required.

Used for adding classification terms (e.g. adding ICD classifiers to SNOMED CT descriptive terms), or mapping into equivalents in other terminologies (e.g. across nursing vocabularies).

TERM_MAPPING
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013

OUTGOING STRUCTURAL RELATIONSHIPS	
◀ Aggregation from TERM_MAPPING to TEXT	[Direction is 'Destination -> Source'.]

CONNECTORS	
➤ Dependency Source -> Destination From: TERM_MAPPING : Class, Public To: TermMappingMatchEnum : Enumeration, Public	

ATTRIBUTES	
◆ match : Character Public	<p>The relative accuracy with which the target term matches with the respective mapped text item. Valid values include:</p> <ul style="list-style-type: none"> >: the mapping is to a broader term, e.g. original text = arbovirus infection , target = viral infection =: the mapping is equivalent to the original item <: the mapping is to a narrower term, e.g. original text = diabetes , mapping = diabetes mellitus . ? : the kind of mapping is unknown. <p>The first three values are taken from the ISO standards 2788 (Guide to Establishment and development of monolingual thesauri) and 5964 (Guide to Establishment and development of multilingual thesauri).</p>

ATTRIBUTES
[Is static False. Containment is Not Specified.]
<p>purpose : CODED_TEXT Public Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)</p> <p>The purpose of the mapping e.g. automated data mining , billing, interoperability.</p> <p>[Is static False. Containment is Not Specified.]</p>

ASSOCIATIONS
<p>Association (direction: Source -> Destination)</p> <p>Source: Private (Class) TERM_MAPPING Target: Public target (Class) CODED_TEXT «is_adl_primitivetype» Cardinality: [1..1]</p>

TEXT

Class in package 'Data Value Types'

A text item, which may contain any amount of legal characters. TEXT may either be coded, codeable or plain.

A text item, which may either be a plain string value (with optional term mappings), or text that has been coded according to a given terminology.

TEXT
 Version 1.0 Phase 1.0 Proposed
 Linda created on 4/22/2012. Last modified 7/12/2013
 Extends DATA_VALUE

OUTGOING STRUCTURAL RELATIONSHIPS
<p>↳ Generalization from TEXT to DATA_VALUE</p> <p>[Direction is 'Source -> Destination'.]</p>

INCOMING STRUCTURAL RELATIONSHIPS
<p>⇒ Generalization from «is_adl_primitivetype» CODED_TEXT to TEXT</p> <p>[Direction is 'Source -> Destination'.]</p>
<p>⇒ Aggregation from TERM_MAPPING to TEXT</p> <p>[Direction is 'Destination -> Source'.]</p>
<p>⇒ Generalization from PLAIN_TEXT to TEXT</p> <p>[Direction is 'Source -> Destination'.]</p>

ATTRIBUTES

<p>◆ value : String Public [Is static False. Containment is Not Specified.]</p>
<p>◆ language : CODED_TEXT Public Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False) An optional indicator of the localised language in which the value is written. Only used when either the text object is in a different language from the enclosing ENTRY, or the text object is being used outside of an ENTRY or other enclosing structure which indicates the language. [Is static False. Containment is .]</p>

TIME

Class «is_adl_primitivetype» in package 'Data Value Types'

Represents an absolute point in time from an origin usually interpreted as meaning the start of the current day, specified to the second. Semantics defined by ISO 8601. Used for recording real world times, rather than scientifically measured fine amounts of time. The partial form is used for approximate times of events and substance administrations.

value constrained to: The ISO 8601 time string.

TIME
 Version 1.0 Phase 1.0 Proposed
 Linda created on 4/22/2012. Last modified 9/8/2014
 Extends DATE_TIME

OUTGOING STRUCTURAL RELATIONSHIPS

◀ Generalization from «is_adl_primitivetype» TIME to «is_adl_primitivetype» DATE_TIME
[Direction is 'Source -> Destination'.]

TermMappingMatchEnum

Enumeration «enumeration» in package 'Data Value Types'

TermMappingMatchEnum
 Version 1.0 Phase 1.0 Proposed
 ZelM created on 10/5/2012. Last modified 3/22/2013
 Extends Character

CONNECTORS

↗ Dependency Source -> Destination
 From: TERM_MAPPING : Class, Public
 To: TermMappingMatchEnum : Enumeration, Public

ATTRIBUTES

> : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
= : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
< : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
?: Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]

URI_VALUE

Class in package 'Data Value Types'

A data type used for referring to information resources. The URI type allows data values which are references to objects on the world wide web to be created. Its specialisation, EHR_URI, enables any element to be identified in the same way as other objects on the web. The EHR_URI type is convenient, because it is a string, like any other URI, and is therefore easily transportable and processable. Because it has its own scheme space (i.e. ehr), instances can be globally unique, as long as EHR identification is globally unique. EHR_URIs are used to express all runtime paths in the EHR.

URI_VALUE
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 7/12/2013
Extends DATA_VALUE

OUTGOING STRUCTURAL RELATIONSHIPS

Generalization from URI_VALUE to DATA_VALUE
[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

Generalization from EHR_URI to URI_VALUE
[Direction is 'Source -> Destination'.]

ATTRIBUTES

value : URI Public
[Is static False. Containment is Not Specified.]

YESNO

Class in package 'Data Value Types'

Values which represent boolean data, such as true/false or yes/no. 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. The YESNO

class should not be used as a replacement for naively modelled enumerated types such as male/female etc. Such values should be coded.

YESNO

Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013
Extends DATA_VALUE

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from YESNO to DATA_VALUE

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ value : Boolean Public

Boolean value of this item. Actual values may be language or implementation dependent.

[Is static False. Containment is Not Specified.]

Party

Package in package 'CIMI Reference Model'

The Demographics package defines the generic concepts of PARTY, ROLE and related details such as contacts and addresses. The archetype model defines the constraint semantics on PARTYs, allowing archetypes for any type of person, organisation, role and role relationship to be described. This approach provides a flexible way of including the arbitrary demographic attributes that may be required.

Party
Version 1.0 Phase 1.0 Proposed
Linda created on 5/1/2012. Last modified 4/10/2013

ACTOR

Class in package 'Party'

An ancestor of all real-world types, including people and organisations. An actor is any real-world entity capable of taking on a role.

GROUP

A group is a real world group of parties which is created by another party (usually an organisation) for some specific purpose. A typical clinical example is that of the specialist care team, e.g. cardiology team. The members of the group usually work together.

AGENT (DEVICE)

A generic concept of any kind of agent, including devices, software systems, but not humans or organisations.

PERSON

The generic description of a person. PERSON provides a dedicated type to which Person archetypes can be targeted.

ORGANISATION

A generic description of an organisation. An organisation is a legally constituted body whose existence (in general) outlives the existence of parties considered to be part of it.

ACTOR
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013
Extends PARTY

OUTGOING STRUCTURAL RELATIONSHIPS

◀ Generalization from ACTOR to PARTY

[Direction is 'Source -> Destination'.]

CONNECTORS

↗ Dependency Destination -> Source

From: ACTOR : Class, Public

To: Entity : Boundary, Public

ATTRIBUTES

♦ type : CODED_TEXT Public

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) ACTOR

Target: Public role (Class) ROLE
Cardinality: [0..*]

PARTY

Class in package 'Party'

An ancestor of all party types, including real world entities and their roles. A party is any entity which can participate in an activity. The name attribute inherited from LOCATABLE is used to indicate the actual type of the party (note that the actual names, i.e. identities of parties are indicated in the identities attribute, not the name attribute).

PARTY

Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 5/22/2014
Extends LOCATABLE

OUTGOING STRUCTURAL RELATIONSHIPS

Generalization from PARTY to «archetype_parent» LOCATABLE

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

Generalization from ROLE to PARTY

[Direction is 'Source -> Destination'.]

Generalization from ACTOR to PARTY

[Direction is 'Source -> Destination'.]

ATTRIBUTES

details : ITEM Public
Multiplicity: ([0..*], Allow duplicates: 0, Is ordered: False)

All other details for this party. These details are usually archetyped.

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public source (Class) PARTY

Target: Public relationship (Class)
PARTY_RELATIONSHIP
Cardinality: [0..*]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) PARTICIPATION

Target: Public party (Class) PARTY
Cardinality: [1..1]

Association (direction: Source -> Destination)

Source: Public (Class) PARTY_RELATIONSHIP

Target: Public target (Class) PARTY
Cardinality: [1..1]

PARTY_RELATIONSHIP

Class in package 'Party'

A generic description of a relationship between the source and target parties.

PARTY_RELATIONSHIP
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013
Extends LOCATABLE

OUTGOING STRUCTURAL RELATIONSHIPS

Generalization from PARTY_RELATIONSHIP to «archetype_parent» LOCATABLE

[Direction is 'Source -> Destination'.]

CONNECTORS

Dependency Destination -> Source

From: PARTY_RELATIONSHIP : Class, Public
To: RoleLink : Boundary, Public

ATTRIBUTES

type : CODED_TEXT Public

The detailed description of the relationship

[Is static False. Containment is Not Specified.]

details : ITEM Public

Multiplicity: ([0..*], Allow duplicates: 0, Is ordered: False)

The detailed description of the relationship.

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

ASSOCIATIONS	
Source: Public (Class) PARTY_RELATIONSHIP	Target: Public target (Class) PARTY Cardinality: [1..1]
Association (direction: Source -> Destination)	Source: Public source (Class) PARTY Target: Public relationship (Class) PARTY_RELATIONSHIP Cardinality: [0..*]

ROLE

Class in package 'Party'

A generic description of a role performed by an actor. The role corresponds to a competency of the party. Roles are used to define the responsibilities undertaken by a party for a purpose. Roles should have credentials qualifying the performer to perform the role.

ROLE
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013
Extends PARTY

OUTGOING STRUCTURAL RELATIONSHIPS	
Generalization from ROLE to PARTY	[Direction is 'Source -> Destination'.]

CONNECTORS	
Dependency Destination -> Source From: ROLE : Class, Public To: Role : Boundary, Public	

ATTRIBUTES	
type : CODED_TEXT Public	[Is static False. Containment is Not Specified.]

ASSOCIATIONS	
Association (direction: Source -> Destination)	Source: Public (Class) ACTOR Target: Public role (Class) ROLE Cardinality: [0..*]

Primitive Types

Package «primitive_types» in package 'CIMI Reference Model'

The Primitive Types package describes the basic types that are assumed in external type systems; this package is a guide for integrating models into the type systems of implementation technologies.

Primitive Types
Version 1.0 Phase 1.0 Proposed
ZelM created on 9/14/2012. Last modified 9/17/2014

Any

Class in package 'Primitive Types'

Any
Version 1.0 Phase 1.0 Proposed
Michael van der Zel created on 9/17/2014. Last modified 9/17/2014

Array

Class in package 'Primitive Types'

Array
Version 1.0 Phase 1.0 Proposed
Linda created on 9/17/2014. Last modified 9/17/2014

List

Class in package 'Primitive Types'

List
Version 1.0 Phase 1.0 Proposed
Linda created on 9/17/2014. Last modified 9/17/2014

Boolean

DataType in package 'Primitive Types'

Boolean
Version 1.0 Phase 1.0 Proposed
ZelM created on 5/30/2013. Last modified 9/17/2014

Byte

DataType in package 'Primitive Types'

Byte
Version 1.0 Phase 1.0 Proposed
ZelM created on 5/30/2013. Last modified 9/17/2014

Character

DataType in package 'Primitive Types'

Character
Version 1.0 Phase 1.0 Proposed
ZelM created on 5/30/2013. Last modified 9/17/2014

Integer

DataType in package 'Primitive Types'

Integer
Version 1.0 Phase 1.0 Proposed
ZelM created on 5/30/2013. Last modified 9/17/2014

Real

DataType in package 'Primitive Types'

Real
Version 1.0 Phase 1.0 Proposed
ZelM created on 5/30/2013. Last modified 9/17/2014

String

DataType in package 'Primitive Types'

String
Version 1.0 Phase 1.0 Proposed
ZelM created on 5/30/2013. Last modified 9/17/2014

URI

DataType in package 'Primitive Types'

URI
Version 1.0 Phase 1.0 Proposed
Linda created on 7/12/2013. Last modified 9/17/2014

Legend

Package in package 'Main'

Legend
Version 1.0 Phase 1.0 Proposed
ZelM created on 6/1/2012. Last modified 6/1/2012

Abstract

Class in package 'Legend'

Abstract
Version 1.0 Phase 1.0 Proposed
Linda created on 6/1/2012. Last modified 6/1/2012

Datatype

Class in package 'Legend'

Datatype
Version 1.0 Phase 1.0 Proposed
Linda created on 6/1/2012. Last modified 6/1/2012

Demographics Entry Point

Class in package 'Legend'

Demographics Entry Point
Version 1.0 Phase 1.0 Proposed
Linda created on 6/1/2012. Last modified 2/7/2013

Entry Point

Class in package 'Legend'

Entry Point
Version 1.0 Phase 1.0 Proposed
Linda created on 6/1/2012. Last modified 6/1/2012

Supporting

Class in package 'Legend'

Supporting
Version 1.0 Phase 1.0 Proposed
Linda created on 6/1/2012. Last modified 6/1/2012

Primitive Types

DataType in package 'Legend'

= Assumed Types

Primitive Types
Version 1.0 Phase 1.0 Proposed
Linda created on 6/1/2012. Last modified 12/2/2012

Mapping CIMI RM 2 HL7 RIM

Package in package 'Main'

Mapping CIMI RM 2 HL7 RIM
Version 1.0 Phase 1.0 Proposed
ZelM created on 3/24/2013. Last modified 3/24/2013

HL7 Entity-Role-Participation-Act diagram

Class diagram in package 'Mapping CIMI RM 2 HL7 RIM'

This diagram shows the mapping between the CIMI RM and the HL7 Backbone

HL7 Entity-Role-Participation-Act
Version 1.0
Linda created on 1/20/2013. Last modified 4/20/2013

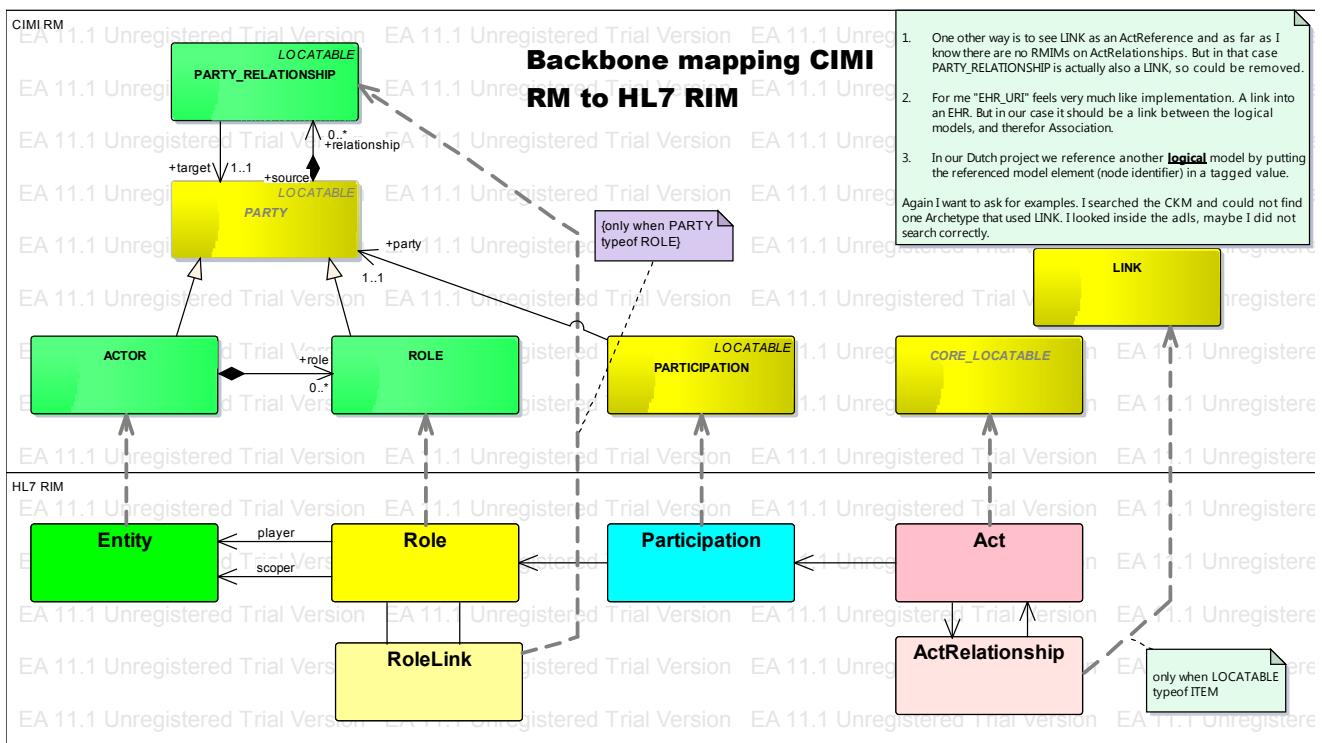


Figure 7: HL7 Entity-Role-Participation-Act

CORE_LOCATABLE

Class in package 'Core'

