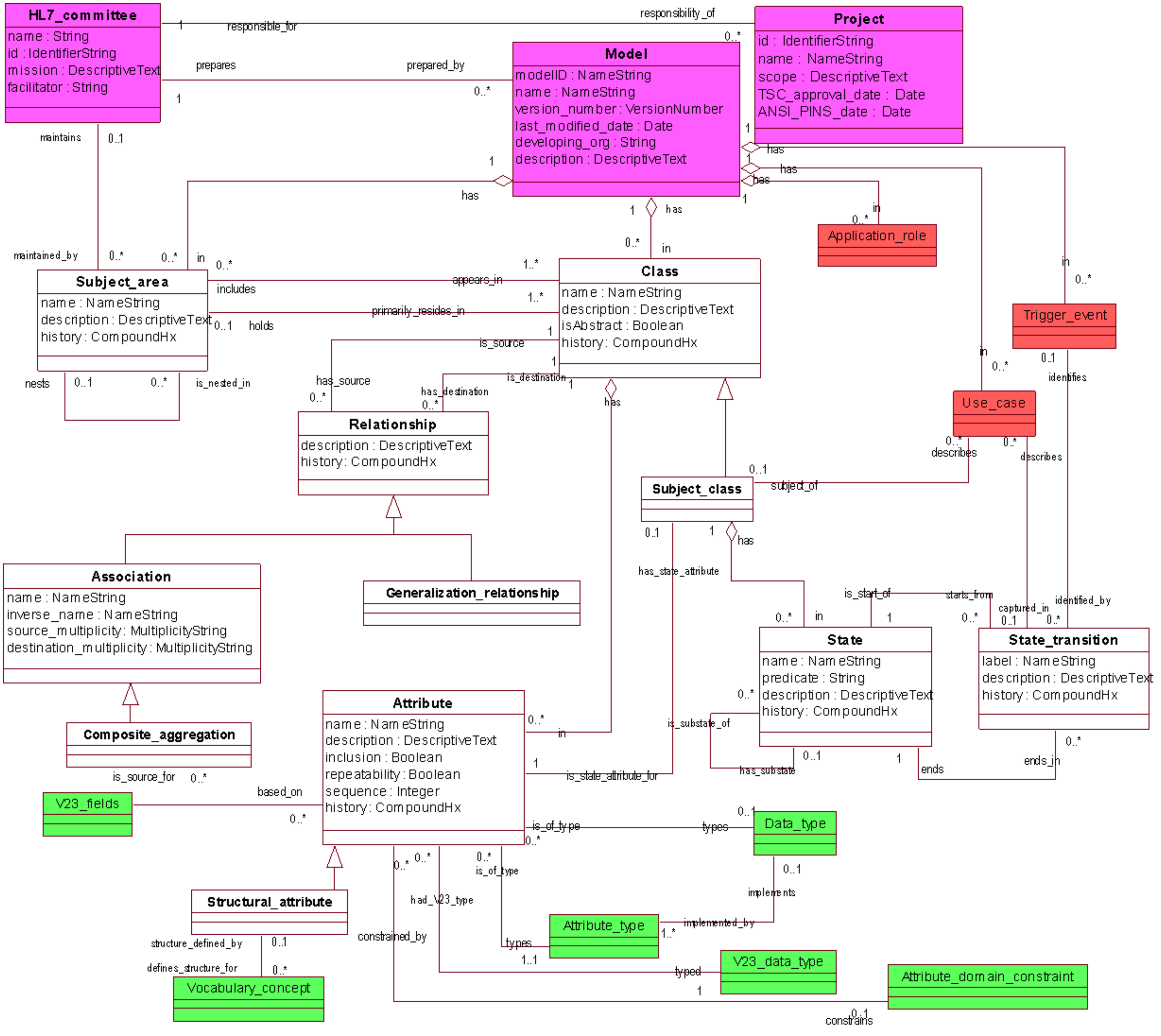
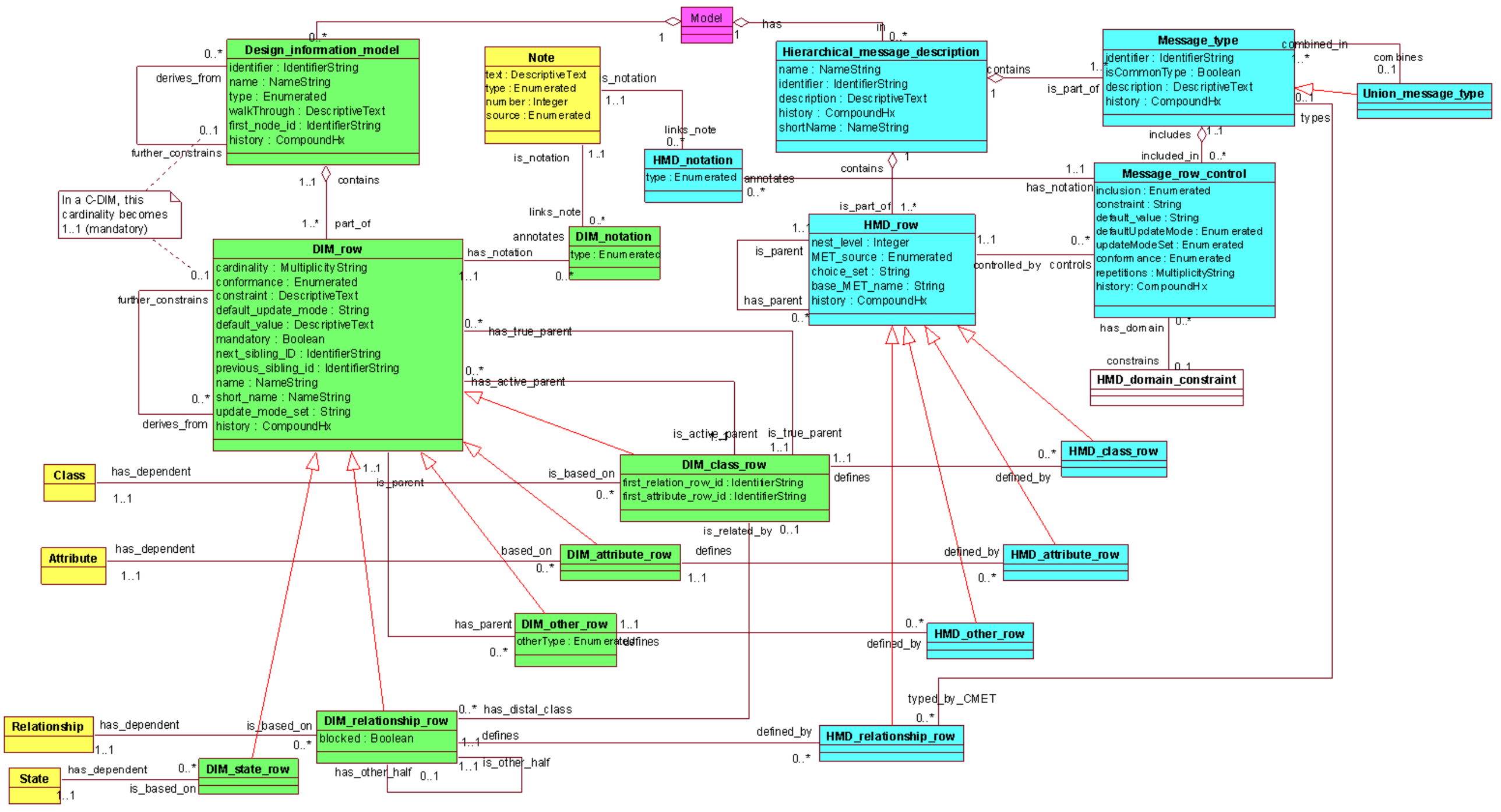

