

## 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 1/8/2015

**Legend**

**Mapping CIMI RM 2 HL7 RIM : HL7 Entity-Role-Participation-Act**  
Main : Archetype Entry Points

**Changelog 1.1.x up**

**3.0.5 (Athinda 2015-Oct-03)**  
CORE LOCATABLE removed, was obsolete, updated HL7 mapping  
Added "Joey" View

**3.0.4 (Patrick Langford 2015-Mar-23)**  
Changed 'function' to 'type'  
Corrected types

**3.0.3 (Patrick Langford 2015-Mar-02)**  
Added 'link' back into LOCATABLE

**3.0.2 (Deepak Sharma 2015-Jan-08)**  
Moved 'language' attribute from TEXT to PLAIN\_TEXT

**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  
IncludeLink 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-01)**  
Removed item\_id from CODED\_TEXT  
Updated all CODED\_TEXT documentation  
Added is\_code\_primitive type stereotype and stereotypes 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 (M7 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 we will 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 ENTRY, CLUSTER, SECTION, CONTAINER, etc. in Reference Archetypes.  
[CLINICAL\_DATA\_GROUP was discussed]

**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 on 23-may-2013)**  
• added LOCATABLE /instance\_id

**1.0.12b (telecon on 19-apr-2013) -> 1.0.0-dstu\_1**  
• LINKdetails[] 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

**1.0.5c > 1.0.6 2012-sep-20 2012-act**  
• 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 F2F Sunday 2012-sep-16**  
• ARCHETYPES.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 Type" > "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"  
• Action type "Character" renamed to "Character" to get consistent naming  
• Cleaned up some hidden (old) notes  
Issue#4: Templatized INTERVAL<T>  
• MULTIMEDIA.data & integrity> is\_im\_runtime and updated cardinality, container type "Array" and is collection  
• container type default to "List"

**referencemodel**  
**CIMI Reference Model**

- + Core
- + Data Value Types
- + Party
- + Primitive Types

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 model (i.e. archetype) 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 changelogs.

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 Hendl, Cecil Lynch, David Moner, Sarah Ryan and Harold Solbrig*. Additional clinical modeling 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 3/23/2015  
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'*

### Know 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*

### 3.0.5 (Atlanda 2015-Oct-03)

CORE LOCATABLE removed, was obsolete, updated HL7 mapping  
Added "Joey" View

### 3.0.4 (Patrick Langford 2015-Mar-23)

Changed 'function' to 'type'  
Corrected typos

### 3.0.3 (Patrick Langford 2015-Mar-02)

Added 'link' back into LOCATABLE

### 3.0.2 (Deepak Sharma 2015-Jan-08)

Moved 'language' attribute from TEXT to PLAIN\_TEXT

### 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\_type* 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 10/3/2015  
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, 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 3.0.4 Phase 1.0 dstu  
CIMI created on 9/14/2012. Last modified 4/22/2015  
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

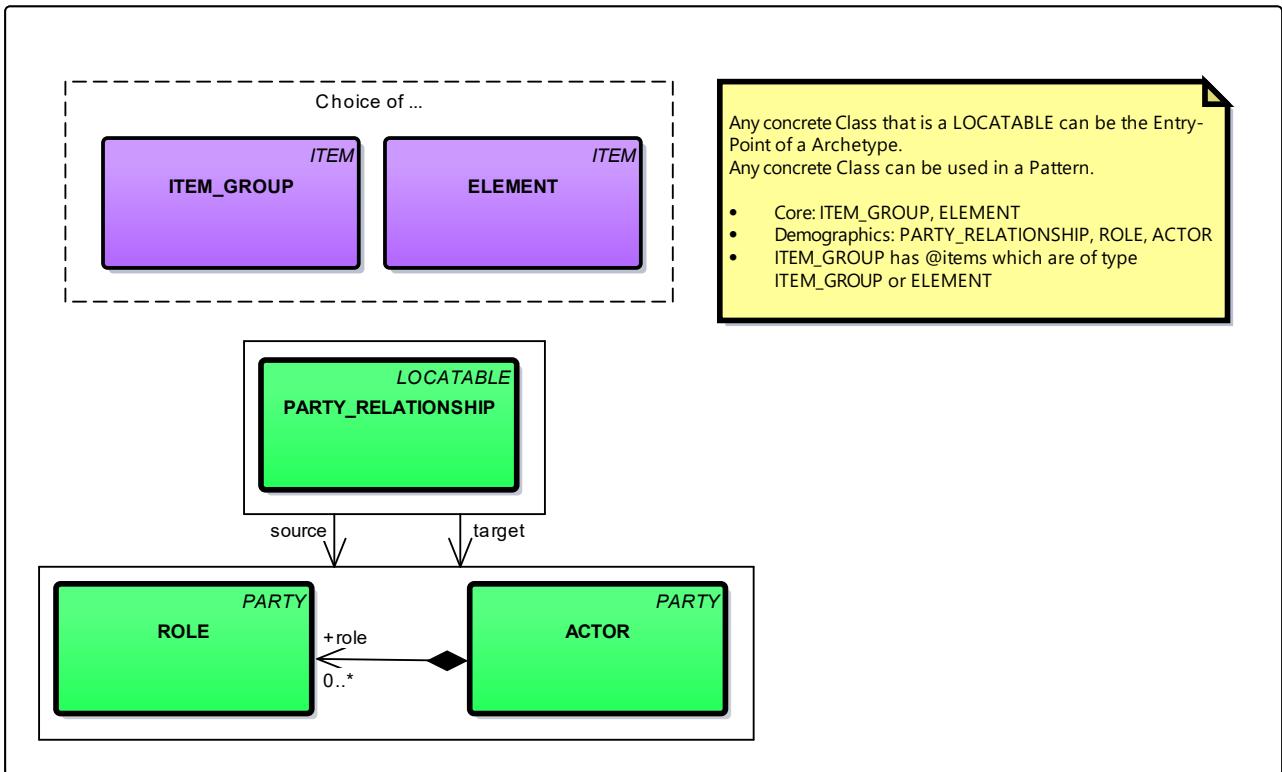


Figure 2: Archetype Entry Points

## <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 10/3/2015  
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 10/3/2015  
Extends ITEM

**OUTGOING STRUCTURAL RELATIONSHIPS** Generalization from ITEM\_GROUP to ITEM

[ Direction is 'Source -&gt; Destination'. ]

**INCOMING STRUCTURAL RELATIONSHIPS** Generalization from result to ITEM\_GROUP

[ Direction is 'Source -&gt; Destination'. ]

 Generalization from observable to ITEM\_GROUP

[ Direction is 'Source -&gt; Destination'. ]

 Generalization from timing to ITEM\_GROUP

[ Direction is 'Source -&gt; Destination'. ]

 Aggregation from PARTICIPATION to ITEM\_GROUP

[ Direction is 'Destination -&gt; 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: Private (Class) ITEM\_GROUP

Target: Public item (Class) ITEM  
Cardinality: [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 10/3/2015  
Extends PARTY

## OUTGOING STRUCTURAL RELATIONSHIPS

 Generalization from ACTOR to PARTY

[ Direction is 'Source -> Destination'. ]

## CONNECTORS

**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 10/3/2015  
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**

 details : ITEM Public

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

The detailed description of the relationship.

[ Is static False. Containment is Not Specified. ]

 type : CODED\_TEXT Public

The detailed description of the relationship

ATTRIBUTES	
[ 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 10/3/2015  
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..*]

## ASSOCIATIONS

## "All in one" View diagram

Class diagram in package 'Main'

"All in one" View  
Version 1.0

Michael van der Zel created on 10/3/2015. Last modified 10/5/2015

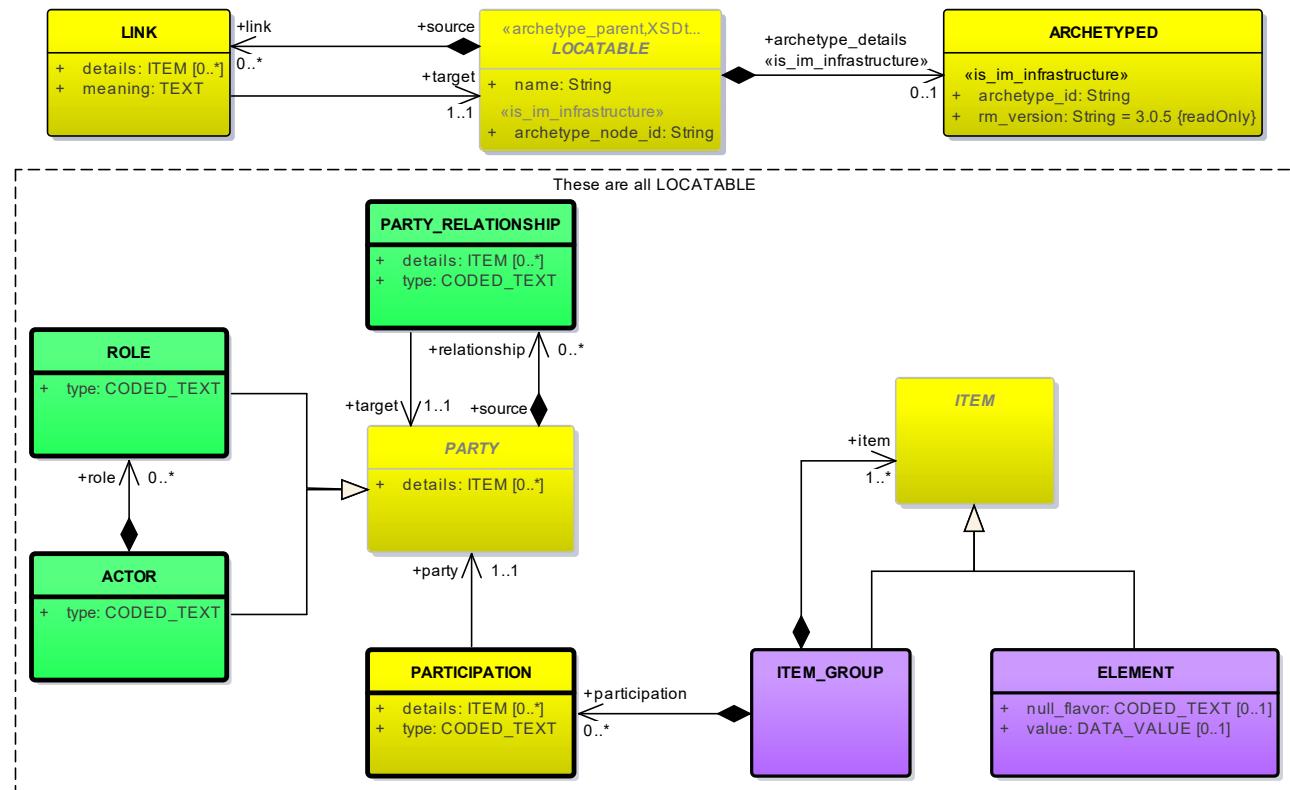


Figure 3: "All in one" View

## These are all LOCATABLE

Boundary in package 'Main'