CORE_LOCATABLE
Version 1.0 Phase 1.0 Proposed
ZelM created on 1/20/2013. Last modified 5/23/2013



LINK

Class in package 'Core'

The LINK type defines a logical relationship between two items, such as two ENTRYS or an ENTRY and a COMPOSITION. Links can be used across compositions, and across EHRs. Links can potentially be used between interior (i.e. non archetype root) nodes, although this probably should be prevented in archetypes. Multiple LINKs can be attached to the root object of any archetyped structure to give the effect of a 1->N link. 1:1 and 1:N relationships between archetyped content elements (e.g. ENTRYS) can be expressed by using one, or more than one, respectively, LINKs. Chains of links can be used to see problem threads or other logical groupings of items. Links should be used between archetyped structures only, i.e. between objects representing complete domain concepts, because relationships between sub-elements of whole concepts are not necessarily meaningful. Sensible */*links only exist between whole ITEM_GROUPS and/or PARTICIPATIONS.

LINK

Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 12/18/2014

CONNECTORS

 **Dependency** Destination -> Source

From: LINK : Class, Public

To: ActRelationship : Boundary, Public

ATTRIBUTES

 meaning : TEXT Public

Used to describe the relationship between the source and the target of the link (usually in clinical terms) - such as the relationship between test results and an order, follow-up to a consultation and so on. The meaning of each link falls under one of the following categories: generic, documenting and reporting, organisational, clinical , circumstantial , and view management .

[Is static False. Containment is .]

 details : ITEM Public

Multiplicity: ([0..*], Allow duplicates: 0, Is ordered: False)

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

 Association (direction: Source -> Destination)

Target can be archetype_id or constraint (business rules/query language) on instance.

Target is the constraint on "type of instance" referenced at runtime.

target LOCATABLE "type of instance" is constrained in the AOM. Just like terminology binding.

Target is an association to a LOCATABLE.

Source: Public (Class) LINK

Target: Public target (Class) ITEM_GROUP
Cardinality: [1..1]

 Association (direction: Source -> Destination)

ASSOCIATIONS

Source: Public source (Class) ITEM_GROUP

Target: Public link (Class) LINK

Cardinality: [0..*]

PARTICIPATION*Class in package 'Core'*

Used to represent any participation of a Party (e.g. any Actor or Role) in some activity, 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.

PARTICIPATION
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 5/22/2014
Extends LOCATABLE

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from PARTICIPATION to «archetype_parent» LOCATABLE

[Direction is 'Source -> Destination'.]

↳ Aggregation from PARTICIPATION to ITEM_GROUP

[Direction is 'Destination -> Source'.]

CONNECTORS

↗ **Dependency** Destination -> Source
 From: PARTICIPATION : Class, Public
 To: Participation : Boundary, Public

ATTRIBUTES

◆ **function** : CODED_TEXT Public

The function of the Party in this participation. Please note that a given party might participate in more than one way in a particular activity.

[Is static False. Containment is Not Specified.]

◆ **details** : ITEM Public
 Multiplicity: ([0..*], Allow duplicates: 0, Is ordered: False)

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

↙ Association (direction: Source -> Destination)

Source: Public (Class) PARTICIPATION

Target: Public party (Class) PARTY

Cardinality: [1..1]

ACTOR

Class in package 'Party'

An ancestor of all real-world types, including people and organisations. An actor is any real-world entity capable of taking on a role.

GROUP

A group is a real world group of parties which is created by another party (usually an organisation) for some specific purpose. A typical clinical example is that of the specialist care team, e.g. cardiology team. The members of the group usually work together.

AGENT (DEVICE)

A generic concept of any kind of agent, including devices, software systems, but not humans or organisations.

PERSON

The generic description of a person. PERSON provides a dedicated type to which Person archetypes can be targeted.

ORGANISATION

A generic description of an organisation. An organisation is a legally constituted body whose existence (in general) outlives the existence of parties considered to be part of it.

ACTOR

Version 1.0 Phase 1.0 Proposed

Linda created on 4/22/2012. Last modified 4/10/2013

Extends PARTY

OUTGOING STRUCTURAL RELATIONSHIPS

◀ Generalization from ACTOR to PARTY

[Direction is 'Source -> Destination'.]

CONNECTORS

↗ Dependency Destination -> Source

From: ACTOR : Class, Public

To: Entity : Boundary, Public

ATTRIBUTES

◆ type : CODED_TEXT Public

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

↙ Association (direction: Source -> Destination)

Source: Public (Class) ACTOR

Target: Public role (Class) ROLE

Cardinality: [0..*]

PARTY

Class in package 'Party'

An ancestor of all party types, including real world entities and their roles. A party is any entity which can participate in an activity. The name attribute inherited from LOCATABLE is used to indicate the actual type of the party (note that the actual names, i.e. identities of parties are indicated in the identities attribute, not the name attribute).

PARTY
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 5/22/2014
Extends LOCATABLE

OUTGOING STRUCTURAL RELATIONSHIPS

◀ Generalization from PARTY to «archetype_parent» LOCATABLE

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

⇒ Generalization from ROLE to PARTY

[Direction is 'Source -> Destination'.]

⇒ Generalization from ACTOR to PARTY

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ details : ITEM Public

Multiplicity: ([0..*], Allow duplicates: 0, Is ordered: False)

All other details for this party. These details are usually archetyped.

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

✍ Association (direction: Source -> Destination)

Source: Public source (Class) PARTY

Target: Public relationship (Class)

PARTY_RELATIONSHIP

Cardinality: [0..*]

✍ Association (direction: Source -> Destination)

Source: Public (Class) PARTICIPATION

Target: Public party (Class) PARTY

Cardinality: [1..1]

✍ Association (direction: Source -> Destination)

Source: Public (Class) PARTY_RELATIONSHIP

Target: Public target (Class) PARTY

Cardinality: [1..1]

PARTY_RELATIONSHIP

Class in package 'Party'

A generic description of a relationship between the source and target parties.

PARTY_RELATIONSHIP
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013
Extends LOCATABLE

OUTGOING STRUCTURAL RELATIONSHIPS

◀ Generalization from PARTY_RELATIONSHIP to «archetype_parent» LOCATABLE

[Direction is 'Source -> Destination'.]

CONNECTORS

↗ Dependency Destination -> Source
 From: PARTY_RELATIONSHIP : Class, Public
 To: RoleLink : Boundary, Public

ATTRIBUTES

◆ type : CODED_TEXT Public

The detailed description of the relationship

[Is static False. Containment is Not Specified.]

◆ details : ITEM Public
 Multiplicity: ([0..*], Allow duplicates: 0, Is ordered: False)

The detailed description of the relationship.

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

↙ Association (direction: Source -> Destination)

Source: Public (Class) PARTY_RELATIONSHIP

Target: Public target (Class) PARTY
 Cardinality: [1..1]

↙ Association (direction: Source -> Destination)

Source: Public source (Class) PARTY

Target: Public relationship (Class)
 PARTY_RELATIONSHIP
 Cardinality: [0..*]

ROLE

Class in package 'Party'

A generic description of a role performed by an actor. The role corresponds to a competency of the party. Roles are used

to define the responsibilities undertaken by a party for a purpose. Roles should have credentials qualifying the performer to perform the role.

ROLE

Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 4/10/2013
Extends PARTY

OUTGOING STRUCTURAL RELATIONSHIPS

 Generalization from ROLE to PARTY

[Direction is 'Source -> Destination'.]

CONNECTORS

 Dependency Destination -> Source

From: ROLE : Class, Public
To: Role : Boundary, Public

ATTRIBUTES

 type : CODED_TEXT Public

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

 Association (direction: Source -> Destination)

Source: Public (Class) ACTOR

Target: Public role (Class) ROLE
Cardinality: [0..*]

Act

Boundary in package 'Mapping CIMI RM 2 HL7 RIM'

Act

Version 1.0 Phase 1.0 Proposed
ZelM created on 1/20/2013. Last modified 1/20/2013
Extends

CONNECTORS

 Dependency Destination -> Source

From: CORE_LOCATABLE : Class, Public
To: Act : Boundary, Public

ActRelationship

Boundary in package 'Mapping CIMI RM 2 HL7 RIM'

ActRelationship

Version 1.0 Phase 1.0 Proposed
 ZelM created on 2/7/2013. Last modified 2/7/2013
 Extends

CONNECTORS

 **Dependency** Destination -> Source

From: LINK : Class, Public
 To: ActRelationship : Boundary, Public

Entity

Boundary in package 'Mapping CIMI RM 2 HL7 RIM'

Entity
 Version 1.0 Phase 1.0 Proposed
 ZelM created on 1/20/2013. Last modified 1/20/2013
 Extends

CONNECTORS

 **Dependency** Destination -> Source

From: ACTOR : Class, Public
 To: Entity : Boundary, Public

Participation

Boundary in package 'Mapping CIMI RM 2 HL7 RIM'

Participation
 Version 1.0 Phase 1.0 Proposed
 ZelM created on 1/20/2013. Last modified 1/20/2013
 Extends

CONNECTORS

 **Dependency** Destination -> Source

From: PARTICIPATION : Class, Public
 To: Participation : Boundary, Public

Role

Boundary in package 'Mapping CIMI RM 2 HL7 RIM'

Role
 Version 1.0 Phase 1.0 Proposed
 ZelM created on 1/20/2013. Last modified 1/20/2013
 Extends

CONNECTORS

CONNECTORS

 **Dependency** Destination -> Source

From: ROLE : Class, Public

To: Role : Boundary, Public

RoleLink

Boundary in package 'Mapping CIMI RM 2 HL7 RIM'

RoleLink

Version 1.0 Phase 1.0 Proposed

ZelM created on 1/20/2013. Last modified 2/7/2013

Extends

CONNECTORS

 **Dependency** Destination -> Source

From: PARTY_RELATIONSHIP : Class, Public

To: RoleLink : Boundary, Public

<anonymous>

Constraint in package 'Mapping CIMI RM 2 HL7 RIM'

only when PARTY typeof ROLE

<anonymous>

Version 1.0 Phase 1.0 Proposed

zelm created on 3/22/2013. Last modified 4/9/2013

Extends

<anonymous>

Note in package 'Mapping CIMI RM 2 HL7 RIM'

only when LOCATABLE typeof ITEM

<anonymous>

Version 1.0 Phase 1.0 Proposed

ZelM created on 4/20/2013. Last modified 4/20/2013

Extends

<anonymous>

Note in package 'Mapping CIMI RM 2 HL7 RIM'

1. One other way is to see LINK as an ActReference and as far as I know there are no RMIMs on ActRelationships. But in that case PARTY_RELATIONSHIP is actually also a LINK, so could be removed.
2. For me "EHR_URI" feels very much like implementation. A link into an EHR. But in our case it should be a link between the logical models, and therefore Association.

3. In our Dutch project we reference another **logical** model by putting the referenced model element (node identifier) in a tagged value.

Again I want to ask for examples. I searched the CKM and could not find one Archetype that used LINK. I looked inside the adls, maybe I did not search correctly.

<anonymous>
Version 1.0 Phase 1.0 Proposed
ZelM created on 4/20/2013. Last modified 4/20/2013
Extends

<anonymous>

Text in package 'Mapping CIMI RM 2 HL7 RIM'

Backbone mapping CIMI RM to HL7 RIM

<anonymous>
Version 1.0 Phase 1.0 Proposed
ZelM created on 1/20/2013. Last modified 3/22/2013
Extends

Mapping DCM 2 CIMI RM

Package in package 'Main'

Mapping DCM 2 CIMI RM
Version 1.0 Phase 1.0 Proposed
ZelM created on 9/14/2012. Last modified 1/19/2013

Mapping DCM 2 CIMI RM diagram

CompositeStructure diagram in package 'Mapping DCM 2 CIMI RM'

Mapping DCM 2 CIMI RM
Version 1.0
ZelM created on 6/5/2012. Last modified 6/3/2014

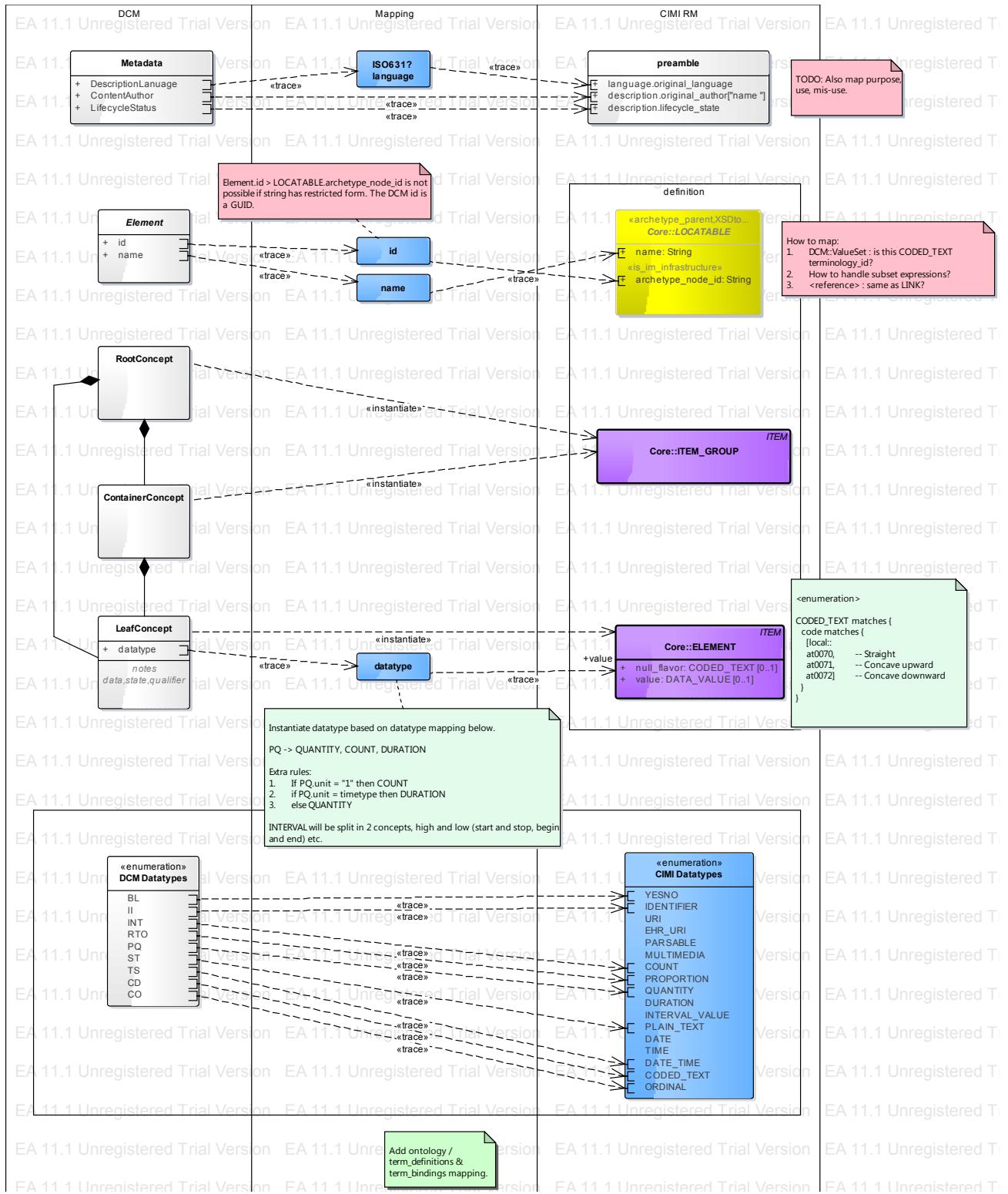


Figure 8: Mapping DCM 2 CIMI RM

ELEMENT

Class in package 'Core'

A type of data ITEM, which does not itself contain ITEMS.