HL7_committee
name : String
id : IdentifierString
mission : DescriptiveText
facilitator : String
responsible_for
responsibility_of
Project
id : IdentifierString
name : NameString
scope : DescriptiveText
TSC_approval_date : Date
ANSI_PINS_date : Date
prepares
prepared_by
Model
modelID : NameString
name : NameString
version_number : VersionNumber
last_modified_date : Date
developing_org : String
description : DescriptiveText
maintains
maintained_by
has
in
Application_role
Subject_area
name : NameString
description : DescriptiveText
history : CompoundHx
includes
appears_in
Class
name : NameString
description : DescriptiveText
isAbstract : Boolean
history: CompoundHx
primarily_resides_in
holds
nests
is_nested_in
is_source
is_destination
has_source
has_destination
Relationship
description : DescriptiveText
history: CompoundHx
Trigger_event
identifies
Use_case
describes
Subject_class
subject_of
Association
name : NameString
inverse_name : NameString
source_multiplicity: MultiplicityString
destination_multiplicity: MultiplicityString
Generalization_relationship
Composite_aggregation
is_source_for
has_state_attribute
State
name : NameString
predicate : String
description : DescriptiveText
history: CompoundHx
is_start_of
starts_from
captured_in
identified_by
State_transition
label : NameString
description : DescriptiveText
history: CompoundHx
is_substate_of
has_substate
ends
ends_in
Attribute
name : NameString
description : DescriptiveText
inclusion : Boolean
repeatability : Boolean
sequence : Integer
history: CompoundHx
is_state_attribute_for
V23_fields
based_on
is_of_type
types
Data_type
implements
Structural_attribute
had_V23_type
Attribute_type
implemented_by
constrained_by
structure_defined_by
defines_structure_for
Vocabulary_concept
typed
V23_data_type
Attribute_domain_constraint
constrains