These are all LOCATABLE

Version 1.0 Phase 1.0 Proposed

Michael van der Zel created on 10/3/2015. Last modified 10/3/2015

Extends

## 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 archetyping. 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 10/7/2015

## 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 = 3.0.5

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 10/3/2015  
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'. ]

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ 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 10/3/2015  
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'. ]

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

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

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 10/3/2015  
Extends ITEM

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

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

## INCOMING STRUCTURAL RELATIONSHIPS

➡ 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: Private (Class) ITEM\_GROUP

Target: Public item (Class) ITEM

Cardinality: [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 10/3/2015

## CONNECTORS

↗ Dependency Destination -> Source

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

## ATTRIBUTES

◆ details : ITEM Public

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

[ Is static False. Containment is Not Specified. ]

ATTRIBUTES	
 meaning : TEXT	Public

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) LOCATABLE «archetype_parent» Cardinality: [1..1]
 Association (direction: Source -> Destination)	Source: Public source (Class) LOCATABLE «archetype_parent»
	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 10/3/2015

**INCOMING STRUCTURAL RELATIONSHIPS**

---

- ➡ Generalization from ITEM to «archetype\_parent» LOCATABLE [ Direction is 'Source -> Destination'. ]
- ➡ Generalization from PARTICIPATION to «archetype\_parent» LOCATABLE [ Direction is 'Source -> Destination'. ]
- ➡ Generalization from PARTY\_RELATIONSHIP to «archetype\_parent» LOCATABLE [ Direction is 'Source -> Destination'. ]

## INCOMING STRUCTURAL RELATIONSHIPS

⇒ Generalization from PARTY 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]

↙ Association (direction: Source -> Destination)

Source: Public source (Class) LOCATABLE «archetype\_parent»

Target: Public link (Class) LINK  
Cardinality: [0..\*]

↙ 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) LOCATABLE

«archetype\_parent»

Cardinality: [1..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 10/3/2015  
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

◆ details : ITEM Public  
Multiplicity: ( [0..\*], Allow duplicates: 0, Is ordered: False )  
[ Is static False. Containment is Not Specified. ]

◆ type : CODED\_TEXT Public

The type of participation 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. ]

## ASSOCIATIONS

↗ Association (direction: Source -> Destination)

ASSOCIATIONS	
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 10/3/2015  
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

**ASSOCIATIONS**

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 10/3/2015  
 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]

ASSOCIATIONS	
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 10/3/2015  
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	
details : ITEM Public Multiplicity: ( [0..*], Allow duplicates: 0, Is ordered: False )	The detailed description of the relationship. [ Is static False. Containment is Not Specified. ]
type : CODED_TEXT Public 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 10/3/2015

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 minimalist 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*.

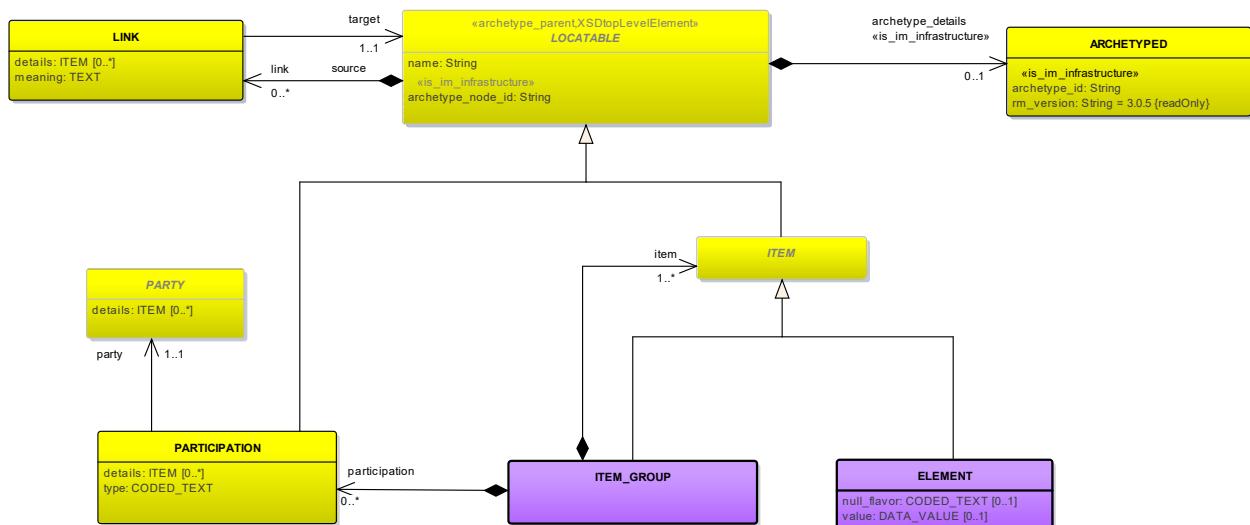
CIMI Reference Model  
Version 3.0.4 Phase 1.0 dstu  
CIMI created on 9/14/2012. Last modified 4/22/2015  
Alias RM

## 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.

CIMI Core Model  
Version 3.0.5  
CIMI MTF created on 5/21/2014. Last modified 10/7/2015



Name: CIMI Core Model  
Author: CIMI MTF  
Version: 3.0.5  
Created: 5/21/2014 12:00:00 AM  
Updated: 10/7/2015 1:56:08 PM

Figure 4: 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 10/7/2015

### 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 = 3.0.5

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 10/3/2015  
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 10/3/2015  
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 details to ITEM	[ Direction is 'Source -> Destination'. ]
⇒ Generalization from ITEM_GROUP to ITEM	[ Direction is 'Source -> Destination'. ]

CONNECTORS	
 <b>Dependency</b> Destination -> Source From: ITEM : Class, Public To: Act : Boundary, Public	

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 10/3/2015  
 Extends ITEM

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

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

INCOMING STRUCTURAL RELATIONSHIPS	
	[ 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: Private (Class) ITEM_GROUP	Target: Public item (Class) ITEM Cardinality: [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 10/3/2015

CONNECTORS	
↗ Dependency    Destination -> Source From: LINK : Class, Public To: ActRelationship : Boundary, Public	

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

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

ASSOCIATIONS
<p>✍ Association (direction: Source -&gt; Destination)</p> <p>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.</p>
<p>Source: Public (Class) LINK</p> <p>Target: Public target (Class) LOCATABLE      «archetype_parent»      Cardinality: [1..1]</p>
<p>✍ Association (direction: Source -&gt; Destination)</p> <p>Source: Public source (Class) LOCATABLE «archetype_parent»</p> <p>Target: Public link (Class) LINK      Cardinality: [0..*]</p>

## 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 10/3/2015

INCOMING STRUCTURAL RELATIONSHIPS
<p>⇒ Generalization from ITEM to «archetype_parent» LOCATABLE</p> <p>[ Direction is 'Source -&gt; Destination'. ]</p>
<p>⇒ Generalization from PARTICIPATION to «archetype_parent» LOCATABLE</p> <p>[ Direction is 'Source -&gt; Destination'. ]</p>
<p>⇒ Generalization from PARTY_RELATIONSHIP to «archetype_parent» LOCATABLE</p>

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

CONNECTORS
<p>↗ Trace «trace» Source -&gt; Destination            From: id : Property, Public            To: LOCATABLE : Class, Public</p>
<p>↗ Trace «trace» Source -&gt; Destination            From: name : Property, Public            To: LOCATABLE : Class, Public</p>

ATTRIBUTES
<p>◆ 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.</p> <p>[ Stereotype is «is_im_infrastructure». Is static False. Containment is Not Specified. ]</p>
<p>◆ 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.</p> <p>[ Is static False. Containment is Not Specified. ]</p>

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

## ASSOCIATIONS

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) LOCATABLE  
«archetype\_parent»  
Cardinality: [1..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 10/3/2015  
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

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

[ Is static False. Containment is Not Specified. ]

◆ type : CODED\_TEXT Public

The type of participation 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. ]

## ASSOCIATIONS

## 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 10/3/2015  
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)

<b>ASSOCIATIONS</b>	
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 3.0.5  
CIMI MTF created on 3/22/2013. Last modified 10/7/2015

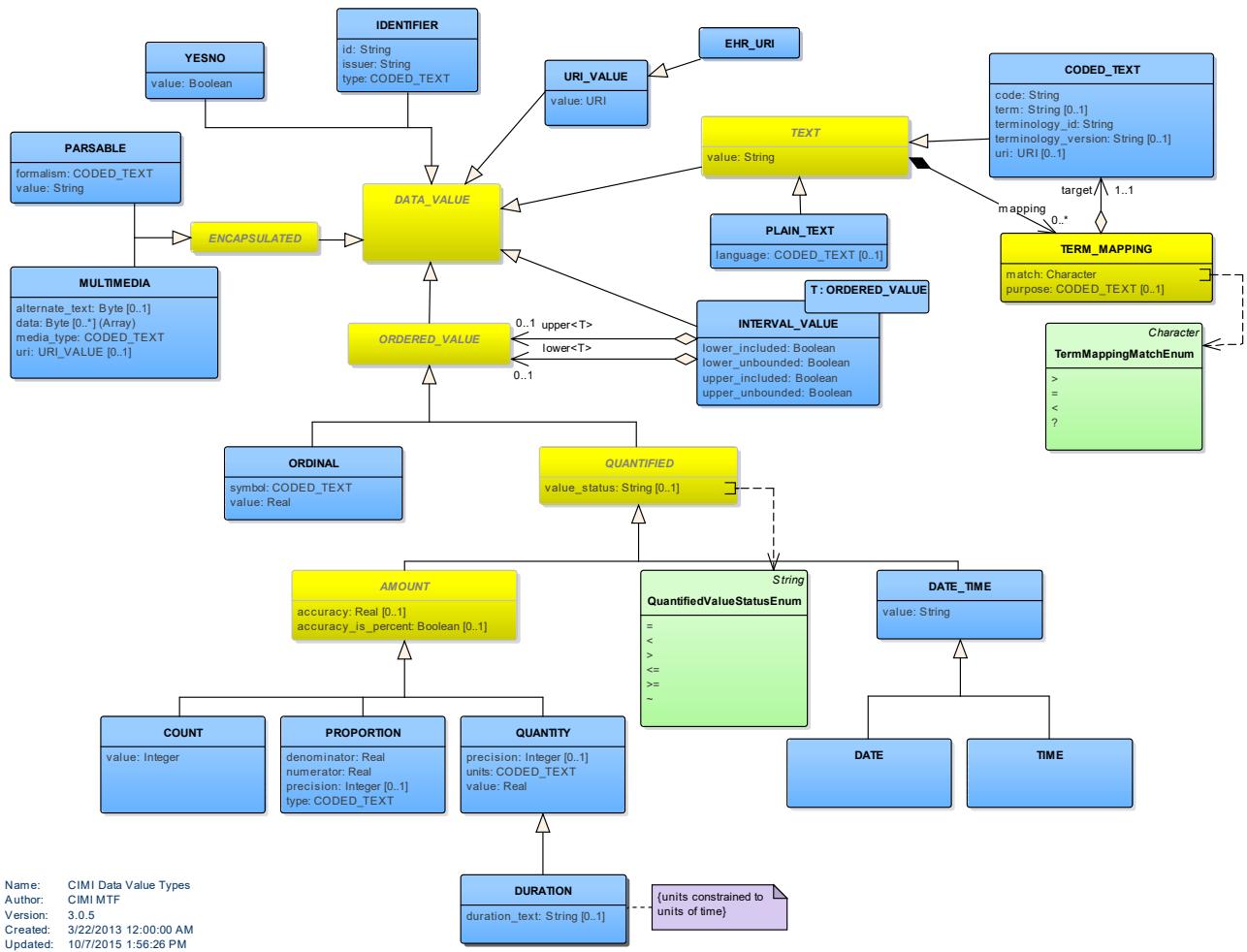


Figure 5: CIMI Data Value Types

## 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'. ]

INCOMING STRUCTURAL RELATIONSHIPS	
⇒ Generalization from PROPORTION to AMOUNT	[ Direction is 'Source -> Destination'. ]
⇒ Generalization from QUANTITY 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	
code : String Public	
A code that uniquely identifies the referenced concept within the context of <i>terminology_id</i> . Examples: 74400008, 2951-2, de	[ Is static False. Containment is Not Specified. ]