ELEMENT
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 5/22/2014
Extends ITEM

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Generalization from ELEMENT to ITEM	[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Generalization from name to ELEMENT	[Direction is 'Source -> Destination'.]
⇒ Generalization from status to ELEMENT	[Direction is 'Source -> Destination'.]
⇒ Generalization from datetime range to ELEMENT	[Direction is 'Source -> Destination'.]

CONNECTORS	
↗ Trace «trace» Source -> Destination From: datatype : Property, Public To: ELEMENT : Class, Public value	
↗ Dependency «instantiate» Source -> Destination From: LeafConcept : Class, Public To: ELEMENT : Class, Public	

ATTRIBUTES	
◆ null_flavor : CODED_TEXT Public Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)	[Is static False. Containment is Not Specified.]
◆ value : DATA_VALUE Public Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)	[Is static False. Containment is Not Specified.]

ITEM_GROUP

Class in package 'Core'

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

ITEM_GROUP
Version 1.0 Phase 1.0 Proposed
Linda created on 4/22/2012. Last modified 12/18/2014
Extends ITEM

OUTGOING STRUCTURAL RELATIONSHIPS	
-----------------------------------	--

OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Generalization from ITEM_GROUP to ITEM

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

➡ Generalization from observable to ITEM_GROUP

[Direction is 'Source -> Destination'.]

➡ Generalization from result to ITEM_GROUP

[Direction is 'Source -> Destination'.]

➡ Generalization from timing to ITEM_GROUP

[Direction is 'Source -> Destination'.]

➡ Aggregation from PARTICIPATION to ITEM_GROUP

[Direction is 'Destination -> Source'.]

CONNECTORS

↗ Dependency «instantiate» Source -> Destination

From: RootConcept : Class, Public

To: ITEM_GROUP : Class, Public

↗ Dependency «instantiate» Source -> Destination

From: ContainerConcept : Class, Public

To: ITEM_GROUP : Class, Public

ASSOCIATIONS

↙ Association (direction: Source -> Destination)

Source: Public source (Class) ITEM_GROUP

Target: Public link (Class) LINK

Cardinality: [0..*]

↙ Association (direction: Source -> Destination)

Source: Private (Class) ITEM_GROUP

Target: Public item (Class) ITEM

Cardinality: [1..*]

↙ Association (direction: Source -> Destination)

Target can be archetype_id or constraint (business rules/query language) on instance.

Target is the constraint on "type of instance" referenced at runtime.

target LOCATABLE "type of instance" is constrained in the AOM. Just like terminology binding.

Target is an association to a LOCATABLE.

ASSOCIATIONS

Source: Public (Class) LINK

Target: Public target (Class) ITEM_GROUP

Cardinality: [1..1]

LOCATABLE*Class «archetype_parent» in package 'Core'*

The root class of all information model classes that can be archetyped. Most classes in the CIMI reference model inherit from the LOCATABLE class, which defines the idea of locatability in an archetyped structure. LOCATABLE defines a runtime name and an archetype_node_id.

LOCATABLE

Version 1.0 Phase 1.0 Proposed

Linda created on 4/22/2012. Last modified 5/22/2014

INCOMING STRUCTURAL RELATIONSHIPS

⇒ Generalization from PARTICIPATION to «archetype_parent» LOCATABLE

[Direction is 'Source -> Destination'.]

⇒ Generalization from PARTY to «archetype_parent» LOCATABLE

[Direction is 'Source -> Destination'.]

⇒ Generalization from ITEM to «archetype_parent» LOCATABLE

[Direction is 'Source -> Destination'.]

⇒ Generalization from PARTY_RELATIONSHIP to «archetype_parent» LOCATABLE

[Direction is 'Source -> Destination'.]

CONNECTORS

↗ Trace «trace» Source -> Destination

From: id : Property, Public

To: LOCATABLE : Class, Public

↗ Trace «trace» Source -> Destination

From: name : Property, Public

To: LOCATABLE : Class, Public

ATTRIBUTES

ATTRIBUTES	
◆ archetype_node_id : String Public	The design-time archetype id of this node taken from its generating archetype. This archetype node id is used to build archetype paths. It is always in the form of an at_code, e.g. at0005. This value enables a "standardised" name for this node to be generated, by referring to the generating archetype local ontology. At an archetype root point, the value of this attribute is always the stringified form of the archetype_id found in the archetype_details object.
[Stereotype is «is_im_infrastructure». Is static False. Containment is Not Specified.]	
◆ name : String Public	The runtime name of this fragment, used to build runtime paths. This is the term provided via a clinical application or batch process to name this EHR construct. Its retention in the EHR faithfully preserves the original label by which this entry was known to end users.
[Is static False. Containment is Not Specified.]	

ASSOCIATIONS	
✍ Association (direction: Source -> Destination)	Source: Private (Class) LOCATABLE «archetype_parent» Target: Public «is_im_infrastructure» archetype_details (Class) ARCHETYPED Cardinality: [0..1]

definition

Boundary in package 'Mapping DCM 2 CIMI RM'

definition
Version 1.0 Phase 1.0 Proposed
ZelM created on 9/16/2012. Last modified 9/16/2012
Extends

<anonymous>

Note in package 'Mapping DCM 2 CIMI RM'

TODO: Also map purpose, use, mis-use.

<anonymous>
Version 1.0 Phase 1.0 Proposed
ZelM created on 9/16/2012. Last modified 9/16/2012
Extends

<anonymous>

Note in package 'Mapping DCM 2 CIMI RM'

Add ontology / term_definitions & term_bindings mapping.

<anonymous>

Version 1.0 Phase 1.0 Proposed
 ZelM created on 9/16/2012. Last modified 9/16/2012
 Extends

Datatype mapping

Boundary in package 'Mapping DCM 2 CIMI RM'

Datatype mapping
 Version 1.0 Phase 1.0 Proposed
 ZelM created on 6/5/2012. Last modified 6/5/2012
 Extends

<anonymous>

Note in package 'Mapping DCM 2 CIMI RM'

Element.id > LOCATABLE.archetype_node_id is not possible if string has restricted form. The DCM id is a GUID.

<anonymous>
 Version 1.0 Phase 1.0 Proposed
 ZelM created on 6/5/2012. Last modified 6/5/2012
 Extends

<anonymous>

Note in package 'Mapping DCM 2 CIMI RM'

Instantiate datatype based on datatype mapping below.

PQ -> QUANTITY, COUNT, DURATION

Extra rules:

1. If PQ.unit = "1" then COUNT
2. if PQ.unit = timetype then DURATION
3. else QUANTITY

INTERVAL will be split in 2 concepts, high and low (start and stop, begin and end) etc.

<anonymous>
 Version 1.0 Phase 1.0 Proposed
 ZelM created on 6/5/2012. Last modified 3/22/2013
 Extends

<anonymous>

Note in package 'Mapping DCM 2 CIMI RM'

How to map:

1. DCM::ValueSet : is this CODED_TEXT terminology_id?
2. How to handle subset expressions?
3. <reference> : same as LINK?

<anonymous>

<anonymous>

Note in package 'Mapping DCM 2 CIMI RM'

<enumeration>

```
CODED_TEXT matches {
  code matches {
    [local:::
      at0070, -- Straight
      at0071, -- Concave upward
      at0072] -- Concave downward
    }
}
```

<anonymous>

Version 1.0 Phase 1.0 Proposed
 ZelM created on 6/5/2012. Last modified 4/11/2013
 Extends

CIMI Datatypes

Enumeration «enumeration» in package 'Mapping DCM 2 CIMI RM'

CIMI Datatypes

Version 1.0 Phase 1.0 Proposed
 ZelM created on 6/5/2012. Last modified 6/5/2012

CONNECTORS

↗ Trace «trace» Source -> Destination

From: DCM Datatypes : Enumeration, Public
 To: CIMI Datatypes : Enumeration, Public

↗ Trace «trace» Source -> Destination

From: DCM Datatypes : Enumeration, Public
 To: CIMI Datatypes : Enumeration, Public

↗ Trace «trace» Source -> Destination

From: DCM Datatypes : Enumeration, Public
 To: CIMI Datatypes : Enumeration, Public

↗ Trace «trace» Source -> Destination

From: DCM Datatypes : Enumeration, Public
 To: CIMI Datatypes : Enumeration, Public

CONNECTORS	
 Trace «trace» Source -> Destination	
From:	DCM Datatypes : Enumeration, Public
To:	CIMI Datatypes : Enumeration, Public
 Trace «trace» Source -> Destination	
From:	DCM Datatypes : Enumeration, Public
To:	CIMI Datatypes : Enumeration, Public
 Trace «trace» Source -> Destination	
From:	DCM Datatypes : Enumeration, Public
To:	CIMI Datatypes : Enumeration, Public
 Trace «trace» Source -> Destination	
From:	DCM Datatypes : Enumeration, Public
To:	CIMI Datatypes : Enumeration, Public
 Trace «trace» Source -> Destination	
From:	DCM Datatypes : Enumeration, Public
To:	CIMI Datatypes : Enumeration, Public
ATTRIBUTES	
 YESNO : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
 IDENTIFIER : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
 URI : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
 EHR_URI : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
 PARSABLE : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
 MULTIMEDIA : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
 COUNT : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
 PROPORTION : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]

ATTRIBUTES	
◆ QUANTITY : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ DURATION : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ INTERVAL_VALUE : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ PLAIN_TEXT : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ DATE : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ TIME : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ DATE_TIME : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ CODED_TEXT : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]
◆ ORDINAL : Public	[Stereotype is «enum». Is static False. Containment is Not Specified.]

ContainerConcept

Class in package 'Mapping DCM 2 CIMI RM'

ContainerConcept
Version 1.0 Phase 1.0 Proposed
ZelM created on 6/5/2012. Last modified 6/5/2012

OUTGOING STRUCTURAL RELATIONSHIPS	
◀ Aggregation from ContainerConcept to RootConcept	[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS	
▶ Aggregation from LeafConcept to ContainerConcept	[Direction is 'Source -> Destination'.]

CONNECTORS

 **Dependency** «instantiate» Source -> Destination

From: ContainerConcept : Class, Public
 To: ITEM_GROUP : Class, Public

DCM Datatypes

Enumeration «enumeration» in package 'Mapping DCM 2 CIMI RM'

DCM Datatypes
 Version 1.0 Phase 1.0 Proposed
 ZelM created on 6/5/2012. Last modified 4/11/2013

CONNECTORS

 **Trace** «trace» Source -> Destination

From: DCM Datatypes : Enumeration, Public
 To: CIMI Datatypes : Enumeration, Public

 **Trace** «trace» Source -> Destination

From: DCM Datatypes : Enumeration, Public
 To: CIMI Datatypes : Enumeration, Public

 **Trace** «trace» Source -> Destination

From: DCM Datatypes : Enumeration, Public
 To: CIMI Datatypes : Enumeration, Public

 **Trace** «trace» Source -> Destination

From: DCM Datatypes : Enumeration, Public
 To: CIMI Datatypes : Enumeration, Public

 **Trace** «trace» Source -> Destination

From: DCM Datatypes : Enumeration, Public
 To: CIMI Datatypes : Enumeration, Public

 **Trace** «trace» Source -> Destination

From: DCM Datatypes : Enumeration, Public
 To: CIMI Datatypes : Enumeration, Public

 **Trace** «trace» Source -> Destination

From: DCM Datatypes : Enumeration, Public
 To: CIMI Datatypes : Enumeration, Public

 **Trace** «trace» Source -> Destination

From: DCM Datatypes : Enumeration, Public
 To: CIMI Datatypes : Enumeration, Public

CONNECTORS	
 Trace «trace»	Source -> Destination
From: DCM Datatypes : Enumeration, Public	
To:	CIMI Datatypes : Enumeration, Public
ATTRIBUTES	
 BL	: Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
 II	: Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
 INT	: Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
 RTO	: Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
 PQ	: Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
 ST	: Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
 TS	: Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
 CD	: Public [Stereotype is «enum». Is static False. Containment is Not Specified.]
 CO	: Public [Stereotype is «enum». Is static False. Containment is Not Specified.]

Element

Class in package 'Mapping DCM 2 CIMI RM'

Element
Version 1.0 Phase 1.0 Proposed
ZelM created on 6/5/2012. Last modified 6/5/2012

CONNECTORS	

CONNECTORS

 **Trace** «trace» Source -> Destination

From: Element : Class, Public

To: name : Property, Public

 **Trace** «trace» Source -> Destination

From: Element : Class, Public

To: id : Property, Public

ATTRIBUTES

 id : Public

[Is static False. Containment is Not Specified.]

 name : Public

[Is static False. Containment is Not Specified.]

LeafConcept

Class in package 'Mapping DCM 2 CIMI RM'

data,state,qualifier

LeafConcept

Version 1.0 Phase 1.0 Proposed
ZelM created on 6/5/2012. Last modified 6/5/2012

OUTGOING STRUCTURAL RELATIONSHIPS

 Aggregation from LeafConcept to RootConcept

[Direction is 'Source -> Destination'.]

 Aggregation from LeafConcept to ContainerConcept

[Direction is 'Source -> Destination'.]

CONNECTORS

 **Dependency** «instantiate» Source -> Destination

From: LeafConcept : Class, Public

To: ELEMENT : Class, Public

 **Trace** «trace» Source -> Destination

From: LeafConcept : Class, Public

To: datatype : Property, Public

ATTRIBUTES

◊ datatype : Public

[Is static False. Containment is Not Specified.]

Metadata

Class in package 'Mapping DCM 2 CIMI RM'

Metadata

Version 1.0 Phase 1.0 Proposed
ZelM created on 6/5/2012. Last modified 6/5/2012

CONNECTORS

↗ Trace «trace» Source -> Destination

From: Metadata : Class, Public

To: ISO631? language : Property, Public

↗ Trace «trace» Source -> Destination

From: Metadata : Class, Public

To: preamble : Class, Public

↗ Trace «trace» Source -> Destination

From: Metadata : Class, Public

To: preamble : Class, Public

ATTRIBUTES

◊ DescriptionLanuage : Public

[Is static False. Containment is Not Specified.]

◊ ContentAuthor : Public

[Is static False. Containment is Not Specified.]

◊ LifecycleStatus : Public

[Is static False. Containment is Not Specified.]