Model
has
in
Design_information_model
identifier : IdentifierString
name : NameString
type : Enumerated
walkThrough : DescriptiveText
first_node_id : IdentifierString
history : CompoundHx
derives_from
further_constrains
In a C-DIM, this cardinality becomes 1..1 (mandatory)
contains
part_of
Note
text : DescriptiveText
type : Enumerated
number : Integer
source : Enumerated
s_notation
links_note
is_notation
HMD_notation
type : Enumerated
annotates
DIM_notation
type : Enumerated
has_notation
Hierarchical_message_description
name : NameString
identifier : IdentifierString
description : DescriptiveText
history : CompoundHx
shortName : NameString
contains
is_part_of
Message_type
identifier : IdentifierString
isCommonType : Boolean
description : DescriptiveText
history : CompoundHx
combined_in
combines
Union_message_type
types
includes
included_in
Message_row_control
inclusion : Enumerated
constraint : String
default_value : String
defaultUpdateMode : Enumerated
updateModeSet : Enumerated
conformance : Enumerated
repetitions : MultiplicityString
history: CompoundHx
has_notation
HMD_row
nest_level : Integer
MET_source : Enumerated
choice_set : String
base_MET_name : String
history : CompoundHx
is_parent
has_parent
controlled_by
controls
has_domain
constrains
HMD_domain_constraint
DIM_row
cardinality : MultiplicityString
conformance : Enumerated
constraint : DescriptiveText
default_update_mode : String
default_value : DescriptiveText
mandatory : Boolean
next_sibling_ID : IdentifierString
previous_sibling_id : IdentifierString
name : NameString
short_name : NameString
update_mode_set : String
history : CompoundHx
has_true_parent
has_active_parent
is_active_parent
is_true_parent
DIM_class_row
first_relation_row_id : IdentifierString
first_attribute_row_id : IdentifierString
defines
defined_by
HMD_class_row
Class
has_dependent
is_based_on
is_parent
Attribute
DIM_attribute_row
based_on
HMD_attribute_row
is_related_by
DIM_other_row
otherType : Enumerated
has_parent
HMD_other_row
Relationship
DIM_relationship_row
blocked : Boolean
has_distal_class
has_other_half
is_other_half
HMD_relationship_row
typed_by_CMET
State
DIM_state_row

Model
modelID : NameString
name : NameString
version_number : VersionNumber
last_modified_date : Date
developing_org : String
description : DescriptiveText

Storyboard
name : NameString
identifier : IdentifierString
purpose : DescriptiveText
history : CompoundHx

Storyboard_example
identifier : IdentifierString
description : DescriptiveText
history: CompoundHx

Design_category
name : NameString
identifier : IdentifierString
level : Enumerated
description : DescriptiveText
history : CompoundHx

Application_role_relationship
type : Enumerated
history : CompoundHx

Design_annotation
target_type : Enumerated

Note
text : DescriptiveText
type : Enumerated
number : Integer
source : Enumerated

Reference_note
standard_name : NameString

Interaction_sequence
sequence_number : Integer
sending_system_name : NameString
receiving_system_name : NameString
history : CompoundHx

Application_role
identifier : IdentifierString
name : NameString
short_name : NameString
description : DescriptiveText
history : CompoundHx

Use_case_category
name : NameString
description : DescriptiveText
history : CompoundHx

Actor
name : NameString
description : DescriptiveText
history : CompoundHx

Use_case_sequence
sequence : Integer
history : CompoundHx

Trigger_event
identifier : IdentifierString
name : NameString
dependency : String
description : DescriptiveText
history : CompoundHx

Interaction
identifier : IdentifierString
description : DescriptiveText
history : CompoundHx
name : NameString
shortName : NameString

Class

Use_case
identifier : IdentifierString
name : NameString
description : DescriptiveText
history : CompoundHx

Use_case_relationship
stereotype : String
history : CompoundHx

State_transition

Subject_class

Receiver_responsibility
identifier : IdentifierString
reason : DescriptiveText
history : CompoundHx

Message_type

Interaction_type
identifier : IdentifierString
name : NameString
description : DescriptiveText
stateBasedTriggerAssociation : Enumerated
history: CompoundHx

Control_event
identifier : IdentifierString
name : NameString
history: CompoundHx

Communication_wrapper
identifier : IdentifierString
name : NameString
history : CompoundHx