ATTRIBUTES
<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' [ 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 [ Is static False. Containment is Not Specified. ]</p>
<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/, <a href="http://loinc.org/246">http://loinc.org/246</a>, http://id.loc.gov/vocabulary/iso639-1 [ Is static False. Containment is Not Specified. ]</p>
<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, <a href="http://loinc.org/id/2951-2">http://loinc.org/id/2951-2</a>, http://id.loc.gov/vocabulary/iso639-1/de [ Is static False. Containment is Not Specified. ]</p>

ASSOCIATIONS
<p>◆ Association (direction: Source -&gt; 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

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 TEXT to DATA_VALUE	[ Direction is 'Source -> Destination'. ]
⇒ Generalization from ENCAPSULATED to DATA_VALUE	[ Direction is 'Source -> Destination'. ]
⇒ Generalization from ORDERED_VALUE to DATA_VALUE	[ Direction is 'Source -> Destination'. ]
⇒ Generalization from YESNO to DATA_VALUE	[ Direction is 'Source -> Destination'. ]
⇒ Generalization from INTERVAL_VALUE to DATA_VALUE	[ Direction is 'Source -> Destination'. ]
⇒ Generalization from IDENTIFIER to DATA_VALUE	[ Direction is 'Source -> Destination'. ]
⇒ Generalization from URI_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» TIME to «is\_adl\_primitivetype» DATE\_TIME
 

[ Direction is 'Source -> Destination'. ]
- ⇒ Generalization from «is\_adl\_primitivetype» DATE 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

⬅ 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. ]

◆ 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. ]

◆ type : CODED\_TEXT Public

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

**ATTRIBUTES**

[ 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'. ]

⇒ 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'. ]

**ATTRIBUTES**

◆ lower\_included : 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. ]

ATTRIBUTES	
◆ upper_unbounded : Boolean Public Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )	[ 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

⬅ Generalization from ORDERED\_VALUE to DATA\_VALUE

[ Direction is 'Source -> Destination'. ]

⬅ Aggregation from ORDERED\_VALUE to INTERVAL\_VALUE

[ Direction is 'Destination -> Source'. ]

⬅ Aggregation from ORDERED\_VALUE to INTERVAL\_VALUE

[ Direction is 'Destination -> Source'. ]

#### INCOMING STRUCTURAL RELATIONSHIPS

➡ Generalization from QUANTIFIED to ORDERED\_VALUE

[ Direction is 'Source -> Destination'. ]

➡ Generalization from ORDINAL 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
<p>◆ symbol : CODED_TEXT Public</p> <p>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 .</p> <p>[ Is static False. Containment is Not Specified. ]</p>
<p>◆ value : Real Public</p> <p>Value in ordered enumeration of values. Any integer value can be used.</p> <p>[ Is static False. Containment is Not Specified. ]</p>

## 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
<p>◀ Generalization from PARSABLE to ENCAPSULATED</p> <p>[ Direction is 'Source -&gt; Destination'. ]</p>

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

## ATTRIBUTES

◆ 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 . ]

## 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

◆ denominator : Real Public

The denominator of the ratio.

[ Is static False. Containment is Not Specified. ]

◆ numerator : Real Public

The numerator of the ratio.

[ Is static False. Containment is Not Specified. ]

◆ precision : Integer Public

ATTRIBUTES	
Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )	
<p>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.</p> <p>[ Is static False. Containment is Not Specified. ]</p>	
 type : CODED_TEXT Public <p>Indicates the semantic type of the proportion. Valid values include: ratio, unitary, percent, fraction, integer_fraction</p> <p>[ Is static False. Containment is Not Specified. ]</p>	

## 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	
 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 )	
<p>The optional status of the magnitude with possible values:</p> <p>= : magnitude is a point value      &lt; : value is &lt; magnitude      &gt; : value is &gt; magnitude</p>	

**ATTRIBUTES**

$\leq$  : value is  $\leq$  magnitude  
 $\geq$  : value is  $\geq$  magnitude  
 $\sim$  : 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. ]
-   **$\leq$**  : Public  
[ Stereotype is «enum». Is static False. Containment is Not Specified. ]
-   **$\geq$**  : Public  
[ Stereotype is «enum». Is static False. Containment is Not Specified. ]
-   **$\sim$**  : 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

◆ 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. ]

◆ units : CODED\_TEXT Public

[ Is static False. Containment is Not Specified. ]

◆ value : Real Public

The numeric size of the quantity.

[ 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

## 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

## 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

 Generalization from «is\_adl\_primitivetype» CODED\_TEXT to TEXT

[ Direction is 'Source -> Destination'. ]

 Aggregation from TERM\_MAPPING to TEXT

[ Direction is 'Destination -> Source'. ]

## INCOMING STRUCTURAL RELATIONSHIPS

➡ Generalization from PLAIN\_TEXT to TEXT

[ Direction is 'Source -> Destination'. ]

## ATTRIBUTES

◆ value : String Public

[ Is static False. Containment is Not Specified. ]

## 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.

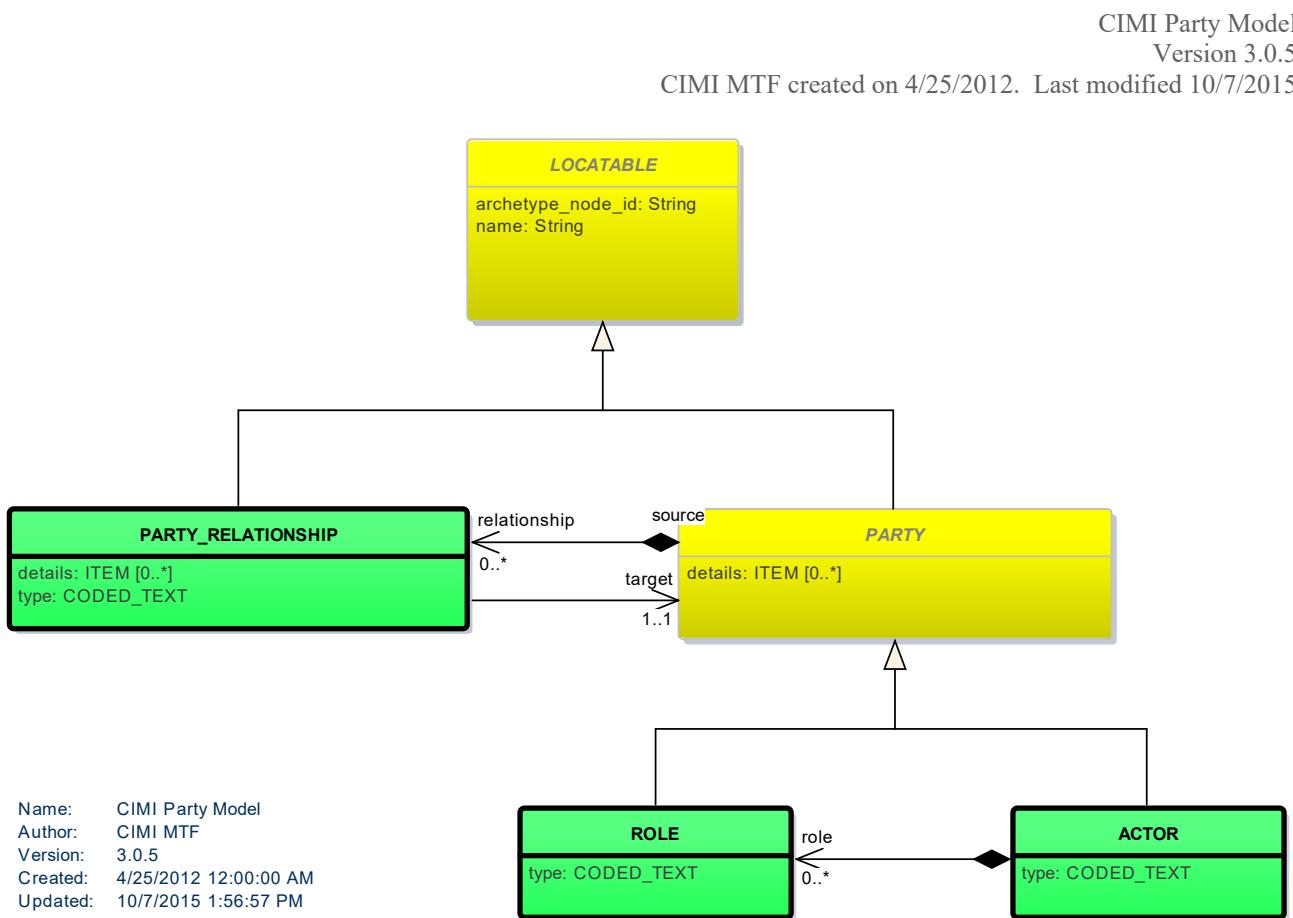


Figure 6: CIMI Party Model

## 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 10/3/2015

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

## INCOMING STRUCTURAL RELATIONSHIPS

➡ Generalization from PARTY 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]

↙ Association (direction: Source -> Destination)

Source: Public source (Class) LOCATABLE «archetype\_parent»

Target: Public link (Class) LINK

Cardinality: [0..\*]

↙ 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) LOCATABLE  
«archetype\_parent»

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 10/3/2015  
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

## 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 10/3/2015  
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)

<b>ASSOCIATIONS</b>	
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 10/3/2015  
Extends LOCATABLE

<b>OUTGOING STRUCTURAL RELATIONSHIPS</b>	
Generalization from PARTY_RELATIONSHIP to «archetype_parent» LOCATABLE	[ Direction is 'Source -> Destination'. ]