RootConcept

Class in package 'Mapping DCM 2 CIMI RM'

RootConcept

Version 1.0 Phase 1.0 Proposed
ZelM created on 6/5/2012. Last modified 6/5/2012

INCOMING STRUCTURAL RELATIONSHIPS

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Aggregation from LeafConcept to RootConcept	[Direction is 'Source -> Destination'.]
⇒ Aggregation from ContainerConcept to RootConcept	[Direction is 'Source -> Destination'.]

CONNECTORS	
<p>↗ Dependency «instantiate» Source -> Destination From: RootConcept : Class, Public To: ITEM_GROUP : Class, Public</p>	

preamble

Class in package 'Mapping DCM 2 CIMI RM'

preamble
Version 1.0 Phase 1.0 Proposed
ZelM created on 9/16/2012. Last modified 9/16/2012

CONNECTORS	
<p>↗ Trace «trace» Source -> Destination From: ISO631? language : Property, Public To: preamble : Class, Public</p>	
<p>↗ Trace «trace» Source -> Destination From: Metadata : Class, Public To: preamble : Class, Public</p>	
<p>↗ Trace «trace» Source -> Destination From: Metadata : Class, Public To: preamble : Class, Public</p>	

ATTRIBUTES	
◆ language.original_language : Public	[Is static False. Containment is Not Specified.]
◆ description.original_author["name "] : Public	[Is static False. Containment is Not Specified.]
◆ description.lifecycle_state : Public	[Is static False. Containment is Not Specified.]

ISO631? language

Property in package 'Mapping DCM 2 CIMI RM'

ISO631? language
Version 1.0 Phase 1.0 Proposed
ZelM created on 6/5/2012. Last modified 6/5/2012

CONNECTORS

↗ **Trace** «trace» Source -> Destination
 From: ISO631? language : Property, Public
 To: preamble : Class, Public

↗ **Trace** «trace» Source -> Destination
 From: Metadata : Class, Public
 To: ISO631? language : Property, Public

datatype

Property in package 'Mapping DCM 2 CIMI RM'

datatype
Version 1.0 Phase 1.0 Proposed
ZelM created on 6/5/2012. Last modified 6/5/2012

CONNECTORS

↗ **Trace** «trace» Source -> Destination
 From: datatype : Property, Public
 To: ELEMENT : Class, Public value

↗ **Trace** «trace» Source -> Destination
 From: LeafConcept : Class, Public
 To: datatype : Property, Public

id

Property in package 'Mapping DCM 2 CIMI RM'

id
Version 1.0 Phase 1.0 Proposed
ZelM created on 6/5/2012. Last modified 6/5/2012

CONNECTORS

↗ **Trace** «trace» Source -> Destination
 From: id : Property, Public
 To: LOCATABLE : Class, Public

CONNECTORS

 **Trace** «trace» Source -> Destination

From: Element : Class, Public

To: id : Property, Public

name

Property in package 'Mapping DCM 2 CIMI RM'

name

Version 1.0 Phase 1.0 Proposed

ZelM created on 6/5/2012. Last modified 6/5/2012

CONNECTORS

 **Trace** «trace» Source -> Destination

From: name : Property, Public

To: LOCATABLE : Class, Public

 **Trace** «trace» Source -> Destination

From: Element : Class, Public

To: name : Property, Public

Model Transformation Map

Package in package 'Main'

Model Transformation Map
Version 1.0 Phase 1.0 Proposed
ZelM created on 9/15/2012. Last modified 9/15/2012

Model Transformation Map diagram

Data Flow diagram in package 'Model Transformation Map'

Model Transformation Map
Version 1.0
ZelM created on 9/15/2012. Last modified 3/22/2013

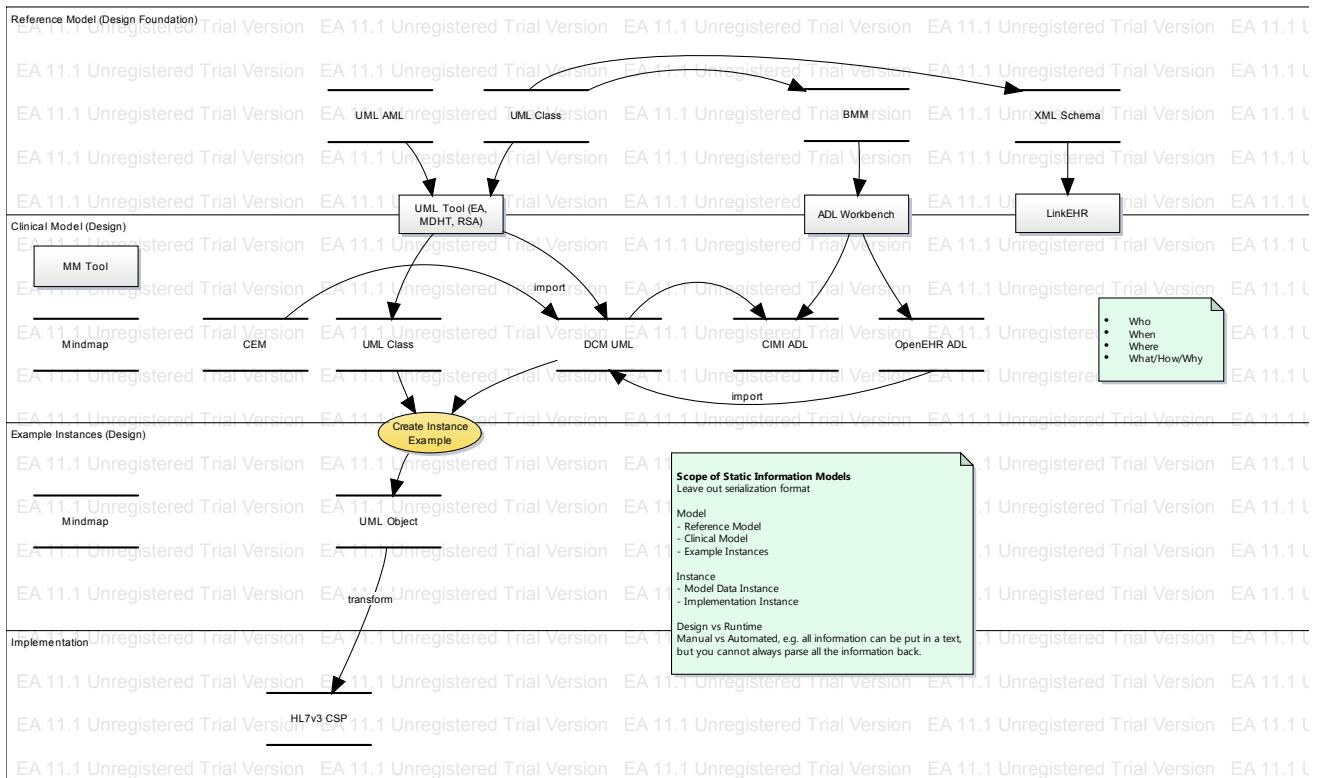


Figure 9: Model Transformation Map

<anonymous>

Note in package 'Model Transformation Map'

Scope of Static Information Models

Leave out serialization format

Model

- Reference Model
- Clinical Model
- Example Instances

Instance

- Model Data Instance
- Implementation Instance

Design vs Runtime

Manual vs Automated, e.g. all information can be put in a text, but you cannot always parse all the information back.

<anonymous>
Version 1.0 Phase 1.0 Proposed
ZelM created on 9/15/2012. Last modified 9/16/2012
Extends

<anonymous>

Note in package 'Model Transformation Map'

- Who
- When
- Where
- What/How/Why

<anonymous>
Version 1.0 Phase 1.0 Proposed
ZelM created on 9/15/2012. Last modified 9/15/2012
Extends

ADL Workbench

External «DFD_External» in package 'Model Transformation Map'

ADL Workbench
Version 1.0 Phase 1.0 Proposed
ZelM created on 9/15/2012. Last modified 3/22/2013

CONNECTORS
<p>↗ DataFlow «DFD_DataFlow» Source -> Destination From: ADL Workbench : External, Public To: OpenEHR ADL : DataStore, Public</p>
<p>↗ DataFlow «DFD_DataFlow» Source -> Destination From: ADL Workbench : External, Public To: CIMI ADL : DataStore, Public</p>
<p>↗ DataFlow «DFD_DataFlow» Source -> Destination From: BMM : DataStore, Public To: ADL Workbench : External, Public</p>

BMM

DataStore «DFD_DataStore» in package 'Model Transformation Map'

BMM
Version 1.0 Phase 1.0 Proposed

ZelM created on 9/15/2012. Last modified 9/15/2012

CONNECTORS **DataFlow** «DFD_DataFlow» Source -> Destination

From: BMM : DataStore, Public
 To: ADL Workbench : External, Public

 **DataFlow** «DFD_DataFlow» Source -> Destination

From: UML Class : DataStore, Public
 To: BMM : DataStore, Public

CEM*DataStore «DFD_DataStore» in package 'Model Transformation Map'*

CEM

Version 1.0 Phase 1.0 Proposed
ZelM created on 9/15/2012. Last modified 9/15/2012**CONNECTORS** **DataFlow** «DFD_DataFlow» import Source -> Destination

From: CEM : DataStore, Public
 To: DCM UML : DataStore, Public

CIMI ADL*DataStore «DFD_DataStore» in package 'Model Transformation Map'*

CIMI ADL

Version 1.0 Phase 1.0 Proposed
ZelM created on 9/15/2012. Last modified 9/15/2012**CONNECTORS** **DataFlow** «DFD_DataFlow» Source -> Destination

From: ADL Workbench : External, Public
 To: CIMI ADL : DataStore, Public

 **DataFlow** «DFD_DataFlow» Source -> Destination

From: DCM UML : DataStore, Public
 To: CIMI ADL : DataStore, Public

DCM UML*DataStore «DFD_DataStore» in package 'Model Transformation Map'*

DCM UML

Version 1.0 Phase 1.0 Proposed

ZelM created on 9/15/2012. Last modified 9/15/2012

CONNECTORS

 **DataFlow** «DFD_DataFlow» Source -> Destination

From: DCM UML : DataStore, Public
 To: CIMI ADL : DataStore, Public

 **DataFlow** «DFD_DataFlow» Source -> Destination

From: DCM UML : DataStore, Public
 To: Create Instance Example : Process, Public

 **DataFlow** «DFD_DataFlow» import Source -> Destination

From: CEM : DataStore, Public
 To: DCM UML : DataStore, Public

 **DataFlow** «DFD_DataFlow» Source -> Destination

From: UML Tool (EA, MDHT, RSA) : External, Public
 To: DCM UML : DataStore, Public

 **DataFlow** «DFD_DataFlow» import Source -> Destination

From: OpenEHR ADL : DataStore, Public
 To: DCM UML : DataStore, Public

HL7v3 CSP

DataStore «DFD_DataStore» in package 'Model Transformation Map'

HL7v3 CSP

Version 1.0 Phase 1.0 Proposed
 ZelM created on 9/16/2012. Last modified 9/16/2012

CONNECTORS

 **DataFlow** «DFD_DataFlow» transform Source -> Destination

From: UML Object : DataStore, Public
 To: HL7v3 CSP : DataStore, Public

LinkEHR

External «DFD_External» in package 'Model Transformation Map'

LinkEHR

Version 1.0 Phase 1.0 Proposed
 zelm created on 3/22/2013. Last modified 3/22/2013

CONNECTORS

CONNECTORS

 **DataFlow** «DFD_DataFlow» Source -> Destination

From: XML Schema : DataStore, Public
 To: LinkEHR : External, Public

MM Tool

External «DFD_External» in package 'Model Transformation Map'

MM Tool

Version 1.0 Phase 1.0 Proposed
 ZelM created on 9/15/2012. Last modified 9/15/2012

Mindmap

DataStore «DFD_DataStore» in package 'Model Transformation Map'

Mindmap

Version 1.0 Phase 1.0 Proposed
 ZelM created on 9/15/2012. Last modified 9/15/2012

Mindmap

DataStore «DFD_DataStore» in package 'Model Transformation Map'

Mindmap

Version 1.0 Phase 1.0 Proposed
 ZelM created on 9/15/2012. Last modified 9/15/2012

OpenEHR ADL

DataStore «DFD_DataStore» in package 'Model Transformation Map'

OpenEHR ADL

Version 1.0 Phase 1.0 Proposed
 ZelM created on 9/15/2012. Last modified 9/15/2012

CONNECTORS

 **DataFlow** «DFD_DataFlow» import Source -> Destination

From: OpenEHR ADL : DataStore, Public
 To: DCM UML : DataStore, Public

 **DataFlow** «DFD_DataFlow» Source -> Destination

From: ADL Workbench : External, Public
 To: OpenEHR ADL : DataStore, Public

UML AML

DataStore «DFD_DataStore» in package 'Model Transformation Map'

UML AML

Version 1.0 Phase 1.0 Proposed
ZelM created on 9/15/2012. Last modified 9/15/2012

CONNECTORS

 **DataFlow** «DFD_DataFlow» Source -> Destination

From: UML AML : DataStore, Public

To: UML Tool (EA, MDHT, RSA) : External, Public

UML Class*DataStore «DFD_DataStore» in package 'Model Transformation Map'*

UML Class

Version 1.0 Phase 1.0 Proposed
ZelM created on 9/15/2012. Last modified 9/15/2012

CONNECTORS

 **DataFlow** «DFD_DataFlow» Source -> Destination

From: UML Class : DataStore, Public

To: XML Schema : DataStore, Public

 **DataFlow** «DFD_DataFlow» Source -> Destination

From: UML Class : DataStore, Public

To: UML Tool (EA, MDHT, RSA) : External, Public

 **DataFlow** «DFD_DataFlow» Source -> Destination

From: UML Class : DataStore, Public

To: BMM : DataStore, Public

UML Class*DataStore «DFD_DataStore» in package 'Model Transformation Map'*

UML Class

Version 1.0 Phase 1.0 Proposed
ZelM created on 9/15/2012. Last modified 9/15/2012

CONNECTORS

 **DataFlow** «DFD_DataFlow» Source -> Destination

From: UML Class : DataStore, Public

To: Create Instance Example : Process, Public

 **DataFlow** «DFD_DataFlow» Source -> Destination

From: UML Tool (EA, MDHT, RSA) : External, Public

To: UML Class : DataStore, Public

UML Object

DataStore «DFD_DataStore» in package 'Model Transformation Map'

UML Object
Version 1.0 Phase 1.0 Proposed
ZelM created on 9/15/2012. Last modified 9/15/2012

CONNECTORS

↗ **DataFlow** «DFD_DataFlow» transform Source -> Destination
 From: UML Object : DataStore, Public
 To: HL7v3 CSP : DataStore, Public

↗ **DataFlow** «DFD_DataFlow» Source -> Destination
 From: Create Instance Example : Process, Public
 To: UML Object : DataStore, Public

UML Tool (EA, MDHT, RSA)

External «DFD_External» in package 'Model Transformation Map'

UML Tool (EA, MDHT, RSA)
Version 1.0 Phase 1.0 Proposed
ZelM created on 9/15/2012. Last modified 3/22/2013

CONNECTORS

↗ **DataFlow** «DFD_DataFlow» Source -> Destination
 From: UML Tool (EA, MDHT, RSA) : External, Public
 To: DCM UML : DataStore, Public

↗ **DataFlow** «DFD_DataFlow» Source -> Destination
 From: UML Tool (EA, MDHT, RSA) : External, Public
 To: UML Class : DataStore, Public

↗ **DataFlow** «DFD_DataFlow» Source -> Destination
 From: UML AML : DataStore, Public
 To: UML Tool (EA, MDHT, RSA) : External, Public

↗ **DataFlow** «DFD_DataFlow» Source -> Destination
 From: UML Class : DataStore, Public
 To: UML Tool (EA, MDHT, RSA) : External, Public