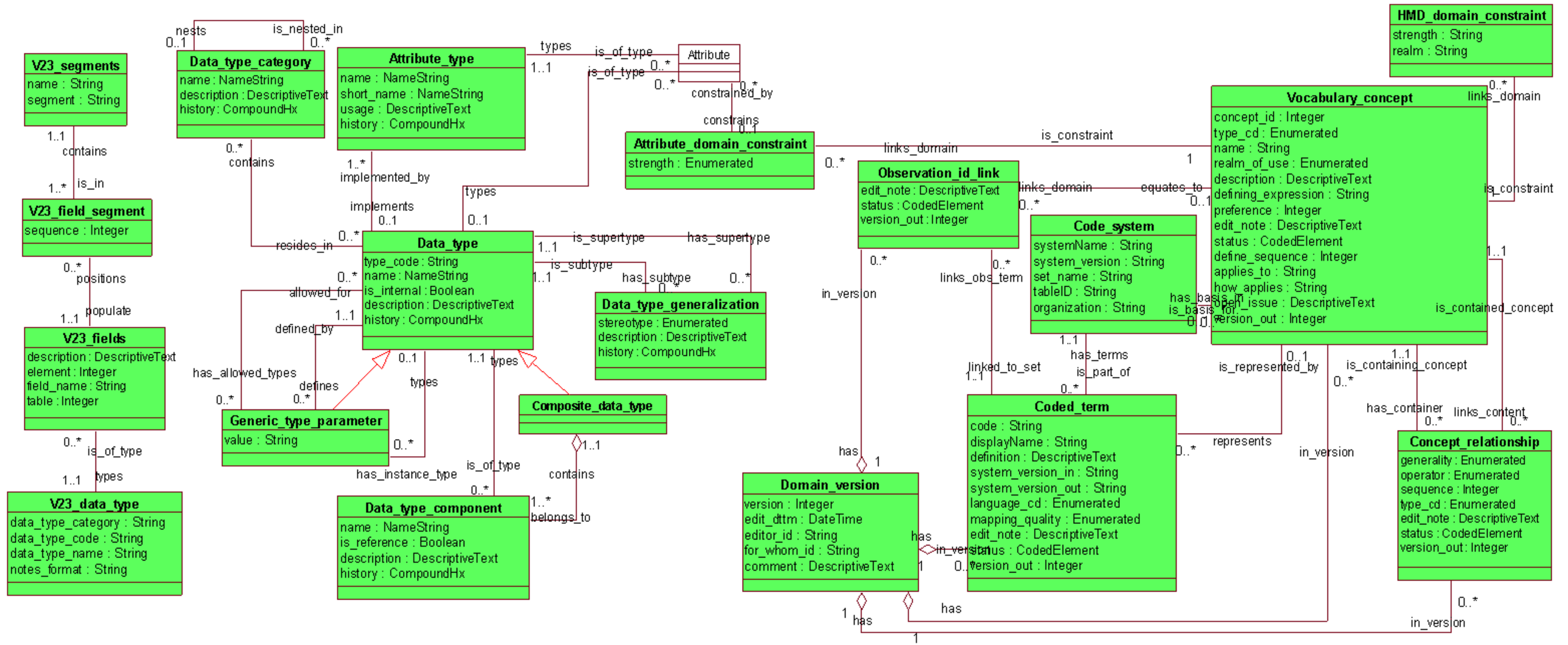

V23_segments
name : String
segment : String
contains
is_in
V23_field_segment
sequence : Integer
positions
populate
V23_fields
description : DescriptiveText
element : Integer
field_name : String
table : Integer
is_of_type
types
V23_data_type
data_type_category : String
data_type_code : String
data_type_name : String
notes_format : String
nests
is_nested_in
Data_type_category
name : NameString
description : DescriptiveText
history : CompoundHx
contains
resides_in
Attribute_type
name : NameString
short_name : NameString
usage : DescriptiveText
history : CompoundHx
implemented_by
implements
types
is_of_type
Attribute
constrained_by
constrains
Attribute_domain_constraint
strength : Enumerated
Data_type
type_code : String
name : NameString
is_internal : Boolean
description : DescriptiveText
history : CompoundHx
is_supertype
has_supertype
is_subtype
has_subtype
Data_type_generalization
stereotype : Enumerated
description : DescriptiveText
history : CompoundHx
allowed_for
has_allowed_types
defined_by
defines
Generic_type_parameter
value : String
has_instance_type
Composite_data_type
contains
belongs_to
Data_type_component
name : NameString
is_reference : Boolean
description : DescriptiveText
history : CompoundHx
HMD_domain_constraint
strength : String
realm : String
links_domain
is_constraint
Vocabulary_concept
concept_id : Integer
type_cd : Enumerated
name : String
realm_of_use : Enumerated
description : DescriptiveText
defining_expression : String
preference : Integer
edit_note : DescriptiveText
status : CodedElement
define_sequence : Integer
applies_to : String
how_applies : String
open_issue : DescriptiveText
version_out : Integer
is_contained_concept
Observation_id_link
edit_note : DescriptiveText
status : CodedElement
version_out : Integer
equates_to
Code_system
systemName : String
system_version : String
set_name : String
tableID : String
organization : String
has_basis_in
is_basis_for
links_obs_term
in_version
has_terms
is_part_of
linked_to_set
is_represented_by
is_containing_concept
has_container
links_content
Coded_term
code : String
displayName : String
definition : DescriptiveText
system_version_in : String
system_version_out : String
language_cd : Enumerated
mapping_quality : Enumerated
edit_note : DescriptiveText
status : CodedElement
version_out : Integer
represents
Concept_relationship
generality : Enumerated
operator : Enumerated
sequence : Integer
type_cd : Enumerated
edit_note : DescriptiveText
status : CodedElement
version_out: Integer
Domain_version
version : Integer
edit_dttm : DateTime
editor_id : String
for_whom_id : String
comment : DescriptiveText
has
in_version

Table of contents for: **HL7_V3_Meta-Model**

Model: **HL7_V3_Meta-Model**

Identifications:

Organization: HL7

Version: V 1-16 20020512

ModelID: MET_0116

Developed by: Modeling and Methodology

Description of: **HL7_V3_Meta-Model**

This model advances the meta-model to account for methodology changes adopted through March, 2002.

All section have been updated, and the concept of a Message Information model (MIM) has been dropped. This model was prepared as part of the preparatory work for developing the HL7 Development Framework (HDF).

Subject Areas for: **HL7_V3_Meta-Model**

Subject Area: **MET_Data_type_model**

The data type model defines the structure of the data types that may be assigned to information model attributes when these attributes are included in messages. It expresses the hierarchical relationship between data types and their components. It defines the role for attribute types in the information model. It also includes the structure of HL7 Version 2.x fields and data types.