<b>CONNECTORS</b>	
Dependency    Destination -> Source From:    PARTY_RELATIONSHIP : Class, Public To:       RoleLink : Boundary, Public	

<b>ATTRIBUTES</b>	
details : ITEM Public Multiplicity: ( [0..*], Allow duplicates: 0, Is ordered: False )	
The detailed description of the relationship.	[ Is static False. Containment is Not Specified. ]
type : CODED_TEXT Public The detailed description of the relationship	[ Is static False. Containment is Not Specified. ]

<b>ASSOCIATIONS</b>	
Association (direction: Source -> Destination)	Source: Public (Class) PARTY_RELATIONSHIP Target: Public target (Class) PARTY Cardinality: [1..1]

## ASSOCIATIONS

 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 10/3/2015

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

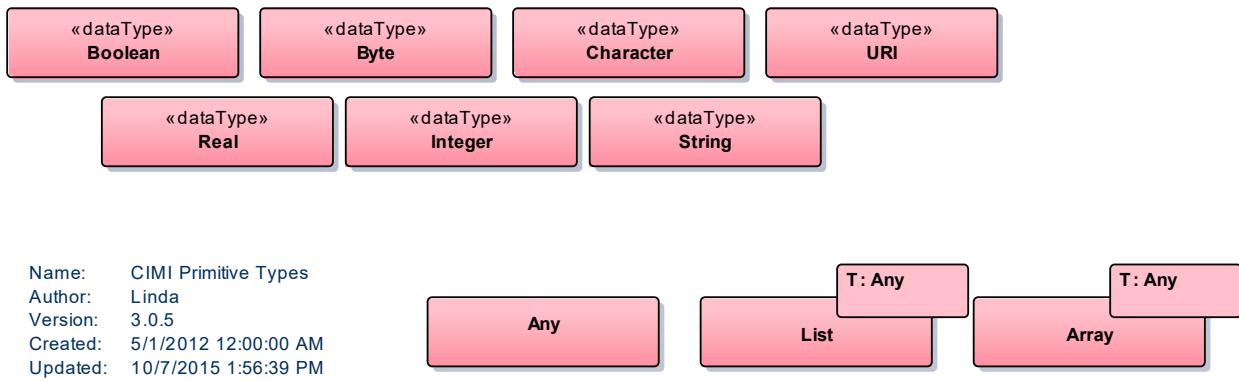
zelm created on 3/22/2013. Last modified 3/22/2013

## 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 3.0.5  
Linda created on 5/1/2012. Last modified 10/7/2015



Name: CIMI Primitive Types  
 Author: Linda  
 Version: 3.0.5  
 Created: 5/1/2012 12:00:00 AM  
 Updated: 10/7/2015 1:56:39 PM

Figure 7: 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 10/7/2015

### 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 = 3.0.5

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 10/3/2015  
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 10/3/2015  
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 details to ITEM	[ Direction is 'Source -> Destination'. ]
⇒ Generalization from ITEM_GROUP to ITEM	[ Direction is 'Source -> Destination'. ]

CONNECTORS	
 <b>Dependency</b> Destination -> Source From: ITEM : Class, Public To: Act : Boundary, Public	

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 10/3/2015  
 Extends ITEM

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

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

INCOMING STRUCTURAL RELATIONSHIPS	
	[ 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: Private (Class) ITEM_GROUP	Target: Public item (Class) ITEM Cardinality: [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 10/3/2015

CONNECTORS	
↗ Dependency    Destination -> Source From: LINK : Class, Public To: ActRelationship : Boundary, Public	

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

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

ASSOCIATIONS
<p>✍ Association (direction: Source -&gt; Destination)</p> <p>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.</p>
<p>Source: Public (Class) LINK</p> <p>Target: Public target (Class) LOCATABLE      «archetype_parent»      Cardinality: [1..1]</p>
<p>✍ Association (direction: Source -&gt; Destination)</p> <p>Source: Public source (Class) LOCATABLE «archetype_parent»</p> <p>Target: Public link (Class) LINK      Cardinality: [0..*]</p>

## 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 10/3/2015

INCOMING STRUCTURAL RELATIONSHIPS
<p>⇒ Generalization from ITEM to «archetype_parent» LOCATABLE</p> <p>[ Direction is 'Source -&gt; Destination'. ]</p>
<p>⇒ Generalization from PARTICIPATION to «archetype_parent» LOCATABLE</p> <p>[ Direction is 'Source -&gt; Destination'. ]</p>
<p>⇒ Generalization from PARTY_RELATIONSHIP to «archetype_parent» LOCATABLE</p>

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

CONNECTORS
<p>↗ Trace «trace» Source -&gt; Destination            From: id : Property, Public            To: LOCATABLE : Class, Public</p>
<p>↗ Trace «trace» Source -&gt; Destination            From: name : Property, Public            To: LOCATABLE : Class, Public</p>

ATTRIBUTES
<p>◆ 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.</p> <p>[ Stereotype is «is_im_infrastructure». Is static False. Containment is Not Specified. ]</p>
<p>◆ 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.</p> <p>[ Is static False. Containment is Not Specified. ]</p>

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

## ASSOCIATIONS

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) LOCATABLE  
«archetype\_parent»  
Cardinality: [1..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 10/3/2015  
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

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

[ Is static False. Containment is Not Specified. ]

◆ type : CODED\_TEXT Public

The type of participation 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. ]

## ASSOCIATIONS

**ASSOCIATIONS**

 Association (direction: Source -> Destination)

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 PROPORTION to AMOUNT

[ Direction is 'Source -> Destination'. ]

➡ Generalization from QUANTITY 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

**ATTRIBUTES**

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**

◆ **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. ]

◆ **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. ]

◆ **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. ]

ATTRIBUTES	
◆ terminology_version : String Public	Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )
<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: <a href="http://snomed.info/sct/900000000000207008">http://snomed.info/sct/900000000000207008</a>, <a href="http://snomed.info/sct/900000000000207008/version/20130731">http://snomed.info/sct/900000000000207008/version/20130731</a>, <a href="http://loinc.org/">http://loinc.org/</a>, <a href="http://loinc.org/246">http://loinc.org/246</a>, <a href="http://id.loc.gov/vocabulary/iso639-1">http://id.loc.gov/vocabulary/iso639-1</a></p>	
	[ Is static False. Containment is Not Specified. ]
◆ uri : URI Public	Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )
<p>A URI that uniquely identifies the referenced concept. Examples: <a href="http://snomed.info/id/74400008">http://snomed.info/id/74400008</a>, <a href="http://loinc.org/id/2951-2">http://loinc.org/id/2951-2</a>, <a href="http://id.loc.gov/vocabulary/iso639-1/de">http://id.loc.gov/vocabulary/iso639-1/de</a></p>	
	[ 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

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

<b>OUTGOING STRUCTURAL RELATIONSHIPS</b>	
⇐ 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» TIME to «is\_adl\_primitivetype» DATE\_TIME  
[ Direction is 'Source -> Destination'. ]
- ➡ Generalization from «is\_adl\_primitivetype» DATE 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

ATTRIBUTES
<p>duration_text : String Public Multiplicity: ( [0..1], Allow duplicates: 0, Is ordered: False )</p> <p>A string representing an ISO8601 duration. [ Is static False. Containment is Not Specified. ]</p>

OPERATIONS
<p>is_strictly_comparable_to (other : ) : Public [ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]</p>

## 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
<p>↳ Generalization from EHR_URI to URI_VALUE [ Direction is 'Source -&gt; Destination'. ]</p>

## 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
<p>↳ Generalization from ENCAPSULATED to DATA_VALUE [ Direction is 'Source -&gt; Destination'. ]</p>

INCOMING STRUCTURAL RELATIONSHIPS
<p>⇒ Generalization from PARSABLE to ENCAPSULATED [ Direction is 'Source -&gt; Destination'. ]</p>

## 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. ]

◆ 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. ]

◆ type : CODED\_TEXT Public

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

[ 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'. ]

➡ 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'. ]

#### ATTRIBUTES

◆ lower\_included : 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. ]

◆ upper\_unbounded : 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

⬅ Generalization from ORDERED\_VALUE to DATA\_VALUE

[ Direction is 'Source -> Destination'. ]

## 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'. ]

## INCOMING STRUCTURAL RELATIONSHIPS

⇒ Generalization from QUANTIFIED to ORDERED\_VALUE

[ Direction is 'Source -> Destination'. ]

⇒ Generalization from ORDINAL 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. ]

**ATTRIBUTES**

◆ 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'. ]

## ATTRIBUTES

- ❖ 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 . ]

## 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

- ❖ denominator : Real Public

The denominator of the ratio.

[ Is static False. Containment is Not Specified. ]

- ❖ numerator : Real Public

The numerator 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

**ATTRIBUTES**

[ 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**

↗ 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

⬅ 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

◆ 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. ]

◆ units : CODED\_TEXT Public

[ Is static False. Containment is Not Specified. ]

◆ value : Real Public

The numeric size of the quantity.

[ 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

CONNECTORS	
 <b>Dependency</b>	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. ]
 >= : 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	

## 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]

## 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	
⇒ 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. ]

## 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	

**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 10/3/2015  
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 10/3/2015  
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)

<b>ASSOCIATIONS</b>	
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 10/3/2015  
Extends LOCATABLE

<b>OUTGOING STRUCTURAL RELATIONSHIPS</b>	
Generalization from PARTY_RELATIONSHIP to «archetype_parent» LOCATABLE	[ Direction is 'Source -> Destination'. ]

<b>CONNECTORS</b>	
Dependency    Destination -> Source From:    PARTY_RELATIONSHIP : Class, Public To:       RoleLink : Boundary, Public	

<b>ATTRIBUTES</b>	
details : ITEM Public Multiplicity: ( [0..*], Allow duplicates: 0, Is ordered: False )	
The detailed description of the relationship.	[ Is static False. Containment is Not Specified. ]
type : CODED_TEXT Public The detailed description of the relationship	[ Is static False. Containment is Not Specified. ]

<b>ASSOCIATIONS</b>	
Association (direction: Source -> Destination)	Source: Public (Class) PARTY_RELATIONSHIP Target: Public target (Class) PARTY Cardinality: [1..1]