XML Schema

DataStore «DFD_DataStore» in package 'Model Transformation Map'

XML Schema

CONNECTORS

↗ **DataFlow** «DFD_DataFlow» Source -> Destination

From: XML Schema : DataStore, Public
To: LinkEHR : External, Public

↗ **DataFlow** «DFD_DataFlow» Source -> Destination

From: UML Class : DataStore, Public
To: XML Schema : DataStore, Public

Create Instance Example

Process «DFD_Process» in package 'Model Transformation Map'

Create Instance Example

Version 1.0 Phase 1.0 Proposed
ZelM created on 9/15/2012. Last modified 9/16/2012

CONNECTORS

↗ **DataFlow** «DFD_DataFlow» Source -> Destination

From: Create Instance Example : Process, Public
To: UML Object : DataStore, Public

↗ **DataFlow** «DFD_DataFlow» Source -> Destination

From: UML Class : DataStore, Public
To: Create Instance Example : Process, Public

↗ **DataFlow** «DFD_DataFlow» Source -> Destination

From: DCM UML : DataStore, Public
To: Create Instance Example : Process, Public

Tool Chain Use Case

Package in package 'Main'

Tool Chain Use Case
Version 1.0 Phase 1.0 Proposed
Michael created on 1/19/2013. Last modified 1/20/2013

Tool Chain diagram

Use Case diagram in package 'Tool Chain Use Case'

Tool Chain
Version 1.0
Michael created on 1/19/2013. Last modified 1/19/2013

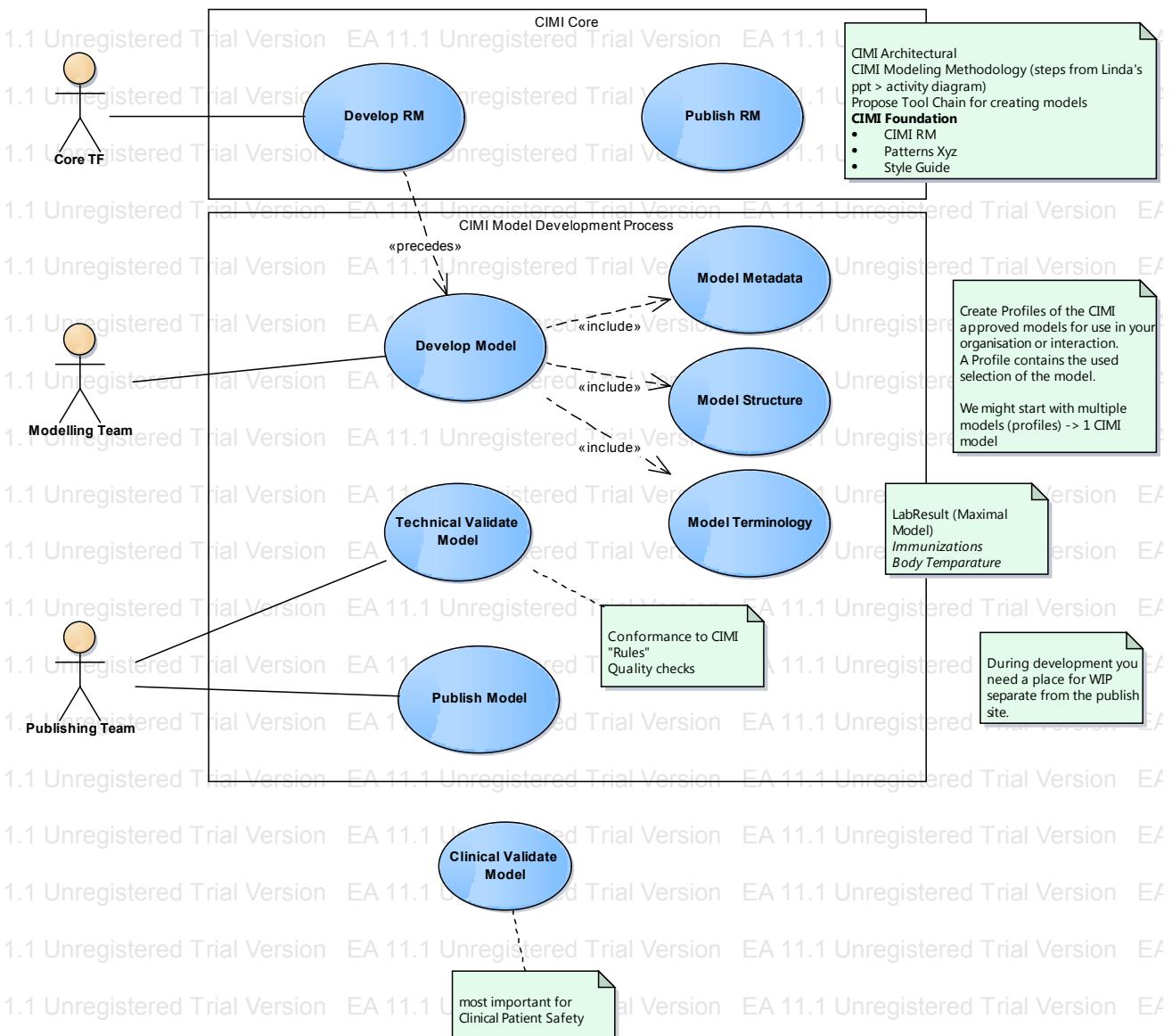


Figure 10: Tool Chain

CIMI Core

Boundary in package 'Tool Chain Use Case'

CIMI Core

Version 1.0 Phase 1.0 Proposed
Michael created on 1/19/2013. Last modified 1/19/2013
Extends

CIMI Model Development Process

Boundary in package 'Tool Chain Use Case'

CIMI Model Development Process
Version 1.0 Phase 1.0 Proposed
Michael created on 1/19/2013. Last modified 1/19/2013
Extends

<anonymous>

Note in package 'Tool Chain Use Case'

most important for Clinical Patient Safety

<anonymous>
Version 1.0 Phase 1.0 Proposed
Michael created on 1/19/2013. Last modified 1/19/2013
Extends

<anonymous>

Note in package 'Tool Chain Use Case'

Conformance to CIMI "Rules"
Quality checks

<anonymous>
Version 1.0 Phase 1.0 Proposed
Michael created on 1/19/2013. Last modified 1/19/2013
Extends

<anonymous>

Note in package 'Tool Chain Use Case'

Create Profiles of the CIMI approved models for use in your organisation or interaction.
A Profile contains the used selection of the model.

We might start with multiple models (profiles) -> 1 CIMI model

<anonymous>
Version 1.0 Phase 1.0 Proposed
Michael created on 1/19/2013. Last modified 1/19/2013
Extends

<anonymous>

Note in package 'Tool Chain Use Case'

CIMI Architectural

CIMI Modeling Methodology (steps from Linda's ppt > activity diagram)

Propose Tool Chain for creating models

CIMI Foundation

- CIMI RM
- Patterns Xyz
- Style Guide

<anonymous>

Version 1.0 Phase 1.0 Proposed

Michael created on 1/19/2013. Last modified 1/19/2013

Extends

<anonymous>*Note in package 'Tool Chain Use Case'*

LabResult (Maximal Model)

*Immunizations**Body Temparature*

<anonymous>

Version 1.0 Phase 1.0 Proposed

Michael created on 1/19/2013. Last modified 1/19/2013

Extends

<anonymous>*Note in package 'Tool Chain Use Case'*

During development you need a place for WIP separate from the publish site.

<anonymous>

Version 1.0 Phase 1.0 Proposed

Michael created on 1/19/2013. Last modified 1/19/2013

Extends

Core TF*Actor in package 'Tool Chain Use Case'*

Core TF

Version 1.0 Phase 1.0 Proposed

Michael created on 1/19/2013. Last modified 1/19/2013

CONNECTORS

CONNECTORS

 **UseCaseLink** Source -> Destination

From: Core TF : Actor, Public

To: Develop RM : UseCase, Public

Modelling Team

Actor in package 'Tool Chain Use Case'

Modelling Team
Version 1.0 Phase 1.0 Proposed
Michael created on 1/19/2013. Last modified 1/19/2013

CONNECTORS

 **UseCaseLink** Source -> Destination

From: Modelling Team : Actor, Public

To: Develop Model : UseCase, Public

Publishing Team

Actor in package 'Tool Chain Use Case'

Publishing Team
Version 1.0 Phase 1.0 Proposed
Michael created on 1/19/2013. Last modified 1/19/2013

CONNECTORS

 **UseCaseLink** Source -> Destination

From: Publishing Team : Actor, Public

To: Technical Validate Model : UseCase, Public

 **UseCaseLink** Source -> Destination

From: Publishing Team : Actor, Public

To: Publish Model : UseCase, Public

Clinical Validate Model

UseCase in package 'Tool Chain Use Case'

Clinical Validate Model
Version 1.0 Phase 1.0 Proposed
Michael created on 1/19/2013. Last modified 1/19/2013

Develop Model

UseCase in package 'Tool Chain Use Case'

Develop Model
Version 1.0 Phase 1.0 Proposed
Michael created on 1/19/2013. Last modified 1/19/2013

CONNECTORS

 **Include** «include» Source -> Destination

From: Develop Model : UseCase, Public
To: Model Metadata : UseCase, Public

 **Include** «include» Source -> Destination

From: Develop Model : UseCase, Public
To: Model Structure : UseCase, Public

 **Include** «include» Source -> Destination

From: Develop Model : UseCase, Public
To: Model Terminology : UseCase, Public

 **Dependency** «precedes» Source -> Destination

From: Develop RM : UseCase, Public
To: Develop Model : UseCase, Public

 **UseCaseLink** Source -> Destination

From: Modelling Team : Actor, Public
To: Develop Model : UseCase, Public

Develop RM

UseCase in package 'Tool Chain Use Case'

Develop RM
Version 1.0 Phase 1.0 Proposed
Michael created on 1/19/2013. Last modified 1/19/2013

CONNECTORS

 **Dependency** «precedes» Source -> Destination

From: Develop RM : UseCase, Public
To: Develop Model : UseCase, Public

 **UseCaseLink** Source -> Destination

From: Core TF : Actor, Public
To: Develop RM : UseCase, Public

Model Metadata

UseCase in package 'Tool Chain Use Case'

Model Metadata
Version 1.0 Phase 1.0 Proposed

Michael created on 1/19/2013. Last modified 1/19/2013

CONNECTORS

↗ **Include** «include» Source -> Destination
From: Develop Model : UseCase, Public
To: Model Metadata : UseCase, Public

Model Structure

UseCase in package 'Tool Chain Use Case'

Model Structure
Version 1.0 Phase 1.0 Proposed
Michael created on 1/19/2013. Last modified 1/19/2013

CONNECTORS

↗ **Include** «include» Source -> Destination
From: Develop Model : UseCase, Public
To: Model Structure : UseCase, Public

Model Terminology

UseCase in package 'Tool Chain Use Case'

Model Terminology
Version 1.0 Phase 1.0 Proposed
Michael created on 1/19/2013. Last modified 1/19/2013

CONNECTORS

↗ **Include** «include» Source -> Destination
From: Develop Model : UseCase, Public
To: Model Terminology : UseCase, Public

Publish Model

UseCase in package 'Tool Chain Use Case'

Publish Model
Version 1.0 Phase 1.0 Proposed
Michael created on 1/19/2013. Last modified 1/19/2013

CONNECTORS

↗ **UseCaseLink** Source -> Destination
From: Publishing Team : Actor, Public
To: Publish Model : UseCase, Public

Publish RM

UseCase in package 'Tool Chain Use Case'

Publish RM
Version 1.0 Phase 1.0 Proposed
Michael created on 1/19/2013. Last modified 1/19/2013

Technical Validate Model

UseCase in package 'Tool Chain Use Case'

Technical Validate Model
Version 1.0 Phase 1.0 Proposed
Michael created on 1/19/2013. Last modified 1/19/2013

CONNECTORS

 **UseCaseLink** Source -> Destination
From: Publishing Team : Actor, Public
To: Technical Validate Model : UseCase, Public

AML Tests

Package in package 'Main'

AML Tests
Version 1.0 Phase 1.0 Proposed
ZelM created on 4/13/2013. Last modified 4/13/2013

CIMI Patterns (proposed UML representation)

Package in package 'AML Tests'

CIMI Patterns (proposed UML representation)
Version 1.0 Phase 1.0 Proposed
zelm created on 10/4/2012. Last modified 4/13/2013

CIMI.RM.ClinicalEntryPattern

Package in package 'CIMI Patterns (proposed UML representation)'

CIMI.RM.ClinicalEntryPattern
Version 1.0 Phase 1.0 Proposed
zelm created on 10/4/2012. Last modified 10/5/2012

CIMI.RM.ClinicalEntryPattern diagram

Class diagram in package 'CIMI.RM.ClinicalEntryPattern'

CIMI.RM.ClinicalEntryPattern
Version 1.0
zelm created on 10/4/2012. Last modified 10/5/2012

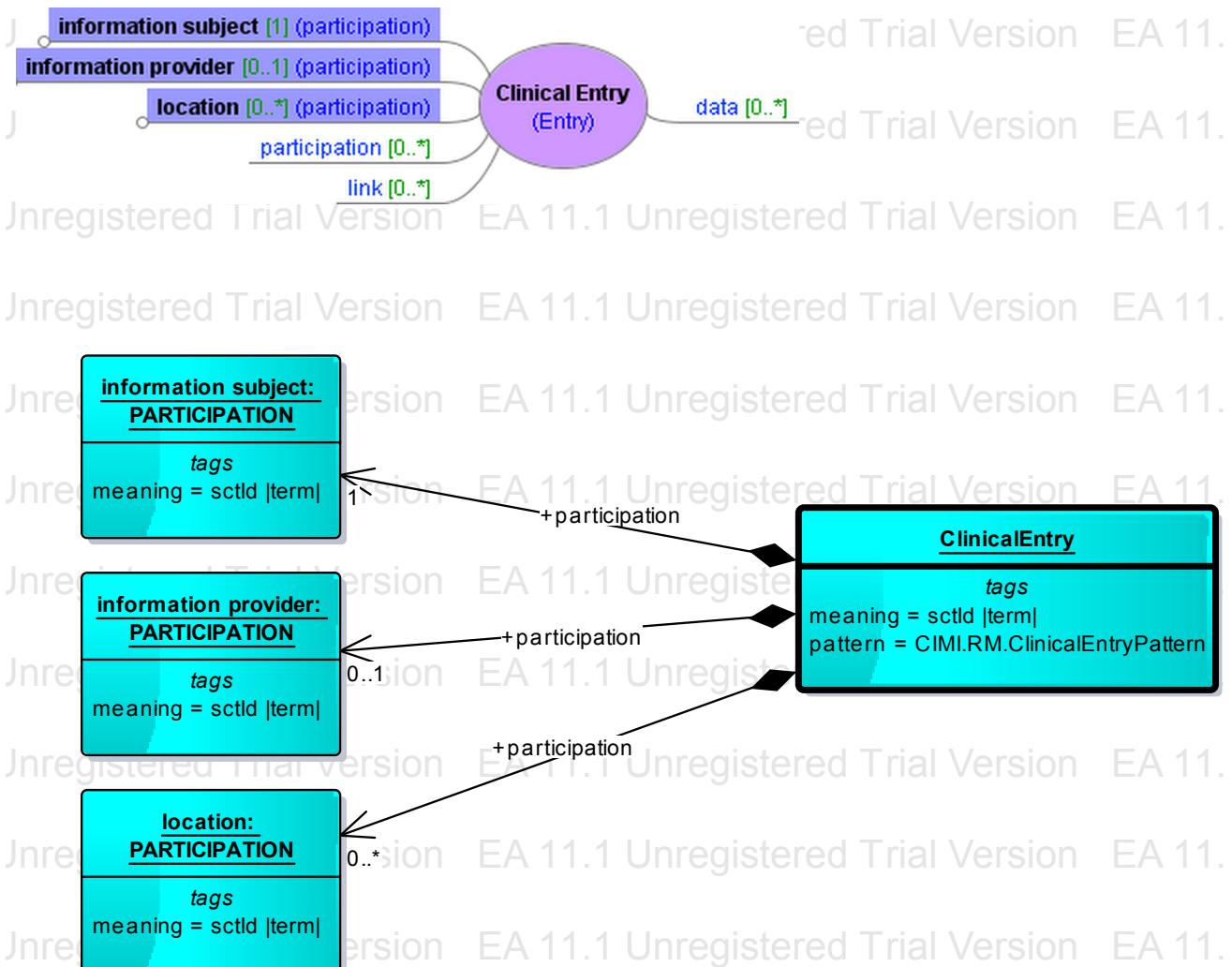


Figure 11: CIMI.RM.ClinicalEntryPattern

<anonymous>

Boundary in package 'CIMI.RM.ClinicalEntryPattern'