Contains classes:
Attribute
Attribute_type
Composite_data_type
Data_type
Data_type_category
Data_type_component
Data_type_generalization
Generic_type_parameter
HL7_committee
Model
V23_data_type
V23_field_segment
V23_fields
V23_segments

Subject Area: **MET_Design_information_model**

Contains classes:
Attribute
Class
Design_information_model
DIM_attribute_domain_constraint
DIM_attribute_row
DIM_class_row
DIM_notation
DIM_other_row
DIM_relationship_row
DIM_row
DIM_state_row
Model
Note
Relationship
State

Subject Area: **MET_Hierarchical_message_description**

The Hierarchical message description model specifies the semantic links between elements of a MIM, the message object diagram (MOD), the abstract message definition for a set of message structures, and the message structures themselves.

Contains classes: **DIM_attribute_row**

DIM_class_row
DIM_other_row
DIM_relationship_row
Hierarchical_message_description
HMD_attribute_row
HMD_class_row
HMD_domain_constraint
HMD_notation
HMD_other_row
HMD_relationship_row
HMD_row
Message_row_control
Message_type
Model
Note
Union_message_type

Subject Area: **MET_Information_model**

The information model defines the content of messages exchanged with HL7. Classes, connections, attributes, and states are the primary building blocks of the information model. Classes provide abstractions of the objects represented by the model. The semantic relationships between classes are expressed using connections. The three types of connections are Generalization-specialization, Whole-part, and Instance. Attributes are the facts applicable to the objects of the class, and states capture changes that trigger events have upon the subject classes of the information model.

Contains classes:
Association
Attribute
Attribute_domain_constraint
Attribute_type
Class
Composite_aggregation
Data_type
Generalization_relationship
HL7_committee
Relationship
State
State_transition
Structural_attribute
Subject_area
Subject_class
V23_data_type
V23_fields
Vocabulary_concept

Subject Area: **MET_Interaction_model**

The interaction model specifies the information flows that are needed to support the use cases defined in the use case model.

It includes the information flows or interactions, the trigger events, the application roles that send and receive the interactions and scenarios that provide an interaction trace for a series of events.

Contains classes:
Application_role
Application_role_relationship
Communication_wrapper
Control_event
Design_annotation
Design_category

HL7_committee
Interaction
Interaction_sequence
Interaction_type
Message_type
Model
Note
Receiver_responsibility
Reference_note
Storyboard
Storyboard_example
Trigger_event

Subject Area: **MET_Message_specification_model**

The message specification model maps the information content of the information model into the abstract and concrete message specifications needed to communicate between computer applications.

It includes: the message information model which is the sub-set of the information model needed to support a set of messages; the hierarchical message description that maps the information content of the MIM into a set of message formats; and the message element type model which describes the type structure used to convey messages.

Contains classes:
Data_type
Design_information_model
DIM_attribute_row
DIM_class_row
DIM_notation
DIM_other_row
DIM_relationship_row
DIM_row
DIM_state_row
Hierarchical_message_description
HL7_committee
HMD_attribute_row
HMD_class_row
HMD_domain_constraint
HMD_notation
HMD_other_row
HMD_relationship_row
HMD_row
Message_row_control
Message_type
Model
Project
Union_message_type

Subject Area: **MET_Model_identification_and_scope**

The components that define the overall model, the project and the domain information models that support the project.

Contains classes:
HL7_committee
Model
Project

Subject Area: **MET_Use_case_model**

The use case model is a collection of actors, use cases and scenarios that comprise a high level functional analysis of healthcare. For HL7, the purpose of this analysis is to the requirements for messaging between

computer applications . The Use Case Model documents the institutional, medical, and business practices that the message(s) being created will support.

Contains classes:
Actor
HL7_committee
Use_case
Use_case_category
Use_case_relationship

Subject Area: **MET_Vocabulary_domain_model**

Defines the structure and relationships for vocabulary domains that are used to constrain coded information model attributes in the information and message design models.

Contains classes:
Attribute
Attribute_domain_constraint
Code_system
Coded_term
Concept_relationship
DIM_attribute_domain_constraint
DIM_attribute_row
Domain_version
HMD_domain_constraint
Observation_id_link
Vocabulary_concept

Subject Area: **Meta_1_Use_case_and_Interaction_models**

Defined for graphing purposes only.

Contains classes:
Actor
Application_role
Application_role_relationship
Class
Communication_wrapper
Control_event
Design_annotation
Design_category
Interaction
Interaction_sequence
Interaction_type
Message_type
Model
Note
Receiver_responsibility
Reference_note
State_transition
Storyboard
Storyboard_example
Subject_class
Trigger_event
Use_case
Use_case_category
Use_case_relationship
Use_case_sequence

Subject Area: **Meta_2_Information_model**

Defined for graphing purposes only.

Contains classes: **Application_role**
Association
Attribute
Attribute_domain_constraint
Attribute_type
Class
Composite_aggregation
Data_type
Generalization_relationship
HL7_committee
Model
Project
Relationship
State
State_transition
Structural_attribute
Subject_area
Subject_class
Trigger_event
Use_case
V23_data_type
V23_fields
Vocabulary_concept

Subject Area: **Meta_3_Data_type_and_Domain_models**

Contains classes: **Attribute**
Attribute_domain_constraint
Attribute_type
Code_system
Coded_term
Composite_data_type
Concept_relationship
Data_type
Data_type_category
Data_type_component
Data_type_generalization
Domain_version
Generic_type_parameter
HMD_domain_constraint
Observation_id_link
V23_data_type
V23_field_segment
V23_fields
V23_segments
Vocabulary_concept

Information model in: **HL7_V3_Meta-Model**

Classes in: **HL7_V3_Meta-Model**

Class: **Actor**

Is part of: **Model**

Associated with: **Use_case**
Use_case_category

Description of: **Actor**

An actor is a role played by someone or something that interacts directly with the elements represented by the classes in the information model. With one exception, actors cannot represent information systems. The exception is a special actor with the literal name "some information system". The name is chosen to reinforce the notion that use cases are not built on a priori knowledge of the functionality of specific healthcare information systems.