## ASSOCIATIONS

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 10/3/2015

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 10/3/2015

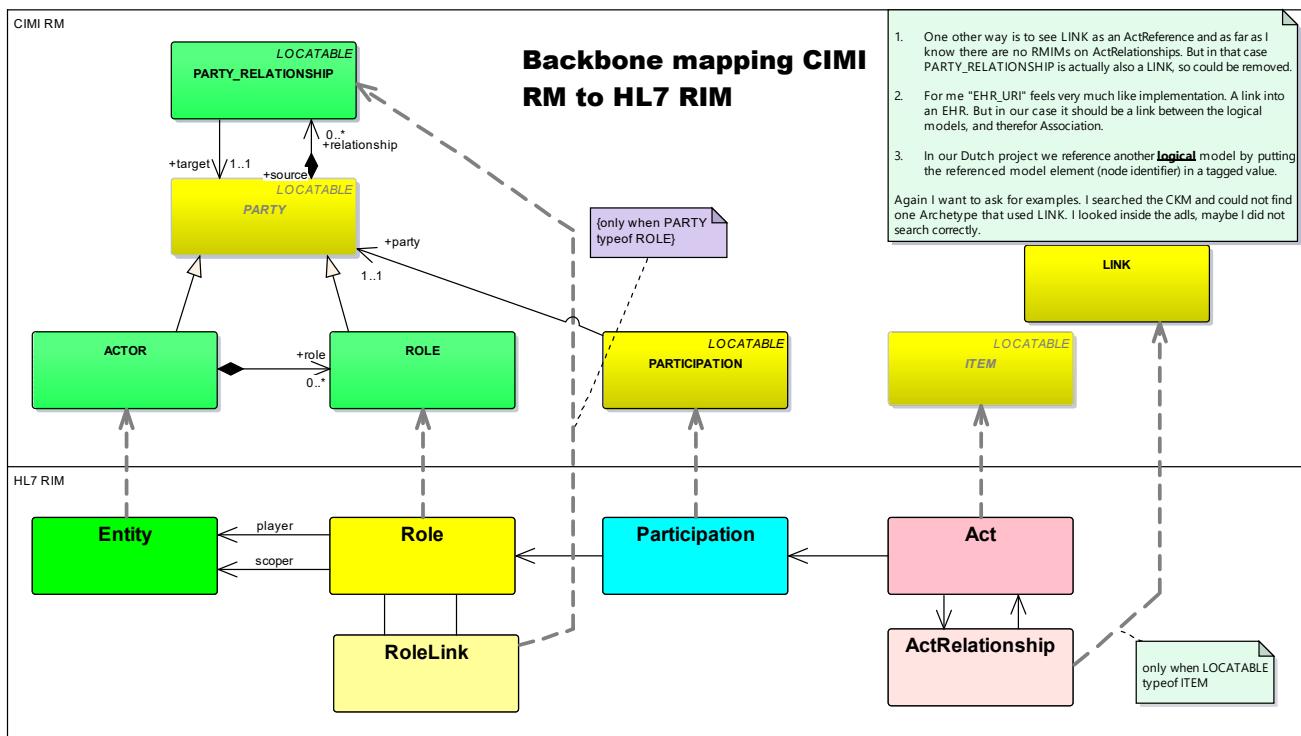


Figure 8: HL7 Entity-Role-Participation-Act

## 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 10/3/2015  
Extends LOCATABLE

OUTGOING STRUCTURAL RELATIONSHIPS
<p>↳ Generalization from ITEM to «archetype_parent» LOCATABLE</p> <p>[ Direction is 'Source -&gt; Destination'. ]</p>

## INCOMING STRUCTURAL RELATIONSHIPS

⇒ Generalization from ELEMENT to ITEM

[ Direction is 'Source -> Destination'. ]

⇒ Generalization from details to ITEM

[ Direction is 'Source -> Destination'. ]

⇒ Generalization from ITEM\_GROUP to ITEM

[ Direction is 'Source -> Destination'. ]

## CONNECTORS

↗ Dependency    Destination -> Source

From:    ITEM : Class, Public

To:       Act : Boundary, Public

## ASSOCIATIONS

↙ Association (direction: Source -> Destination)

Source: Private (Class) ITEM\_GROUP

Target: Public item (Class) ITEM

Cardinality: [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 10/3/2015

## CONNECTORS

↗ Dependency    Destination -> Source

From:    LINK : Class, Public

To:       ActRelationship : Boundary, Public

## ATTRIBUTES

ATTRIBUTES	
◆ details : ITEM Public	Multiplicity: ( [0..*], Allow duplicates: 0, Is ordered: False )
	[ Is static False. Containment is Not Specified. ]
◆ meaning : TEXT Public	
<p>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 .</p> <p>[ Is static False. Containment is . ]</p>	
ASSOCIATIONS	
<p>✍ Association (direction: Source -&gt; Destination)</p> <p>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.</p>	
Source: Public (Class) LINK	Target: Public target (Class) LOCATABLE «archetype_parent» Cardinality: [1..1]
<p>✍ Association (direction: Source -&gt; Destination)</p> <p>Source: Public source (Class) LOCATABLE «archetype_parent»</p>	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 10/3/2015  
 Extends LOCATABLE

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

## OUTGOING STRUCTURAL RELATIONSHIPS

 Aggregation from PARTICIPATION to ITEM\_GROUP

[ Direction is 'Destination -> Source'. ]

## CONNECTORS

 **Dependency** Destination -> Source

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

## ATTRIBUTES

 details : ITEM Public

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

[ Is static False. Containment is Not Specified. ]

 type : CODED\_TEXT Public

The type of participation 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. ]

## 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 10/3/2015  
 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 10/3/2015  
 Extends LOCATABLE

## OUTGOING STRUCTURAL RELATIONSHIPS

 Generalization from PARTY to «archetype\_parent» LOCATABLE

[ Direction is 'Source -> Destination'. ]

## INCOMING STRUCTURAL RELATIONSHIPS

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 10/3/2015  
Extends LOCATABLE

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

CONNECTORS
------------

## CONNECTORS

 **Dependency** Destination -> Source  
 From: PARTY\_RELATIONSHIP : Class, Public  
 To: RoleLink : Boundary, Public

## ATTRIBUTES

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

The detailed description of the relationship.

[ Is static False. Containment is Not Specified. ]

 type : CODED\_TEXT Public

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 10/3/2015  
 Extends PARTY

## OUTGOING STRUCTURAL RELATIONSHIPS

 Generalization from ROLE to PARTY

[ Direction is 'Source -> Destination'. ]

## CONNECTORS

**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: ITEM : 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

 **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
<p> <b>Dependency</b> Destination -&gt; Source        From: PARTY_RELATIONSHIP : Class, Public        To: RoleLink : Boundary, Public</p>

## <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

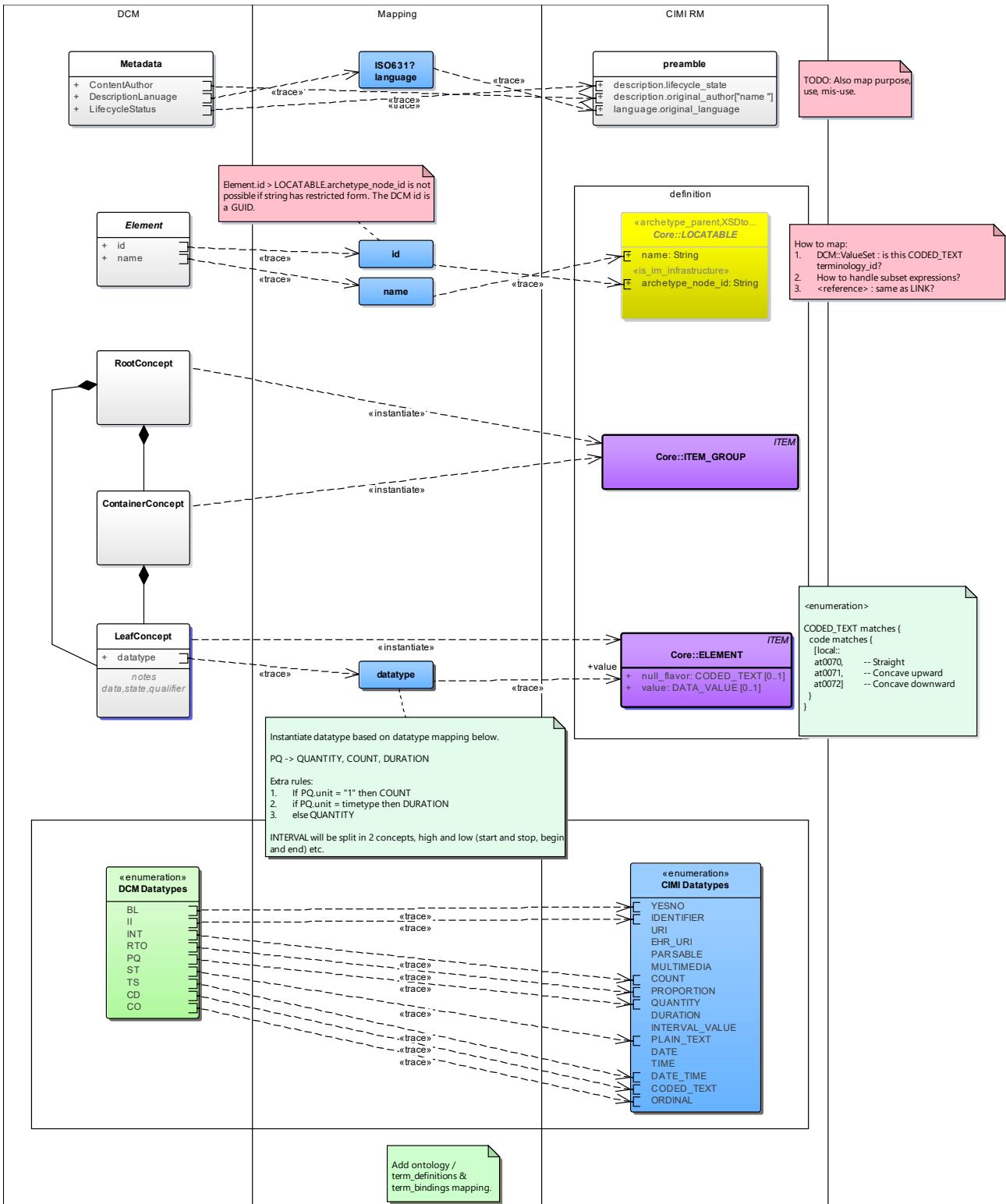


Figure 9: Mapping DCM 2 CIMI RM

## 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 10/3/2015  
Extends ITEM

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

<b>INCOMING STRUCTURAL RELATIONSHIPS</b>	
⇒ 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'. ]

<b>CONNECTORS</b>	
↗ 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	

<b>ATTRIBUTES</b>	
◆ 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 10/3/2015  
Extends ITEM

<b>OUTGOING STRUCTURAL RELATIONSHIPS</b>
--

## OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Generalization from ITEM\_GROUP to ITEM

[ Direction is 'Source -> Destination'. ]

## INCOMING STRUCTURAL RELATIONSHIPS

➡ Generalization from result to ITEM\_GROUP

[ Direction is 'Source -> Destination'. ]

➡ Generalization from observable 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: Private (Class) ITEM\_GROUP

Target: Public item (Class) ITEM

Cardinality: [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 10/3/2015

## INCOMING STRUCTURAL RELATIONSHIPS

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

<b>ASSOCIATIONS</b>	
Association (direction: Source -> Destination)	
Source: Public source (Class) LOCATABLE «archetype_parent»	Target: Public link (Class) LINK Cardinality: [0..*]
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) LOCATABLE «archetype_parent» Cardinality: [1..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>  
Version 1.0 Phase 1.0 Proposed  
ZelM created on 6/5/2012. Last modified 4/11/2013  
Extends