<anonymous>
 Version 1.0 Phase 1.0 Proposed
 zelm created on 10/4/2012. Last modified 10/4/2012
 Extends

ClinicalEntry

Object in package 'CIMI.RM.ClinicalEntryPattern'

ClinicalEntry
 Version 1.0 Phase 1.0 Proposed
 zelm created on 10/5/2012. Last modified 10/5/2012

ASSOCIATIONS
Association (direction: Source -> Destination)

ASSOCIATIONS	
Source: Public participation (Object) ClinicalEntry	Target: Public (Object) information subject Cardinality: [1]
Association (direction: Source -> Destination)	
Source: Public participation (Object) ClinicalEntry	Target: Public (Object) location Cardinality: [0..*]
Association (direction: Source -> Destination)	
Source: Public participation (Object) ClinicalEntry	Target: Public (Object) information provider Cardinality: [0..1]

information provider

Object in package 'CIMI.RM.ClinicalEntryPattern'

information provider
Version 1.0 Phase 1.0 Proposed
zelm created on 10/5/2012. Last modified 10/5/2012

ASSOCIATIONS	
Association (direction: Source -> Destination)	
Source: Public participation (Object) ClinicalEntry	Target: Public (Object) information provider Cardinality: [0..1]

information subject

Object in package 'CIMI.RM.ClinicalEntryPattern'

information subject
Version 1.0 Phase 1.0 Proposed
zelm created on 10/5/2012. Last modified 10/5/2012

ASSOCIATIONS	
Association (direction: Source -> Destination)	
Source: Public participation (Object) ClinicalEntry	Target: Public (Object) information subject Cardinality: [1]

location

Object in package 'CIMI.RM.ClinicalEntryPattern'

location
Version 1.0 Phase 1.0 Proposed
zelm created on 10/5/2012. Last modified 10/5/2012

ASSOCIATIONS
<p> Association (direction: Source -> Destination)</p> <p>Source: Public participation (Object) ClinicalEntry Target: Public (Object) location Cardinality: [0..*]</p>

CIMI.RM.ObservationPattern (mindmap)

Package in package 'CIMI Patterns (proposed UML representation)'

CIMI.RM.ObservationPattern (mindmap)

Version 1.0 Phase 1.0 Proposed
ZelM created on 10/13/2012. Last modified 10/13/2012

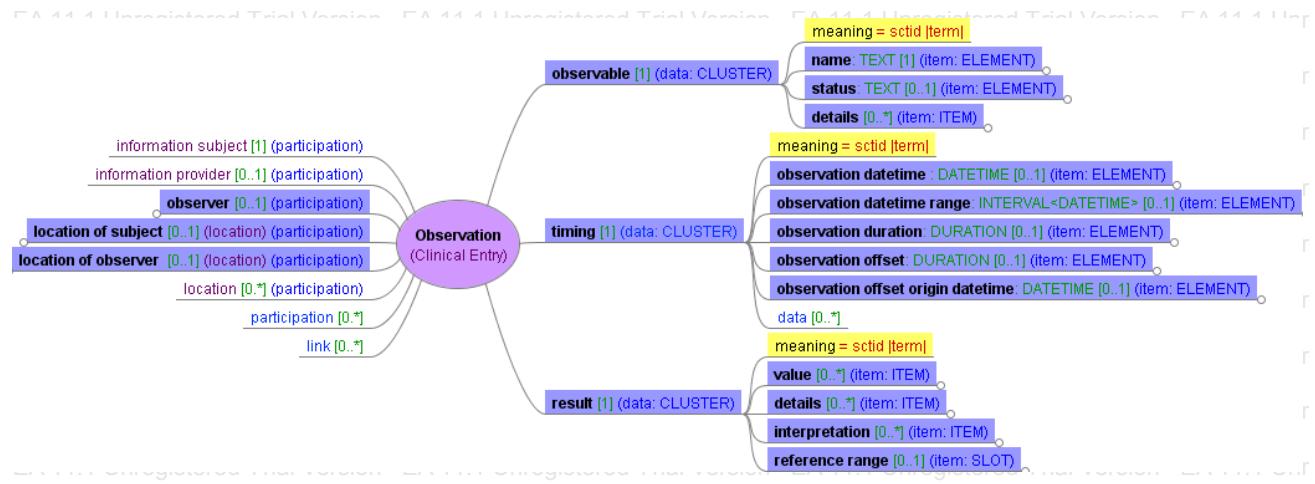
CIMI.RM.ObservationPattern (mindmap) diagram

Custom diagram in package 'CIMI.RM.ObservationPattern (mindmap)'

CIMI.RM.ObservationPattern (mindmap)

Version 1.0

ZelM created on 10/13/2012. Last modified 10/13/2012



EA 11.1 Unregistered Trial Version EA 11.1 Unregistered Trial Version EA 11.1 Unregistered Trial Version EA 11.1 Un

Figure 12: CIMI.RM.ObservationPattern (mindmap)

<anonymous>

Boundary in package 'CIMI.RM.ObservationPattern (mindmap)'

<anonymous>

Version 1.0 Phase 1.0 Proposed
ZelM created on 10/13/2012. Last modified 10/4/2012
Extends

CIMI.RM.ObservationPattern (Object)

Package in package 'CIMI Patterns (proposed UML representation)'

CIMI.RM.ObservationPattern (Object)

Version 1.0 Phase 1.0 Proposed
zelm created on 10/4/2012. Last modified 10/13/2012

CIMI.RM.ObservationPattern (Object) diagram

Class diagram in package 'CIMI.RM.ObservationPattern (Object)'

CIMI.RM.ObservationPattern (Object)

Version 1.0

zelm created on 10/4/2012. Last modified 10/13/2012

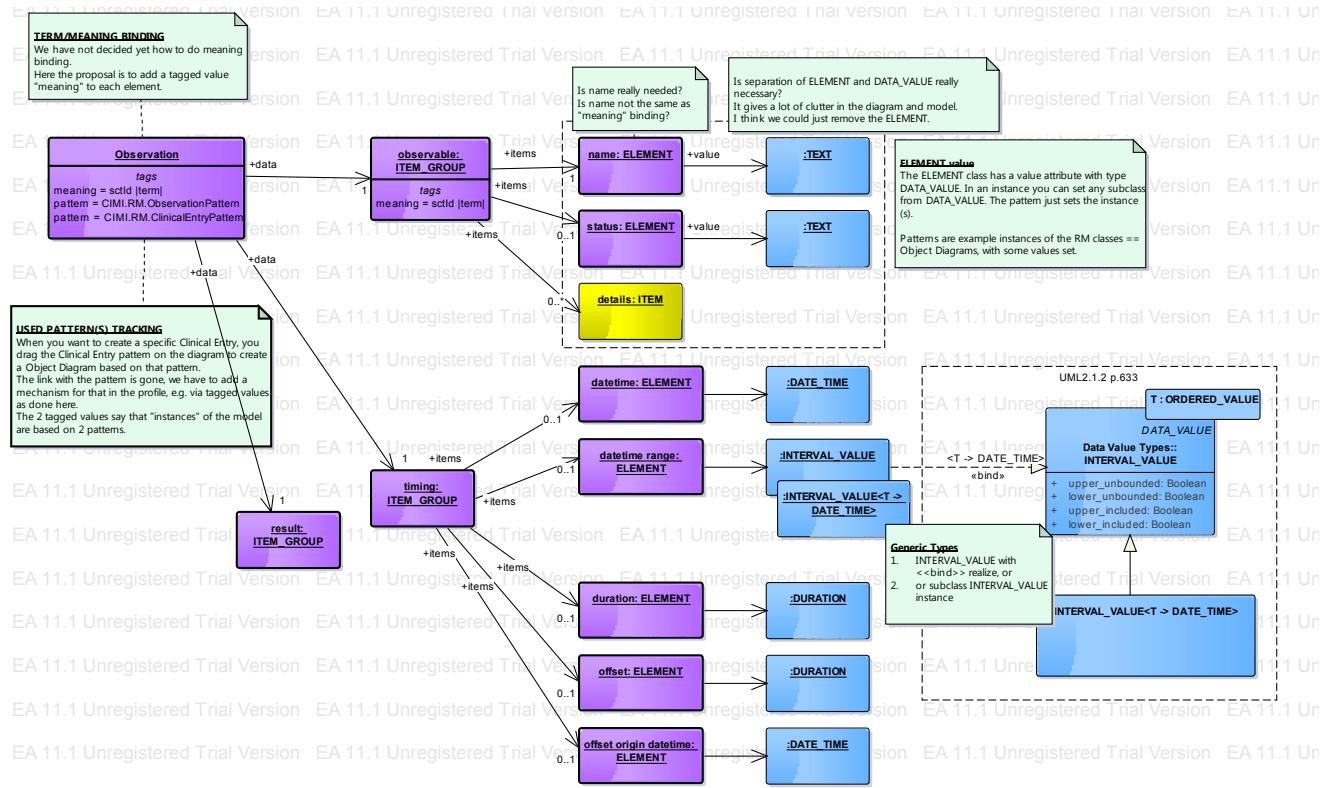


Figure 13: CIMI.RM.ObservationPattern (Object)

INTERVAL_VALUE

Class in package 'Data Value Types'

Generic class defining an interval (i.e. range) of a comparable type. An interval is used to define intervals of dates, times, quantities and so on. The type parameter, T, must be a descendant of the type ORDERED, which is necessary (but not sufficient) for instances to be compared (strictly_comparable is also needed). Without the INTERVAL class, quite a few more classes would be needed to express logical intervals, namely interval versions of all the date/time classes, and of quantity classes. Further, it allows the semantics of intervals to be stated in one place unequivocally, including the conditions for strict comparison.

INTERVAL_VALUE

Version 1.0 Phase 1.0 Proposed

Linda created on 5/1/2012. Last modified 4/13/2013

Extends DATA_VALUE

OUTGOING STRUCTURAL RELATIONSHIPS

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Generalization from INTERVAL_VALUE to DATA_VALUE	[Direction is 'Source -> Destination'.]
INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Realization «bind» from <anonymous> to INTERVAL_VALUE	[Name is <T -> DATE_TIME>. Direction is 'Source -> Destination'.]
⇒ Aggregation from ORDERED_VALUE to INTERVAL_VALUE	[Direction is 'Destination -> Source'.]
⇒ Generalization from INTERVAL_VALUE<T -> DATE_TIME> to INTERVAL_VALUE	[Direction is 'Source -> Destination'.]
⇒ Aggregation from ORDERED_VALUE to INTERVAL_VALUE	[Direction is 'Destination -> Source'.]
ATTRIBUTES	
◆ upper_unbounded : Boolean Public	[Is static False. Containment is Not Specified.]
◆ lower_unbounded : Boolean Public	[Is static False. Containment is Not Specified.]
◆ upper_included : Boolean Public	[Is static False. Containment is Not Specified.]
◆ lower_included : Boolean Public	[Is static False. Containment is Not Specified.]

<anonymous>

Boundary in package 'CIMI.RM.ObservationPattern (Object)'

<anonymous>
Version 1.0 Phase 1.0 Proposed
ZelM created on 10/5/2012. Last modified 10/5/2012
Extends

UML2.1.2 p.633

Boundary in package 'CIMI.RM.ObservationPattern (Object)'

UML2.1.2 p.633
Version 1.0 Phase 1.0 Proposed

<anonymous>

Note in package 'CIMI.RM.ObservationPattern (Object)'

ELEMENT.value

The ELEMENT class has a value attribute with type DATA_VALUE. In an instance you can set any subclass from DATA_VALUE. The pattern just sets the instance(s).

Patterns are example instances of the RM classes == Object Diagrams, with some values set.

<anonymous>
Version 1.0 Phase 1.0 Proposed
zelm created on 10/5/2012. Last modified 10/5/2012
Extends

<anonymous>

Note in package 'CIMI.RM.ObservationPattern (Object)'

USED PATTERN(S) TRACKING

When you want to create a specific Clinical Entry, you drag the Clinical Entry pattern on the diagram to create a Object Diagram based on that pattern.

The link with the pattern is gone, we have to add a mechanism for that in the profile, e.g. via tagged values as done here. The 2 tagged values say that "instances" of the model are based on 2 patterns.

<anonymous>
Version 1.0 Phase 1.0 Proposed
zelm created on 10/5/2012. Last modified 10/5/2012
Extends

<anonymous>

Note in package 'CIMI.RM.ObservationPattern (Object)'

TERM/MEANING BINDING

We have not decided yet how to do meaning binding.

Here the proposal is to add a tagged value "meaning" to each element.

<anonymous>
Version 1.0 Phase 1.0 Proposed
zelm created on 10/5/2012. Last modified 10/5/2012
Extends

<anonymous>

Note in package 'CIMI.RM.ObservationPattern (Object)'

Generic Types

1. INTERVAL_VALUE with <>bind>> realize, or
2. or subclass INTERVAL_VALUE instance

<anonymous>
Version 1.0 Phase 1.0 Proposed
zelm created on 10/5/2012. Last modified 10/5/2012
Extends

<anonymous>

Note in package 'CIMI.RM.ObservationPattern (Object)'

Is name really needed?
Is name not the same as "meaning" binding?

<anonymous>
Version 1.0 Phase 1.0 Proposed
zelm created on 10/5/2012. Last modified 10/5/2012
Extends

<anonymous>

Note in package 'CIMI.RM.ObservationPattern (Object)'

Is separation of ELEMENT and DATA_VALUE really necessary?
It gives a lot of clutter in the diagram and model.
I think we could just remove the ELEMENT.

<anonymous>
Version 1.0 Phase 1.0 Proposed
ZelM created on 10/5/2012. Last modified 10/5/2012
Extends

INTERVAL_VALUE<T -> DATE_TIME>

Class in package 'CIMI.RM.ObservationPattern (Object)'

INTERVAL_VALUE<T -> DATE_TIME>
Version 1.0 Phase 1.0 Proposed
zelm created on 10/5/2012. Last modified 10/5/2012
Extends INTERVAL_VALUE

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Generalization from INTERVAL_VALUE<T -> DATE_TIME> to INTERVAL_VALUE
[Direction is 'Source -> Destination'.]