Composition for: **Actor**

in (1,1) :: Model :: **has (0,n)**

The relationship between actors and the models of which they are a part.

Associations for: **Actor**

participates_in :: (0..n) Use_case :: **involves** :: (1..n)

included_in :: (0..n) Use_case_category :: **includes** :: (0..n)

Attributes of: **Actor**

description : DescriptiveText

A short informative statement that allows people to determine, with certainty, whether a particular real world role is an instance of the actor.

history : CompoundHx

This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

name : NameString

The actors in the model are each given a unique name. The actor name is a singular noun or noun phrase.

Class: **Application_role**

Is part of: **Model**

Associated with: **Application_role_relationship**
Application_role_relationship
Design_annotation
Design_category
Interaction
Interaction

Description of: **Application_role**

Application roles are abstractions that name roles that may be played by health-care information system components when sending or receiving HL7 messages. Thus they are a defined set of responsibilities with respect to interactions. The role may have responsibility to send or receive one or more interactions. The application role and its responsibilities may form the basis for establishing conformance specifications for a standard.

Application roles should be stereotyped with respect to their responsibilities for information about a subject class. The technical committee should start its thinking about application roles from the perspective that there are three fundamental purposes for message exchange, three basic "messaging modes". These are:

Declarative - This includes messages that are sent with the intent of conveying information from one party to another. For example, a healthcare provider might send a message every time a person is registered as a patient with that provider.

Imperative - This includes messages that direct or request a party to do something. For example, a healthcare provider might send a message to a laboratory every time the provider needs the laboratory to perform a test. Note, that even though the message must include information about the test and the patient the test is for, the primary purpose of the message is to request that the test be done.

Interrogative - This includes messages that ask for information, that ask a question. For example, a healthcare provider might send a message to an MPI mediator asking whether the MPI mediator has information about a specific person.

Application roles will have stereotyped names constructed as <subject class> <relationship>. These stereotypes are specific to the messaging modes.

- For the declarative mode, the typical roles are "declarer" and "recipient"

- For the imperative model, the typical roles are "placer" and "filler"

- For the interrogative mode, the typical roles are "questioner" and "answerer"

There is no requirement that a Technical Committee create all of these "<subject class><relationship>" roles. Nor is it is limited to these possibilities. But it is expected that the committee will consider these stereotypes.

Application roles may contain other application roles, in which case they inherit or contain the characteristics and responsibilities of the contained role based on the type of containment relationship.

Composition for: **Application_role**

in (1,1) :: Model :: **has (0,n)**
The relationship between application roles and the models of which they are a part.

Associations for: **Application_role**

source_for :: (0..n) Application_role_relationship :: **has_source** :: (1..1)
The application role includes (based on the application role relationship type) the responsibilities for the target application roles of any application role relationship for which it is the source.

target_of :: (0..n) Application_role_relationship :: **has_target** :: (1..1)
An application role's responsibilities are assumed by (based on the application role relationship type) the source application roles of any application role relationship for which it is the target.

has :: (0..n) Design_annotation :: **for** :: (0..1)

defined_in :: (1..1) Design_category :: **includes** :: (0..n)

receives :: (0..n) Interaction :: **received_by** :: (1..1)
A reference to the application role that is responsible for receiving the message involved in this interaction. The receiving role must be prepared to accept the message and to fulfill the receiver responsibility.

sends :: (0..n) Interaction :: **sent_by** :: (1..1)
The sending role has responsibilities to recognize the trigger event for the interaction and to cause the appropriate message to be sent.

Attributes of: **Application_role**

description : DescriptiveText
The text that describes the application role and summarizes the interaction responsibilities that are part of that role.

history : CompoundHx
This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

identifier : IdentifierString
An identifier assigned to the application role. The identifier is unique within the scope of the model in which the application role is defined. In HL7, committees manage the unique identifiers for their application roles following a naming standard established by HL7.

name : NameString
A name assigned to the application role. The name is unique within the scope of the model in which the application role is defined and is derived based on a formal naming standard established by HL7. The present standard is: (<Focal class> <mood><State transition capability><Application capability/coupling>, and the standards for Query Roles remains to be established).

short_name : NameString
A short, common-use name assigned to the application role. The short name is intended to be familiar to domain experts and is not required to be unique.

Class: **Application_role_relationship**

Is part of: **Model**

Associated with: **Application_role**
Application_role

Description of: **Application_role_relationship**
Application roles relationships allow the building of hierarchies of application roles such that one application role automatically takes on the characteristics of the application role it is related to.

At present, Application roles relationships can only be of two types:

Includes - The source application role behaves as though it were the sender and receiver for all interactions that are defined for the target application role.

OneOf: The source application role behaves as though it were the sender and receiver of all interactions that are defined for one and only one of its target application roles.

Application roles relationships may not be defined in a recursive manner. In traveling from source to target along all defined application role relationships there should be no situation where the original application role is identified as a target. I.e. the chain of application roles must not form a loop.