## <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
 <b>Trace</b> «trace» Source -> Destination From: DCM Datatypes : Enumeration, Public To: CIMI Datatypes : Enumeration, Public
 <b>Trace</b> «trace» Source -> Destination From: DCM Datatypes : Enumeration, Public To: CIMI Datatypes : Enumeration, Public
 <b>Trace</b> «trace» Source -> Destination From: DCM Datatypes : Enumeration, Public To: CIMI Datatypes : Enumeration, Public
 <b>Trace</b> «trace» Source -> Destination From: DCM Datatypes : Enumeration, Public To: CIMI Datatypes : Enumeration, Public
 <b>Trace</b> «trace» Source -> Destination

CONNECTORS	
From:	DCM Datatypes : Enumeration, Public
To:	CIMI Datatypes : Enumeration, Public
 <b>Trace</b> «trace»    Source -> Destination	
From:	DCM Datatypes : Enumeration, Public
To:	CIMI Datatypes : Enumeration, Public
 <b>Trace</b> «trace»    Source -> Destination	
From:	DCM Datatypes : Enumeration, Public
To:	CIMI Datatypes : Enumeration, Public
 <b>Trace</b> «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. ]
 QUANTITY : Public	[ Stereotype is «enum». Is static False. Containment is Not Specified. ]
 DURATION : Public	[ Stereotype is «enum». Is static False. Containment is Not Specified. ]

ATTRIBUTES	
◆ 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

 **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. ]

ATTRIBUTES	
◆ 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	
◆ 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	

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

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

ATTRIBUTES
<p> ContentAuthor : Public        [ Is static False. Containment is Not Specified. ]</p>
<p> DescriptionLanguage : Public        [ Is static False. Containment is Not Specified. ]</p>
<p> LifecycleStatus : Public        [ Is static False. Containment is Not Specified. ]</p>

## 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
<p> Aggregation from LeafConcept to RootConcept        [ Direction is 'Source -&gt; Destination'. ]</p>
<p> Aggregation from ContainerConcept to RootConcept        [ Direction is 'Source -&gt; Destination'. ]</p>

CONNECTORS
<p> <b>Dependency</b> «instantiate»    Source -&gt; 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

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

From:    Metadata : Class, Public  
To:       preamble : Class, Public

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

From:    ISO631? language : Property, Public  
To:       preamble : Class, Public

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

From:    Metadata : Class, Public  
To:       preamble : Class, Public

### ATTRIBUTES

◆ description.lifecycle\_state :    Public

[ Is static False. Containment is Not Specified. ]

◆ description.original\_author["name "] :    Public

[ Is static False. Containment is Not Specified. ]

◆ language.original\_language :    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

↗ **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

**CONNECTORS**

 **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

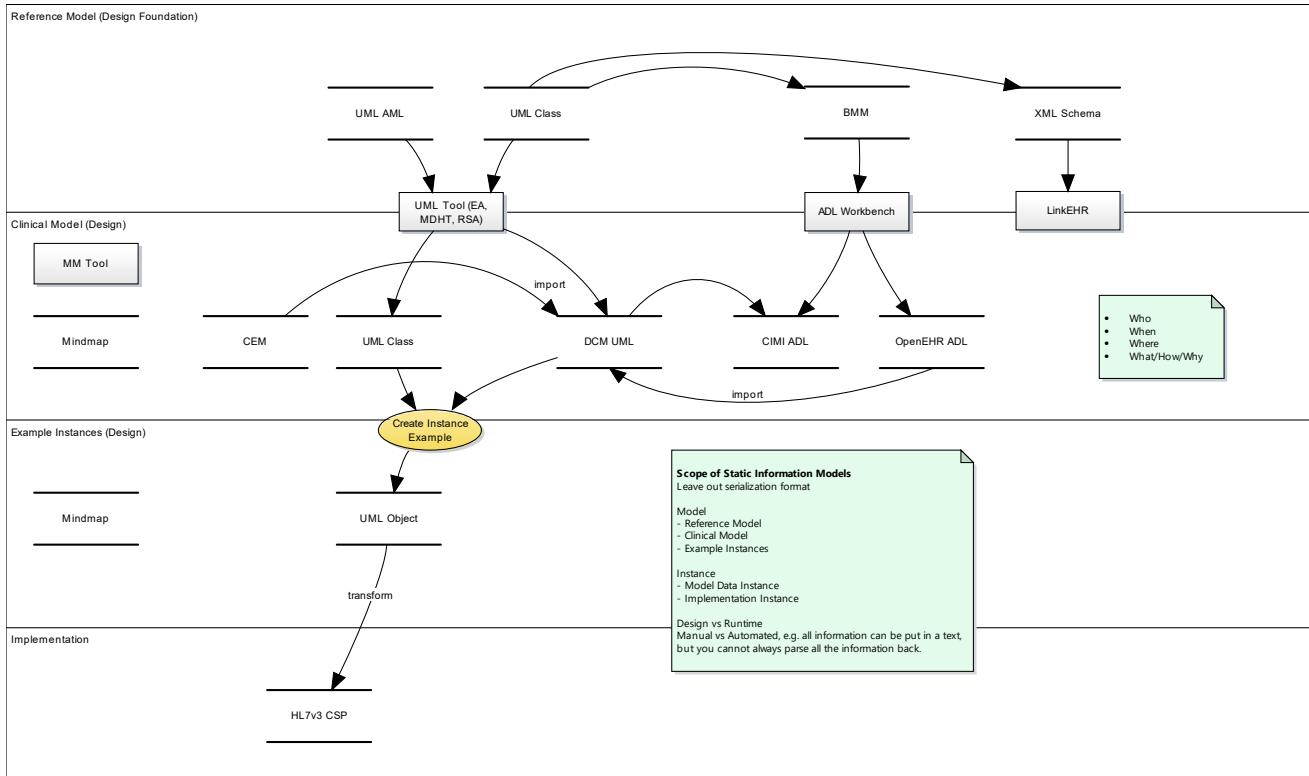


Figure 10: Model Transformation Map

## <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> <b>DataFlow</b> «DFD_DataFlow»    Source -&gt; Destination            From:    ADL Workbench : External, Public            To:      OpenEHR ADL : DataStore, Public</p>
<p> <b>DataFlow</b> «DFD_DataFlow»    Source -&gt; Destination            From:    ADL Workbench : External, Public            To:      CIMI ADL : DataStore, Public</p>
<p> <b>DataFlow</b> «DFD_DataFlow»    Source -&gt; Destination            From:    BMM : DataStore, Public            To:      ADL Workbench : External, Public</p>

## **BMM**

*DataStore «DFD\_DataStore» in package 'Model Transformation Map'*

BMM

## 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

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

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

 **DataFlow** «DFD\_DataFlow»   Source -> Destination  
 From: UML AML : 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

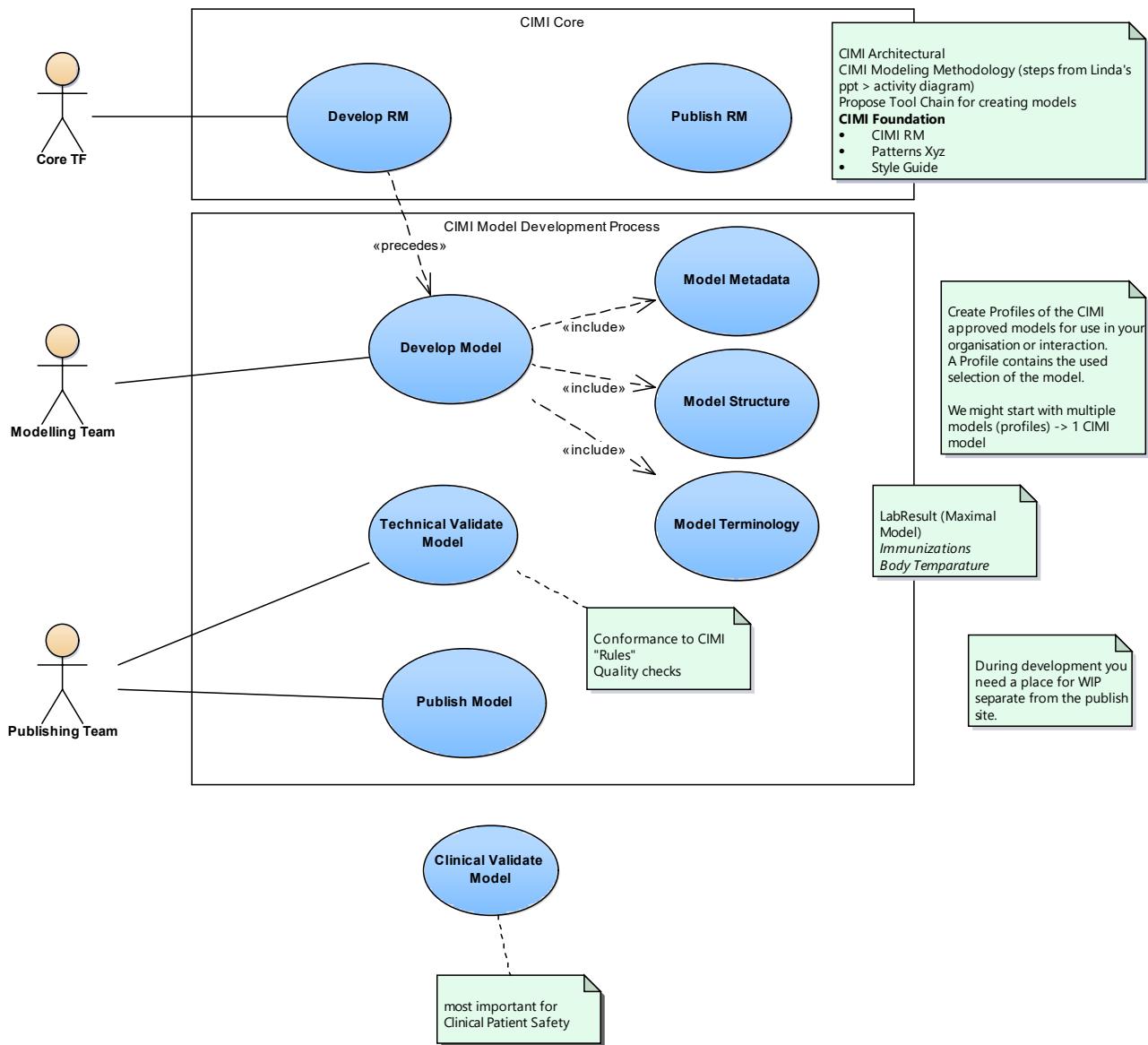


Figure 11: Tool Chain

## 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: Publish Model : UseCase, Public

 **UseCaseLink**    Source -> Destination  
 From: Publishing Team : Actor, Public  
 To: Technical Validate 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	
 <b>Include</b> «include»	Source -> Destination
From:	Develop Model : UseCase, Public
To:	Model Metadata : UseCase, Public
 <b>Include</b> «include»	Source -> Destination
From:	Develop Model : UseCase, Public
To:	Model Terminology : UseCase, Public
 <b>Include</b> «include»	Source -> Destination
From:	Develop Model : UseCase, Public
To:	Model Structure : UseCase, Public
 <b>Dependency</b> «precedes»	Source -> Destination
From:	Develop RM : UseCase, Public
To:	Develop Model : UseCase, Public
 <b>UseCaseLink</b>	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	
 <b>Dependency</b> «precedes»	Source -> Destination
From:	Develop RM : UseCase, Public
To:	Develop Model : UseCase, Public
 <b>UseCaseLink</b>	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