<anonymous>

Object in package 'CIMI.RM.ObservationPattern (Object)'

<anonymous>

Version 1.0 Phase 1.0 Proposed
zelm created on 10/5/2012. Last modified 10/5/2012**ASSOCIATIONS**

Association (direction: Source -> Destination)

Source: Public (Object) offset origin datetime

Target: Public (Object) <anonymous>

<anonymous>*Object in package 'CIMI.RM.ObservationPattern (Object)'*

<anonymous>

Version 1.0 Phase 1.0 Proposed
zelm created on 10/5/2012. Last modified 10/5/2012**ASSOCIATIONS**

Association (direction: Source -> Destination)

Source: Public (Object) datetime

Target: Public (Object) <anonymous>

<anonymous>*Object in package 'CIMI.RM.ObservationPattern (Object)'*

<anonymous>

Version 1.0 Phase 1.0 Proposed
zelm created on 10/5/2012. Last modified 10/5/2012**ASSOCIATIONS**

Association (direction: Source -> Destination)

Source: Public (Object) offset

Target: Public (Object) <anonymous>

<anonymous>*Object in package 'CIMI.RM.ObservationPattern (Object)'*

<anonymous>

Version 1.0 Phase 1.0 Proposed
zelm created on 10/5/2012. Last modified 10/5/2012**ASSOCIATIONS**

Association (direction: Source -> Destination)

Source: Public (Object) duration

Target: Public (Object) <anonymous>

<anonymous>*Object in package 'CIMI.RM.ObservationPattern (Object)'*

<anonymous>

Version 1.0 Phase 1.0 Proposed
zelm created on 10/5/2012. Last modified 10/5/2012**OUTGOING STRUCTURAL RELATIONSHIPS** Realization «bind» from <anonymous> to INTERVAL_VALUE

[Name is <T -> DATE_TIME>. Direction is 'Source -> Destination'.]

ASSOCIATIONS Association (direction: Source -> Destination)

Source: Public (Object) datetime range

Target: Public (Object) <anonymous>

<anonymous>*Object in package 'CIMI.RM.ObservationPattern (Object)'*

<anonymous>

Version 1.0 Phase 1.0 Proposed
zelm created on 10/5/2012. Last modified 10/5/2012**<anonymous>***Object in package 'CIMI.RM.ObservationPattern (Object)'*

<anonymous>

Version 1.0 Phase 1.0 Proposed
zelm created on 10/5/2012. Last modified 10/5/2012**ASSOCIATIONS** Association (direction: Source -> Destination)

Source: Public value (Object) status

Target: Public (Object) <anonymous>

<anonymous>*Object in package 'CIMI.RM.ObservationPattern (Object)'*

<anonymous>

Version 1.0 Phase 1.0 Proposed
zelm created on 10/5/2012. Last modified 10/5/2012**ASSOCIATIONS**

ASSOCIATIONS	
Association (direction: Source -> Destination)	
Source: Public value (Object) name	Target: Public (Object) <anonymous>

Observation

Object in package 'CIMI.RM.ObservationPattern (Object)'

Observation
Version 1.0 Phase 1.0 Proposed
zelm created on 10/5/2012. Last modified 10/5/2012

ASSOCIATIONS	
Association (direction: Source -> Destination)	
Source: Public data (Object) Observation	Target: Public (Object) timing Cardinality: [1]
Association (direction: Source -> Destination)	
Source: Public data (Object) Observation	Target: Public (Object) result Cardinality: [1]
Association (direction: Source -> Destination)	
Source: Public data (Object) Observation	Target: Public (Object) observable Cardinality: [1]

datetime range

Object in package 'CIMI.RM.ObservationPattern (Object)'

datetime range
Version 1.0 Phase 1.0 Proposed
zelm created on 10/5/2012. Last modified 10/5/2012

ASSOCIATIONS	
Association (direction: Source -> Destination)	
Source: Public (Object) datetime range	Target: Public (Object) <anonymous>
Association (direction: Source -> Destination)	
Source: Public items (Object) timing	Target: Public (Object) datetime range Cardinality: [0..1]

datetime

Object in package 'CIMI.RM.ObservationPattern (Object)'

datetime
Version 1.0 Phase 1.0 Proposed
zelm created on 10/5/2012. Last modified 10/5/2012

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Object) datetime

Target: Public (Object) <anonymous>

Association (direction: Source -> Destination)

Source: Public items (Object) timing

Target: Public (Object) datetime
Cardinality: [0..1]

details

Object in package 'CIMI.RM.ObservationPattern (Object)'

details
Version 1.0 Phase 1.0 Proposed
zelm created on 10/5/2012. Last modified 10/5/2012

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public items (Object) observable

Target: Public (Object) details
Cardinality: [0..*]

duration

Object in package 'CIMI.RM.ObservationPattern (Object)'

duration
Version 1.0 Phase 1.0 Proposed
zelm created on 10/5/2012. Last modified 10/5/2012

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Object) duration

Target: Public (Object) <anonymous>

Association (direction: Source -> Destination)

ASSOCIATIONS

Source: Public items (Object) timing

Target: Public (Object) duration

Cardinality: [0..1]

name*Object in package 'CIMI.RM.ObservationPattern (Object)'*

name

Version 1.0 Phase 1.0 Proposed
zelm created on 10/4/2012. Last modified 10/4/2012**ASSOCIATIONS**

Association (direction: Source -> Destination)

Source: Public value (Object) name

Target: Public (Object) <anonymous>

Association (direction: Source -> Destination)

Source: Public items (Object) observable

Target: Public (Object) name
Cardinality: [1]***observable****Object in package 'CIMI.RM.ObservationPattern (Object)'*

observable

Version 1.0 Phase 1.0 Proposed
zelm created on 10/4/2012. Last modified 10/4/2012**ASSOCIATIONS**

Association (direction: Source -> Destination)

Source: Public items (Object) observable

Target: Public (Object) details
Cardinality: [0..*]

Association (direction: Source -> Destination)

Source: Public items (Object) observable

Target: Public (Object) status
Cardinality: [0..1]

Association (direction: Source -> Destination)

Source: Public items (Object) observable

Target: Public (Object) name
Cardinality: [1]

Association (direction: Source -> Destination)

ASSOCIATIONS

Source: Public data (Object) Observation

Target: Public (Object) observable

Cardinality: [1]

offset origin datetime*Object in package 'CIMI.RM.ObservationPattern (Object)'*

offset origin datetime

Version 1.0 Phase 1.0 Proposed
zelm created on 10/5/2012. Last modified 10/5/2012**ASSOCIATIONS**

Association (direction: Source -> Destination)

Source: Public (Object) offset origin datetime

Target: Public (Object) <anonymous>

Association (direction: Source -> Destination)

Source: Public items (Object) timing

Target: Public (Object) offset origin datetime
Cardinality: [0..1]**offset***Object in package 'CIMI.RM.ObservationPattern (Object)'*

offset

Version 1.0 Phase 1.0 Proposed
zelm created on 10/5/2012. Last modified 10/5/2012**ASSOCIATIONS**

Association (direction: Source -> Destination)

Source: Public (Object) offset

Target: Public (Object) <anonymous>

Association (direction: Source -> Destination)

Source: Public items (Object) timing

Target: Public (Object) offset
Cardinality: [0..1]**result***Object in package 'CIMI.RM.ObservationPattern (Object)'*

result

Version 1.0 Phase 1.0 Proposed
zelm created on 10/4/2012. Last modified 10/4/2012

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public data (Object) Observation

Target: Public (Object) result
Cardinality: [1]

status

Object in package 'CIMI.RM.ObservationPattern (Object)'

status

Version 1.0 Phase 1.0 Proposed
zelm created on 10/4/2012. Last modified 10/4/2012

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public value (Object) status

Target: Public (Object) <anonymous>

Association (direction: Source -> Destination)

Source: Public items (Object) observable

Target: Public (Object) status
Cardinality: [0..1]

timing

Object in package 'CIMI.RM.ObservationPattern (Object)'

timing

Version 1.0 Phase 1.0 Proposed
zelm created on 10/4/2012. Last modified 10/4/2012

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public items (Object) timing

Target: Public (Object) offset origin datetime
Cardinality: [0..1]

Association (direction: Source -> Destination)

Source: Public items (Object) timing

Target: Public (Object) offset
Cardinality: [0..1]

Association (direction: Source -> Destination)

Source: Public items (Object) timing

Target: Public (Object) datetime
Cardinality: [0..1]

ASSOCIATIONS	
 Association (direction: Source -> Destination)	Source: Public items (Object) timing Target: Public (Object) duration Cardinality: [0..1]
 Association (direction: Source -> Destination)	Source: Public items (Object) timing Target: Public (Object) datetime range Cardinality: [0..1]
 Association (direction: Source -> Destination)	Source: Public data (Object) Observation Target: Public (Object) timing Cardinality: [1]

CIMI.RM.ObservationPattern (Class)

Package in package 'CIMI Patterns (proposed UML representation)'

CIMI.RM.ObservationPattern (Class)
Version 1.0 Phase 1.0 Proposed
ZelM created on 10/13/2012. Last modified 10/13/2012

CIMI.RM.ObservationPattern (Class) diagram

Class diagram in package 'CIMI.RM.ObservationPattern (Class)'

CIMI.RM.ObservationPattern (Class)
Version 1.0
ZelM created on 10/13/2012. Last modified 12/16/2013

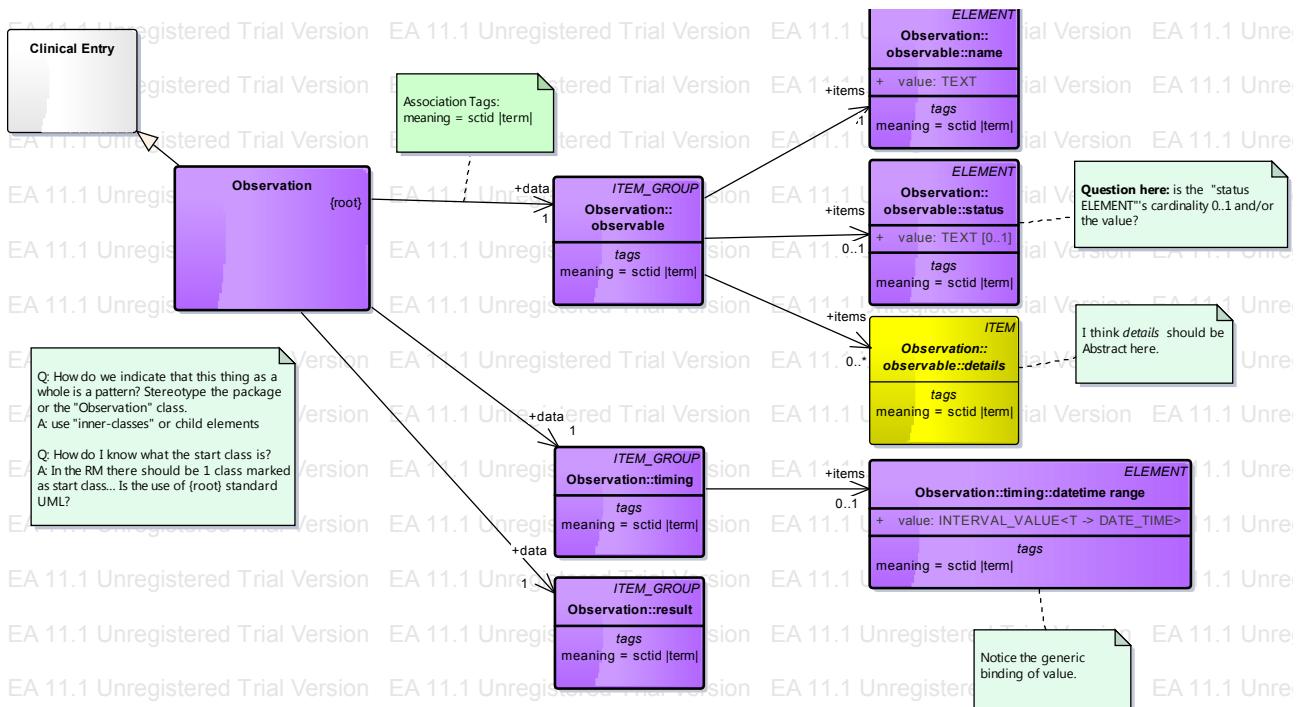


Figure 14: CIMI.RM.ObservationPattern (Class)

<anonymous>

Note in package 'CIMI.RM.ObservationPattern (Class)'

<anonymous>
Version 1.0 Phase 1.0 Proposed
ZelM created on 10/13/2012. Last modified 10/13/2012
Extends

<anonymous>

Note in package 'CIMI.RM.ObservationPattern (Class)'

Question here: is the "status ELEMENT"s cardinality 0..1 and/or the value?

<anonymous>
Version 1.0 Phase 1.0 Proposed
ZelM created on 10/13/2012. Last modified 10/13/2012
Extends

<anonymous>

Note in package 'CIMI.RM.ObservationPattern (Class)'

Q: How do we indicate that this thing as a whole is a pattern? Stereotype the package or the "Observation" class.

A: use "inner-classes" or child elements

Q: How do I know what the start class is?

A: In the RM there should be 1 class marked as start class... Is the use of {root} standard UML?

<anonymous>

Version 1.0 Phase 1.0 Proposed

ZelM created on 10/13/2012. Last modified 10/16/2012

Extends

<anonymous>

Note in package 'CIMI.RM.ObservationPattern (Class)'

I think *details* should be Abstract here.

<anonymous>

Version 1.0 Phase 1.0 Proposed

ZelM created on 10/13/2012. Last modified 10/13/2012

Extends

<anonymous>

Note in package 'CIMI.RM.ObservationPattern (Class)'

Notice the generic binding of value.

<anonymous>

Version 1.0 Phase 1.0 Proposed

ZelM created on 10/15/2012. Last modified 10/15/2012

Extends

Clinical Entry

Class in package 'CIMI.RM.ObservationPattern (Class)'

Clinical Entry

Version 1.0 Phase 1.0 Proposed

ZelM created on 10/13/2012. Last modified 10/13/2012

INCOMING STRUCTURAL RELATIONSHIPS

⇒ Generalization from Observation to Clinical Entry

[Direction is 'Source -> Destination'.]

Observation

Class in package 'CIMI.RM.ObservationPattern (Class)'

Observation
Version 1.0 Phase 1.0 Proposed
ZelM created on 10/13/2012. Last modified 7/12/2013
Extends Clinical Entry

ELEMENTS OWNED BY Observation	
 observable : Class	
 result : Class	
 timing : Class	

OUTGOING STRUCTURAL RELATIONSHIPS	
 Generalization from Observation to Clinical Entry	[Direction is 'Source -> Destination'.]

ASSOCIATIONS	
 Association (direction: Source -> Destination)	
Source: Public (Class) Observation	Target: Public data (Class) timing Cardinality: [1]
 Association (direction: Source -> Destination)	
Source: Public (Class) Observation	Target: Public data (Class) result Cardinality: [1]
 Association (direction: Source -> Destination)	
Source: Public (Class) Observation	Target: Public data (Class) observable Cardinality: [1]

observable

Class owned by 'Observation', in package 'CIMI.RM.ObservationPattern (Class)'

observable
Version 1.0 Phase 1.0 Proposed
ZelM created on 10/13/2012. Last modified 10/15/2012
Extends ITEM_GROUP

ELEMENTS OWNED BY observable

ELEMENTS OWNED BY observable	
 details : Class	
 name : Class	
 status : Class	

OUTGOING STRUCTURAL RELATIONSHIPS	
 Generalization from observable to ITEM_GROUP	[Direction is 'Source -> Destination'.]