Composition for: **Application_role_relationship**

in (1,1) :: Model :: **has (0,n)**
The relationship between application role relationships and the models of which they are a part.

Associations for: **Application_role_relationship**

has_source :: (1..1) Application_role :: **source_for** :: (0..n)

The application role includes (based on the application role relationship type) the responsibilities for the target application roles of any application role relationship for which it is the source.

has_target :: (1..1) Application_role :: **target_of** :: (0..n)
An application role's responsibilities are assumed by (based on the application role relationship type) the source application roles of any application role relationship for which it is the target.

Attributes of: **Application_role_relationship**

history : CompoundHx

type : Enumerated
Defines how the source and target application role relate. At present, the permitted values are 'includes', and "oneOf," where the later implies that when conformance is claimed, the claimant must select one (and only one) of the included roles to implement.

Class: **Association**

Subtype of: **Relationship**

Supertype of: **Composite_aggregation**

Description of: **Association**
An association is a relationship between classes that depicts the occurrence of a reference attribute used to connect class instances (objects). The associated objects can be of the same or different classes. When the association is defined one of the two classes is designated the "source class" and the other the "target" class. These designations are necessary to unambiguously define an association, but the designations have no semantic implications about the roles of the associated classes within the information model being defined. The selection of which class to label as "source" is arbitrary.

Attributes of: **Association**

destination_multiplicity : MultiplicityString
A set of values and value ranges indicating the number of destination class instances involved in the connection. In value ranges the minimum shall be zero or more and the maximum shall be equal to or greater than the minimum. The maximum number may be expressed as unlimited.

inverse_name : NameString
A short action phrase that specifies the role of the destination class in the association. Each association between the same pair of classes must have a unique inverse name.

name : NameString
A short action phrase that specifies the role of the source class in the association. Each association between the same pair of classes must have a unique name.

source_multiplicity : MultiplicityString
A set of values and value ranges indicating the number of source class instances involved in the connection. In value ranges the minimum shall be zero or more and the maximum shall be equal to or greater than the minimum. The maximum number may be expressed as unlimited.

Class: **Attribute**

Supertype of: **Structural_attribute**

Is part of: **Class**

Associated with: **Attribute_domain_constraint**
Attribute_type
Data_type
DIM_attribute_row
Subject_class
V23_data_type
V23_fields
V23_segments

Description of: **Attribute**

Attributes in the information model are the major source of the data content used in HL7 communications. Attributes are abstractions of the data captured about classes. Attributes capture separate aspects of the class and take their values independent of one another. Attribute domain specifications are captured in datatypes.

Composition for: **Attribute**

in (1,1) :: Class :: **has (0,n)**

The relationship between attributes and the classes of which they are a part.

Associations for: **Attribute**

constrained_by :: (0..1) Attribute_domain_constraint :: **constrains** :: (0..n)

is_of_type :: (1..1) Attribute_type :: **types** :: (0..n)

A reference between an attribute and its attribute type. The attribute type code must also be the terminal component of the attribute name.

is_of_type :: (0..1) Data_type :: **types** :: (0..n)

A link between an attribute and the datatype that has been assigned to it.

has_dependent :: (0..n) DIM_attribute_row :: **based_on** :: (1..1)

Each R-MIM attribute is based on one MIM attribute.

is_state_attribute_for :: (0..1) Subject_class :: **has_state_attribute** :: (1..1)

The state attribute of a class contains a value indicating the current state of the class. In the event that the class has concurrent states, the attribute must be a set of state values.

had_V23_type :: (1..1) V23_data_type :: **typed** :: (0..n)

Provides an indication of the data type used in Version 2.x for a particular attribute, if such prior usage has been identified.

based_on :: (0..n) V23_fields :: **is_source_for** :: (0..n)

Provides a linkage for an information model attribute to its equivalent version 2.x field, if such linkage exists and has been identified.

stems_from :: (0..n) V23_segments :: **source_of** :: (0..n)

Many attributes are traced to equivalent content in HL7 Version 2.x. This connection is secondary to the path that traces an attribute to an HL7 field to a segment. It is provided for modelers who wish to specify particular segments for information model attributes.

Attributes of: **Attribute**

description : DescriptiveText

A short informative description of the Class characteristic captured by the Attribute.

history : CompoundHx

This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

Note, the version data must fit within the range of the applicable versions for the class of which this element is a part.

inclusion : Boolean
An indication of whether the inclusion of the attribute is mandatory in all HL7 HMDs and messages. Setting the inclusion to mandatory in the information model is deprecated.

name : NameString
Singular nouns are used for attribute names. Attribute names are unique within the class they describe and within the set of attributes inherited by the class they describe. The terminal element of the name shall indicate the attribute type for the attribute.

repeatability : Boolean
Indicates whether this attribute may repeat when it is included in the message structure of a hierarchical message description. The default is false, non-repeating.

sequence : Integer
Identifies the relative sort order of the attribute within the containing class. Lower numbers appear first. In the circumstance where two attributes within a class have the same number, sorting will occur based on alphabetically ascending attribute name.

Class: **Attribute_domain_constraint**

Associated with: **Attribute**
Vocabulary_concept

Description of: **Attribute_domain_constraint**
Constrains a coded attribute to a particular vocabulary domain.