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

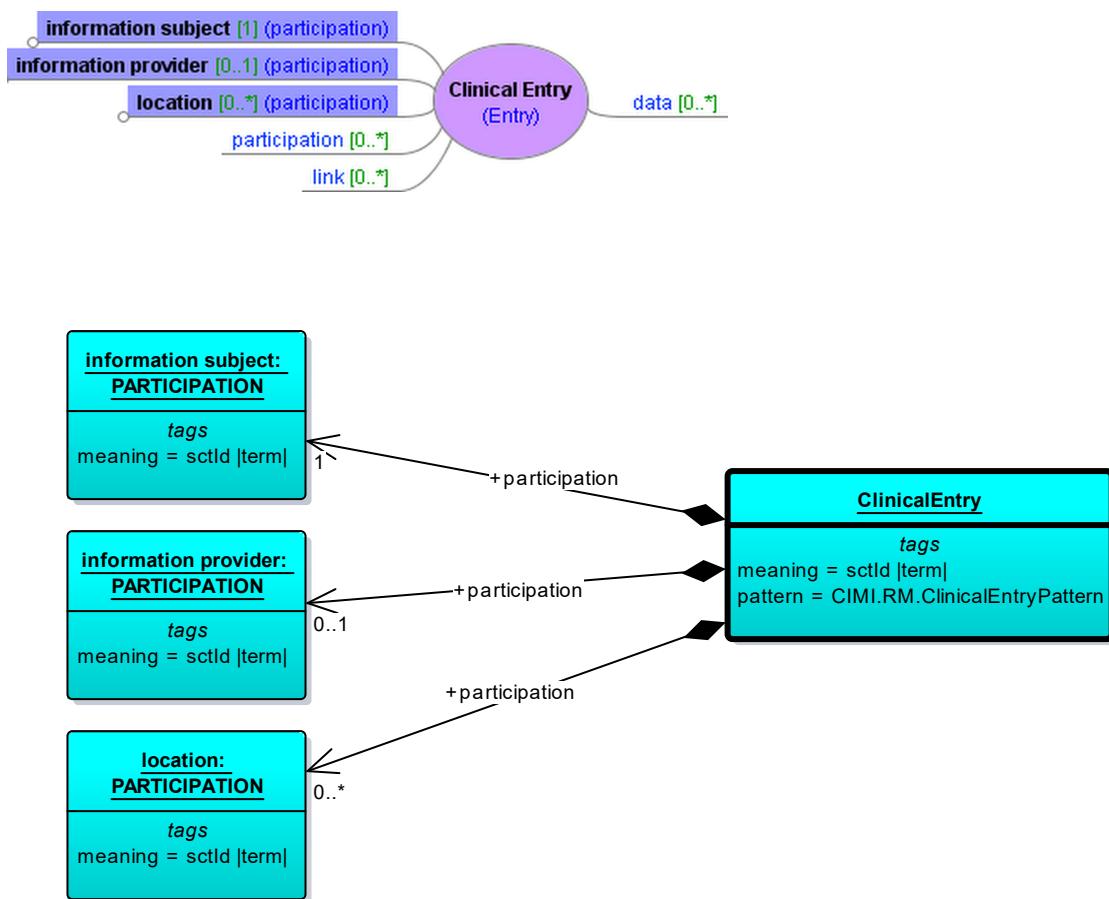


Figure 12: CIMI.RM.ClinicalEntryPattern

### <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)	Source: Public participation (Object) ClinicalEntry Target: Public (Object) information provider Cardinality: [0..1]
Association (direction: Source -> Destination)	Source: Public participation (Object) ClinicalEntry Target: Public (Object) information provider Cardinality: [0..1]
Association (direction: Source -> Destination)	Source: Public participation (Object) ClinicalEntry Target: Public (Object) information provider Cardinality: [0..1]

<b>ASSOCIATIONS</b>	
Source: Public participation (Object) ClinicalEntry	Target: Public (Object) location Cardinality: [0..*]
Association (direction: Source -> Destination)	Source: Public participation (Object) ClinicalEntry Target: Public (Object) information subject Cardinality: [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

<b>ASSOCIATIONS</b>	
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

<b>ASSOCIATIONS</b>	
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

<b>ASSOCIATIONS</b>	
Association (direction: Source -> Destination)	

**ASSOCIATIONS**

Source: Public participation (Object) ClinicalEntry

Target: Public (Object) location

Cardinality: [0..\*]

## 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

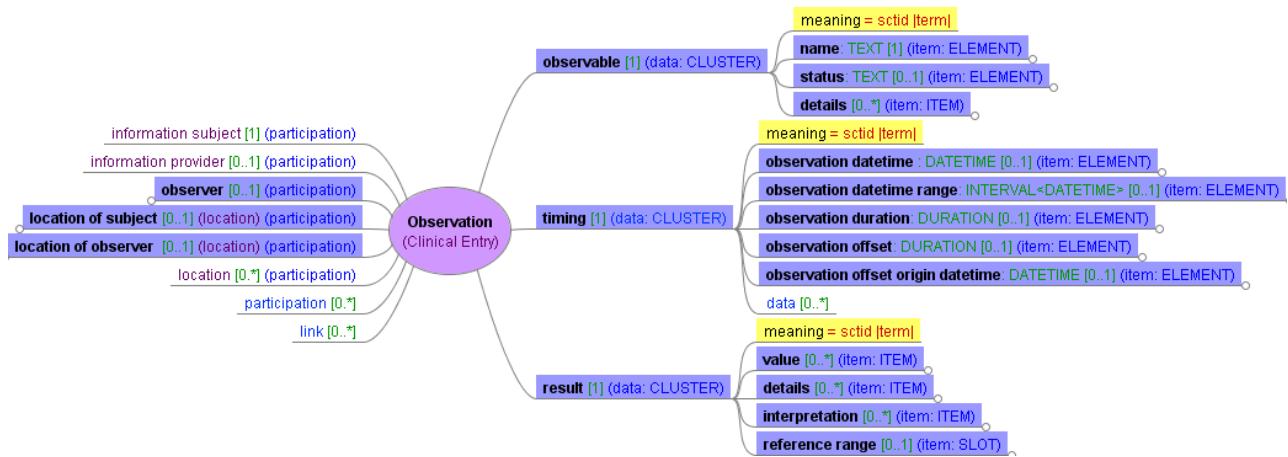


Figure 13: CIMI.RM.ObservationPattern (mindmap)

## <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

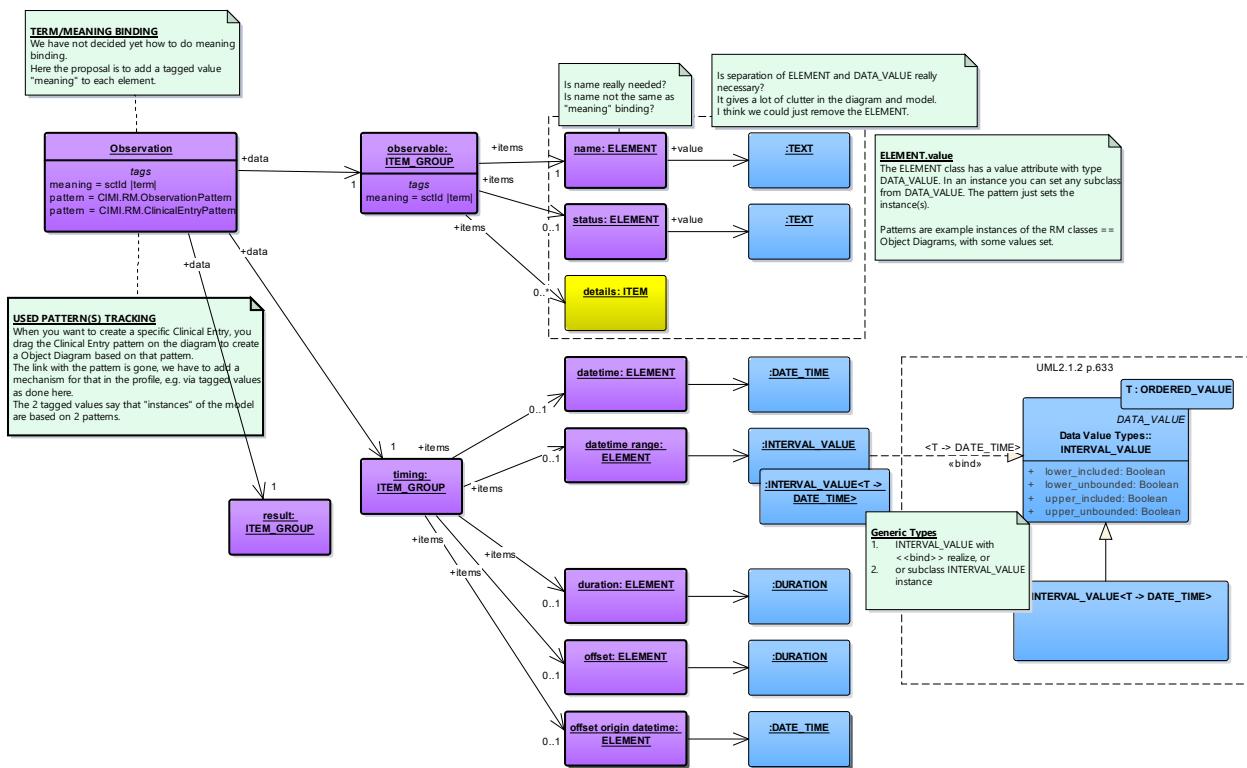


Figure 14: CIMI.RM.ObservationPattern (Object)

## 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'. ]
⇒ 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'. ]
ATTRIBUTES	
◆ lower_included : 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. ]
◆ upper_unbounded : 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

zelm created on 10/5/2012. Last modified 10/5/2012

Extends

## <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.

&lt;anonymous&gt;

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.

&lt;anonymous&gt;

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.