ASSOCIATIONS	
 Association (direction: Source -> Destination)	
Source: Public (Class) observable	Target: Public items (Class) status Cardinality: [0..1]
 Association (direction: Source -> Destination)	
Source: Public (Class) observable	Target: Public items (Class) name Cardinality: [1]
 Association (direction: Source -> Destination)	
Source: Public (Class) observable	Target: Public items (Class) details Cardinality: [0..*]
 Association (direction: Source -> Destination)	
Source: Public (Class) Observation	Target: Public data (Class) observable Cardinality: [1]

details

Class owned by 'observable', in package 'CIMI.RM.ObservationPattern (Class)'

details
Version 1.0 Phase 1.0 Proposed
ZelM created on 10/13/2012. Last modified 10/15/2012
Extends ITEM

OUTGOING STRUCTURAL RELATIONSHIPS	
 Generalization from details to ITEM	[Direction is 'Source -> Destination'.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) observable

Target: Public items (Class) details
Cardinality: [0..*]

name

Class owned by 'observable', in package 'CIMI.RM.ObservationPattern (Class)'

name
Version 1.0 Phase 1.0 Proposed
ZelM created on 10/13/2012. Last modified 10/15/2012
Extends ELEMENT

OUTGOING STRUCTURAL RELATIONSHIPS

Generalization from name to ELEMENT

[Direction is 'Source -> Destination'.]

ATTRIBUTES

value : TEXT Public

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) observable

Target: Public items (Class) name
Cardinality: [1]

status

Class owned by 'observable', in package 'CIMI.RM.ObservationPattern (Class)'

status
Version 1.0 Phase 1.0 Proposed
ZelM created on 10/13/2012. Last modified 10/15/2012
Extends ELEMENT

OUTGOING STRUCTURAL RELATIONSHIPS

Generalization from status to ELEMENT

[Direction is 'Source -> Destination'.]

ATTRIBUTES

value : TEXT Public

Multiplicity: ([0..1], Allow duplicates: 0, Is ordered: False)

ATTRIBUTES
[Is static False. Containment is Not Specified.]

ASSOCIATIONS
<p> Association (direction: Source -> Destination)</p> <p>Source: Public (Class) observable Target: Public items (Class) status Cardinality: [0..1]</p>

result

Class owned by 'Observation', in package 'CIMI.RM.ObservationPattern (Class)'

result
Version 1.0 Phase 1.0 Proposed
ZelM created on 10/13/2012. Last modified 10/15/2012
Extends ITEM_GROUP

OUTGOING STRUCTURAL RELATIONSHIPS
<p> Generalization from result to ITEM_GROUP</p> <p>[Direction is 'Source -> Destination'.]</p>

ASSOCIATIONS
<p> Association (direction: Source -> Destination)</p> <p>Source: Public (Class) Observation Target: Public data (Class) result Cardinality: [1]</p>

timing

Class owned by 'Observation', in package 'CIMI.RM.ObservationPattern (Class)'

timing
Version 1.0 Phase 1.0 Proposed
ZelM created on 10/13/2012. Last modified 10/15/2012
Extends ITEM_GROUP

ELEMENTS OWNED BY timing
<p> datetime range : Class</p>

OUTGOING STRUCTURAL RELATIONSHIPS
<p> Generalization from timing to ITEM_GROUP</p> <p>[Direction is 'Source -> Destination'.]</p>

ASSOCIATIONS

✍ Association (direction: Source -> Destination)

Source: Public (Class) timing

Target: Public items (Class) datetime range
Cardinality: [0..1]

✍ Association (direction: Source -> Destination)

Source: Public (Class) Observation

Target: Public data (Class) timing
Cardinality: [1]

datetime range

Class owned by 'timing', in package 'CIMI.RM.ObservationPattern (Class)'

datetime range
Version 1.0 Phase 1.0 Proposed
ZelM created on 10/14/2012. Last modified 10/15/2012
Extends ELEMENT

OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Generalization from datetime range to ELEMENT

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ value : INTERVAL_VALUE<T -> DATE_TIME> Public

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

✍ Association (direction: Source -> Destination)

Source: Public (Class) timing

Target: Public items (Class) datetime range
Cardinality: [0..1]

Slots in UML Example

Package in package 'AML Tests'

Slots in UML Example
Version 1.0 Phase 1.0 Proposed
ZelM created on 12/4/2012. Last modified 4/13/2013

Slots in UML Example diagram

Class diagram in package 'Slots in UML Example'

Slots in UML Example
Version 1.0
ZelM created on 12/4/2012. Last modified 12/4/2012

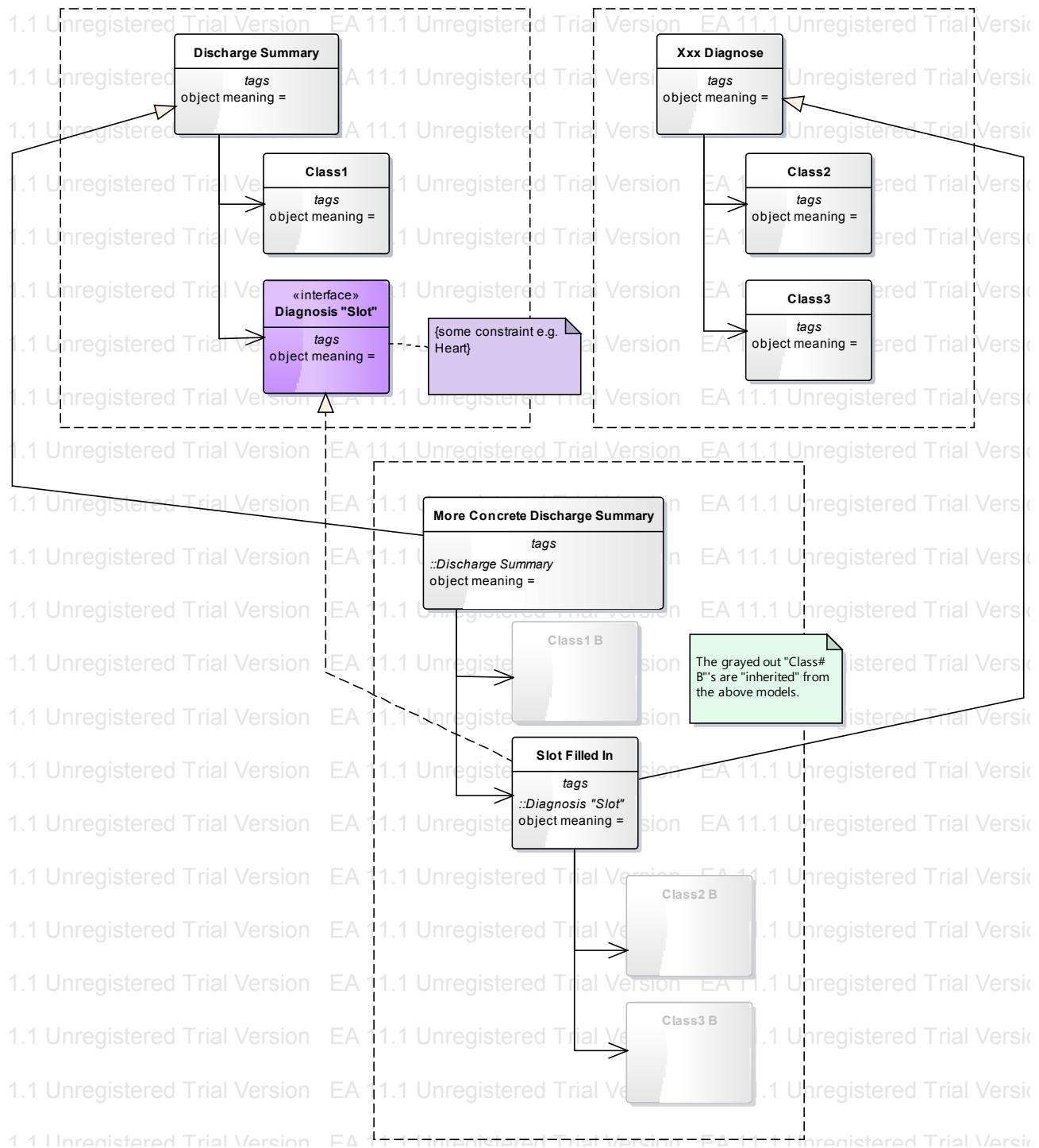


Figure 15: Slots in UML Example

<anonymous>

Boundary in package 'Slots in UML Example'

<anonymous>
Version 1.0 Phase 1.0 Proposed
ZelM created on 12/4/2012. Last modified 12/4/2012
Extends

<anonymous>

Boundary in package 'Slots in UML Example'

<anonymous>
Version 1.0 Phase 1.0 Proposed
ZelM created on 12/4/2012. Last modified 12/4/2012
Extends

<anonymous>

Boundary in package 'Slots in UML Example'

<anonymous>
Version 1.0 Phase 1.0 Proposed
ZelM created on 12/4/2012. Last modified 12/4/2012
Extends

<anonymous>

Constraint in package 'Slots in UML Example'

some constraint e.g. Heart

<anonymous>
Version 1.0 Phase 1.0 Proposed
ZelM created on 12/4/2012. Last modified 12/4/2012
Extends

<anonymous>

Note in package 'Slots in UML Example'

The grayed out "Class# B"'s are "inherited" from the above models.

<anonymous>
Version 1.0 Phase 1.0 Proposed
ZelM created on 12/4/2012. Last modified 12/4/2012
Extends

Class1

Class in package 'Slots in UML Example'

Class1
Version 1.0 Phase 1.0 Proposed
ZelM created on 12/4/2012. Last modified 12/4/2012

ASSOCIATIONS
<p> Association (direction: Source -> Destination)</p> <p>Source: Public (Class) Discharge Summary Target: Public (Class) Class1</p>

Class1 B

Class in package 'Slots in UML Example'

Class1 B
Version 1.0 Phase 1.0 Proposed
ZelM created on 12/4/2012. Last modified 12/4/2012

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) More Concrete Discharge Summary

Target: Public (Class) Class1 B

Class2

Class in package 'Slots in UML Example'

Class2
Version 1.0 Phase 1.0 Proposed
ZelM created on 12/4/2012. Last modified 12/4/2012

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) Xxx Diagnose

Target: Public (Class) Class2

Class2 B

Class in package 'Slots in UML Example'

Class2 B
Version 1.0 Phase 1.0 Proposed
ZelM created on 12/4/2012. Last modified 12/4/2012

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) Slot Filled In

Target: Public (Class) Class2 B

Class3

Class in package 'Slots in UML Example'

Class3
Version 1.0 Phase 1.0 Proposed
ZelM created on 12/4/2012. Last modified 12/4/2012

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) Xxx Diagnose

Target: Public (Class) Class3

Class3 B

Class in package 'Slots in UML Example'

Class3 B
Version 1.0 Phase 1.0 Proposed
ZelM created on 12/4/2012. Last modified 12/4/2012

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) Slot Filled In

Target: Public (Class) Class3 B

Discharge Summary

Class in package 'Slots in UML Example'

Discharge Summary
Version 1.0 Phase 1.0 Proposed
ZelM created on 12/4/2012. Last modified 12/4/2012

INCOMING STRUCTURAL RELATIONSHIPS

Generalization from More Concrete Discharge Summary to Discharge Summary

[Direction is 'Source -> Destination'.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) Discharge Summary

Target: Public (Interface) Diagnosis "Slot"

Association (direction: Source -> Destination)

Source: Public (Class) Discharge Summary

Target: Public (Class) Class1

More Concrete Discharge Summary

Class in package 'Slots in UML Example'

More Concrete Discharge Summary
Version 1.0 Phase 1.0 Proposed
ZelM created on 12/4/2012. Last modified 12/4/2012

Extends Discharge Summary

OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Generalization from More Concrete Discharge Summary to Discharge Summary

[Direction is 'Source -> Destination'.]

ASSOCIATIONS

✍ Association (direction: Source -> Destination)

Source: Public (Class) More Concrete Discharge Summary

Target: Public (Class) Class1 B

✍ Association (direction: Source -> Destination)

Source: Public (Class) More Concrete Discharge Summary

Target: Public (Class) Slot Filled In

Slot Filled In

Class in package 'Slots in UML Example'

Slot Filled In
 Version 1.0 Phase 1.0 Proposed
 ZelM created on 12/4/2012. Last modified 12/4/2012
 Extends Xxx Diagnose

OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Realization from Slot Filled In to Diagnosis "Slot"

[Direction is 'Source -> Destination'.]

⬅ Generalization from Slot Filled In to Xxx Diagnose

[Direction is 'Source -> Destination'.]

ASSOCIATIONS

✍ Association (direction: Source -> Destination)

Source: Public (Class) Slot Filled In

Target: Public (Class) Class2 B

✍ Association (direction: Source -> Destination)

Source: Public (Class) Slot Filled In

Target: Public (Class) Class3 B

✍ Association (direction: Source -> Destination)

Source: Public (Class) More Concrete Discharge Summary

Target: Public (Class) Slot Filled In

Xxx Diagnose

Class in package 'Slots in UML Example'

Xxx Diagnose
Version 1.0 Phase 1.0 Proposed
ZelM created on 12/4/2012. Last modified 12/4/2012

INCOMING STRUCTURAL RELATIONSHIPS

⇒ Generalization from Slot Filled In to Xxx Diagnose

[Direction is 'Source -> Destination'.]

ASSOCIATIONS

↙ Association (direction: Source -> Destination)

Source: Public (Class) Xxx Diagnose

Target: Public (Class) Class3

↙ Association (direction: Source -> Destination)

Source: Public (Class) Xxx Diagnose

Target: Public (Class) Class2

Diagnosis "Slot"

Interface in package 'Slots in UML Example'

Diagnosis "Slot"
Version 1.0 Phase 1.0 Proposed
ZelM created on 12/4/2012. Last modified 12/4/2012

INCOMING STRUCTURAL RELATIONSHIPS

⇒ Realization from Slot Filled In to Diagnosis "Slot"

[Direction is 'Source -> Destination'.]

ASSOCIATIONS

↙ Association (direction: Source -> Destination)

Source: Public (Class) Discharge Summary

Target: Public (Interface) Diagnosis "Slot"

Closed class

Package in package 'AML Tests'

Closed class
Version 1.0 Phase 1.0 Proposed
ZelM created on 4/13/2013. Last modified 4/13/2013

Closed class diagram

Class diagram in package 'Closed class'

Closed class
Version 1.0
ZelM created on 4/13/2013. Last modified 4/13/2013

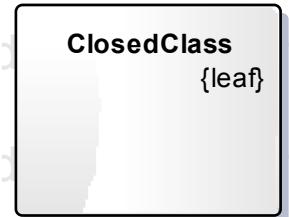


Figure 16: Closed class

ClosedClass

Class in package 'Closed class'

ClosedClass
Version 1.0 Phase 1.0 Proposed
ZelM created on 4/13/2013. Last modified 4/13/2013

Parameterised Classes

Package in package 'AML Tests'

Parameterised Classes
Version 1.0 Phase 1.0 Proposed
ZelM created on 4/13/2013. Last modified 4/13/2013

Parameterised Classes diagram

Class diagram in package 'Parameterised Classes'

Parameterised Classes
Version 1.0
ZelM created on 4/13/2013. Last modified 4/13/2013

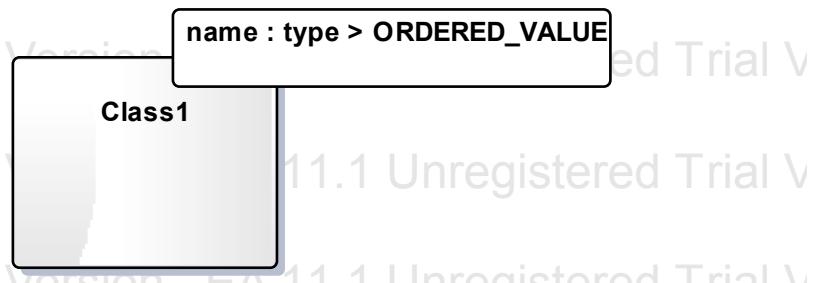


Figure 17: Parameterised Classes

Class1

Class in package 'Parameterised Classes'

Class1
Version 1.0 Phase 1.0 Proposed
ZelM created on 4/13/2013. Last modified 4/13/2013