&lt;anonymous&gt;

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)'*

&lt;anonymous&gt;

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) &lt;anonymous&gt;

**<anonymous>***Object in package 'CIMI.RM.ObservationPattern (Object)'*

&lt;anonymous&gt;

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) &lt;anonymous&gt;

**<anonymous>***Object in package 'CIMI.RM.ObservationPattern (Object)'*

&lt;anonymous&gt;

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) &lt;anonymous&gt;

**<anonymous>***Object in package 'CIMI.RM.ObservationPattern (Object)'*

&lt;anonymous&gt;

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) &lt;anonymous&gt;

**<anonymous>***Object in package 'CIMI.RM.ObservationPattern (Object)'*

&lt;anonymous&gt;

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 &lt;T -&gt; DATE\_TIME&gt;. Direction is 'Source -&gt; Destination'. ]

**ASSOCIATIONS** Association (direction: Source -> Destination)

Source: Public (Object) datetime range

Target: Public (Object) &lt;anonymous&gt;

**<anonymous>***Object in package 'CIMI.RM.ObservationPattern (Object)'*

&lt;anonymous&gt;

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)'*

&lt;anonymous&gt;

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) &lt;anonymous&gt;

**<anonymous>***Object in package 'CIMI.RM.ObservationPattern (Object)'*

&lt;anonymous&gt;

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) observable  
Cardinality: [1]

 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]

## 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)

Source: Public items (Object) timing

Target: Public (Object) duration  
Cardinality: [0..1]

## ASSOCIATIONS

### 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) name  
Cardinality: [1]

- ✓ 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 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

## ASSOCIATIONS

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  
Cardinality: [0..1]

 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 items (Object) timing

Target: Public (Object) datetime  
Cardinality: [0..1]

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 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

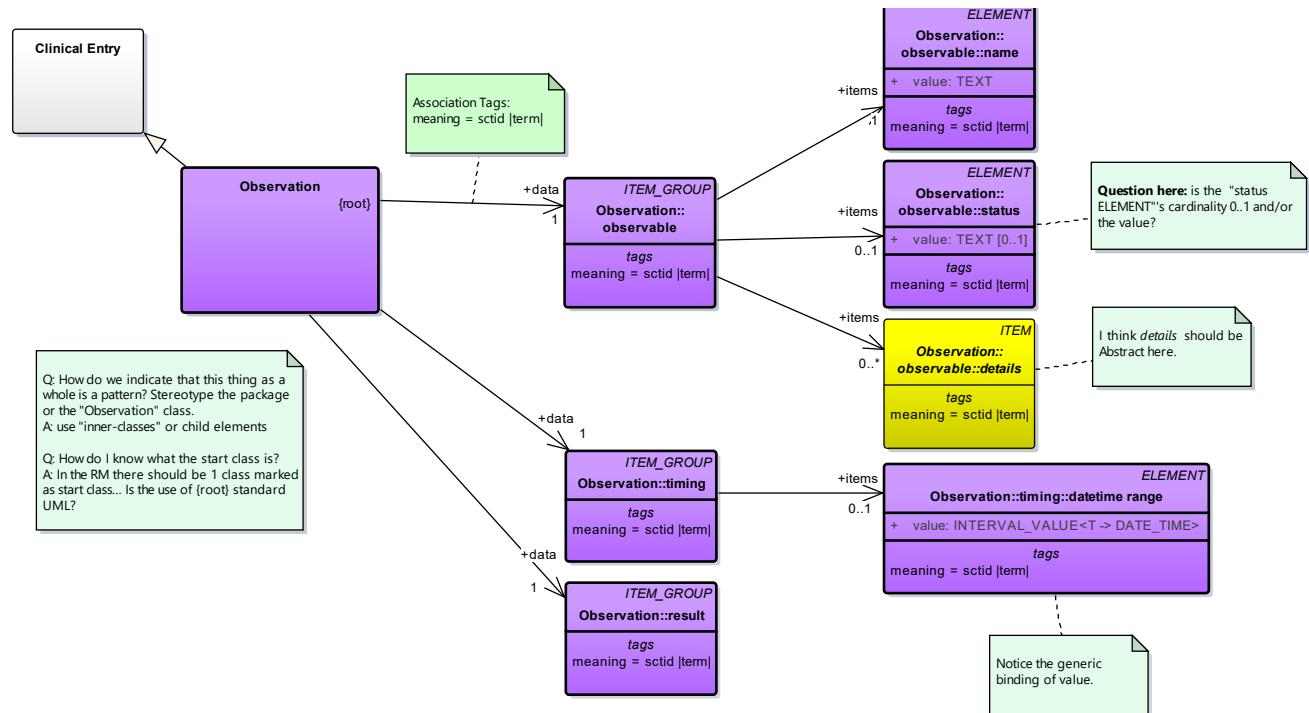


Figure 15: CIMI.RM.ObservationPattern (Class)

## <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) result

Cardinality: [1]

↙ Association (direction: Source -> Destination)

Source: Public (Class) Observation

Target: Public data (Class) observable

Cardinality: [1]

↙ Association (direction: Source -> Destination)

Source: Public (Class) Observation

Target: Public data (Class) timing

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

<b>ELEMENTS OWNED BY observable</b>	
	details : Class
	name : Class
	status : Class

<b>OUTGOING STRUCTURAL RELATIONSHIPS</b>	
Generalization from observable to ITEM_GROUP	[ Direction is 'Source -> Destination'. ]

<b>ASSOCIATIONS</b>	
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) details Cardinality: [0..*]
Association (direction: Source -> Destination)	
Source: Public (Class) observable	Target: Public items (Class) name Cardinality: [1]
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

<b>OUTGOING STRUCTURAL RELATIONSHIPS</b>	
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
Association (direction: Source -> Destination) Source: Public (Class) observable Target: Public items (Class) status Cardinality: [0..1]

### 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
Generalization from result to ITEM_GROUP [ Direction is 'Source -> Destination'. ]

ASSOCIATIONS
Association (direction: Source -> Destination) Source: Public (Class) Observation Target: Public data (Class) result Cardinality: [1]

### 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
datetime range : Class

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

<b>ASSOCIATIONS</b>	
 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

<b>OUTGOING STRUCTURAL RELATIONSHIPS</b>	
 Generalization from datetime range to ELEMENT	[ Direction is 'Source -> Destination'. ]

<b>ATTRIBUTES</b>	
 value : INTERVAL_VALUE<T -> DATE_TIME> Public	[ Is static False. Containment is Not Specified. ]

<b>ASSOCIATIONS</b>	
 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

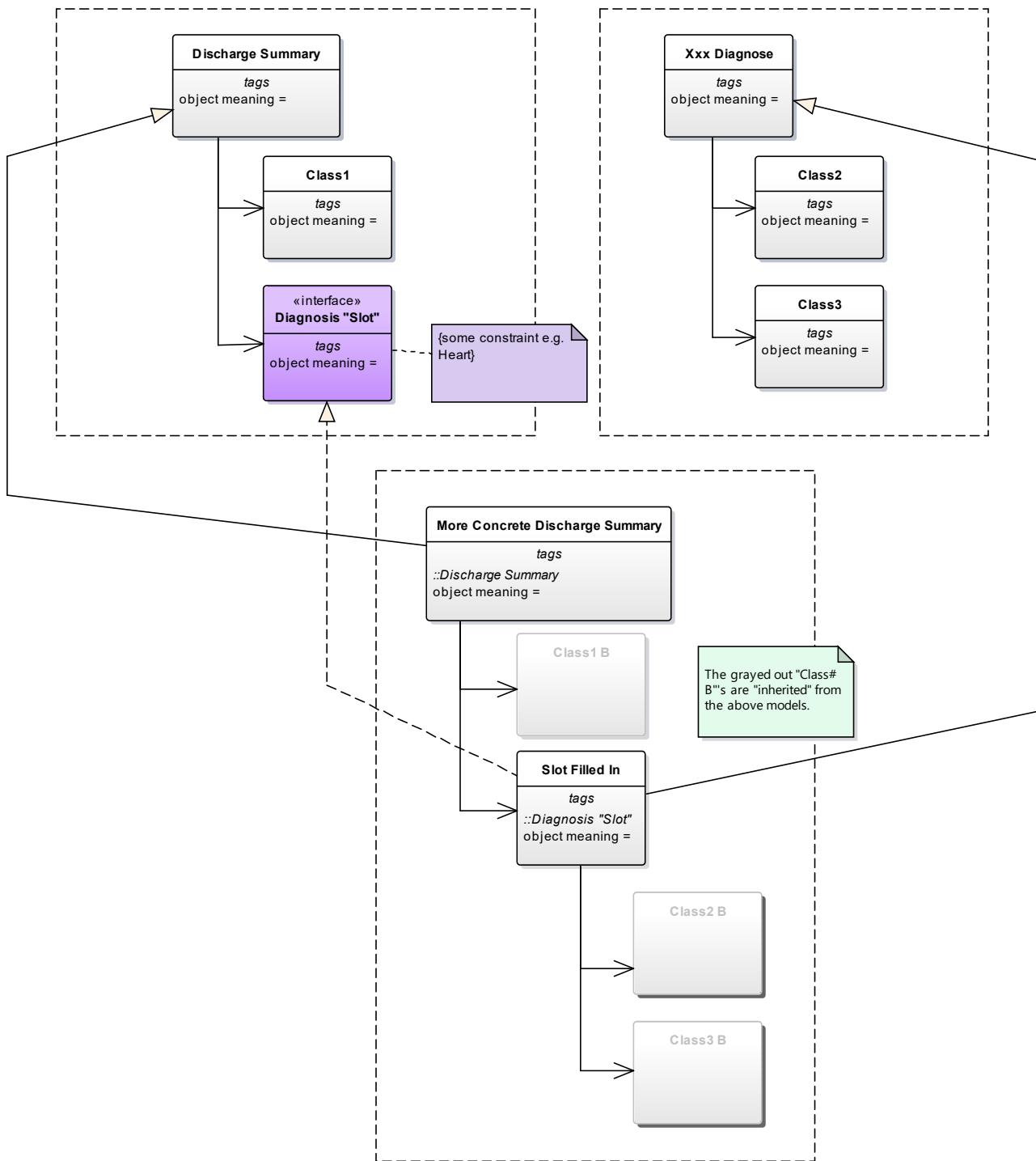


Figure 16: Slots in UML Example

### <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
 Association (direction: Source -> Destination) Source: Public (Class) Discharge Summary      Target: Public (Class) Class1

## 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

**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) Slot Filled In

✍ Association (direction: Source -> Destination)

Source: Public (Class) More Concrete Discharge Summary

Target: Public (Class) Class1 B

**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) Class2

↙ Association (direction: Source -> Destination)

Source: Public (Class) Xxx Diagnose

Target: Public (Class) Class3

## 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 17: 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

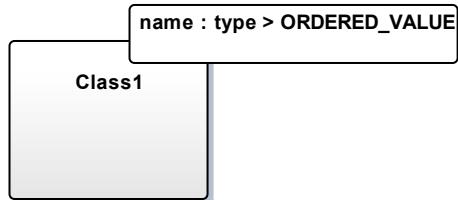


Figure 18: Parameterised Classes

## 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