For any class, the special attribute status_cd has as its domain all of the states of the class.

Associations for: **Attribute_domain_constraint**

constrains :: (0..n) Attribute :: **constrained_by** :: (0..1)

links_domain :: (1..1) Vocabulary_concept :: **is_constraint** :: (0..n)

Attributes of: **Attribute_domain_constraint**

strength : Enumerated
The strength of the constraint is either CWE (coded with exceptions) or CNE (coded, no exceptions). If no value is given, CWE is the default.

Class: **Attribute_type**

Associated with: **Attribute**
Data_type

Description of: **Attribute_type**
An indication of the form of the attribute, and of its usage. The use of attribute type words in attribute names aids in creating uniformity in the names, helps to avoid unintentional redundancy, and adds clarity to the model.

Associations for: **Attribute_type**

types :: (0..n) Attribute :: **is_of_type** :: (1..1)
A reference between an attribute and its attribute type. The attribute type code must also be the terminal component of the attribute name.

implemented_by :: (0..1) Data_type :: **implements** :: (1..n)
Each Attribute type may be implemented by one or more data types.

Attributes of: **Attribute_type**

history : CompoundHx
This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

name : NameString
The full name of the attribute type.

short_name : NameString
The representation of the attribute type that is appended to the name of the attribute to indicate the attribute type.

usage : DescriptiveText
A brief description of the intended usage of this attribute type.

Class: **Class**

Supertype of: **Subject_class**

Is part of: **Model**

Composite of: **Attribute**

Associated with: **DIM_class_row**
Relationship
Relationship
Subject_area
Subject_area

Description of: **Class**
An abstraction of a set of real-world things (objects) such that all of the objects have the same characteristics and all instances are subject to and conform to the same rules. Classes are the people, places, roles, things, and events about which information is kept.

Composition for: **Class**

has (0,n) :: Attribute :: **in (1,1)**
The relationship between attributes and the classes of which they are a part.

in (1,1) :: Model :: **has (0,n)**
The relationship between classes and the models of which they are a part.

Associations for: **Class**

has_dependent :: (0..n) DIM_class_row :: **is_based_on** :: (1..1)

is_destination :: (0..n) Relationship :: **has_destination** :: (1..1)
A reference to the class that is the target of the association.

is_source :: (0..n) Relationship :: **has_source** :: (1..1)
A reference to the class from which the association perspective is captured.

appears_in :: (0..n) Subject_area :: **includes** :: (1..n)
The linkage between a Subject area and each of the Classes that are in that Subject area.

primarily_resides_in :: (0..1) Subject_area :: **holds** :: (1..n)
The linkage between a Class and the Subject area that is its primary residence. This must be established if a Class resides in more than one Subject area.

Attributes of: **Class**

description : DescriptiveText
A short informative statement that allows one to tell, with certainty, whether a particular real world thing is an instance of the Class as conceptualized in the information model.

history : CompoundHx
This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

isAbstract : Boolean
This variable indicates whether or not this Class is an abstract class. An abstract class is a class that can not be instantiated and is customarily the generalization class in a generalization/specialization structure.

name : NameString
The Classes in the information model are given a unique name. The Class name is a singular noun or noun phrase.

Class: **Code_system**

Associated with: **Coded_term**
Vocabulary_concept

Description of: **Code_system**
A system is a published code system by an organization, that defines it, publishes it, maintains it, and thus guarantees for its usefulness and continuity.

If the source system includes named value sets or specifically identified tables, each of these is considered as an independent code source.

Rationale: This denormalization was undertaken because building a separate set of classes for the internal value sets and tables of a source system is very difficult, since there is not a single standard structure for such systems.

Associations for: **Code_system**

has_terms :: (0..n) Coded_term :: **is_part_of** :: (1..1)
Each coded term comes from a single code system.

is_basis_for :: (0..n) Vocabulary_concept :: **has_basis_in** :: (0..1)
Each value set is based on a single code system.

Attributes of: **Code_system**

organization : String
The name of the organization that maintains the code system. Examples are WHO, College of American Pathologists, ISO, IANA, and HL7.

set_name : String

system_version : String
The code system version for the term.

systemName : String
Code System - indicates the name of the code system. e.g. ICD, ICPM, ICPC, SNOMED, 639-1, MIME types, ...

tableID : String
Table Identifier - the table number or identifier (if one exists) in the source vocabulary where this concept originated.

Class: **Coded_term**

Is part of: **Domain_version**

Associated with: **Code_system**
Observation_id_link
Vocabulary_concept

Description of: **Coded_term**
This table shows the relationship between HL7 concepts and codes and descriptions from specific vocabularies. It allows a user of HL7 to see how the concepts in a given domain can be represented within a specific vocabulary/coding system. It links one specific concept to one term in a particular code system.

Composition for: **Coded_term**

in_version (1,1) :: Domain_version :: **has (0,n)**

Associations for: **Coded_term**

is_part_of :: (1..1) Code_system :: **has_terms** :: (0..n)
Each coded term comes from a single code system.

linked_to_set :: (0..n) Observation_id_link :: **links_obs_term** :: (1..1)
Each linked observation identifier is drawn from a particular code system. Initially these will all be drawn from LOINC.

represents :: (0..1) Vocabulary_concept :: **is_represented_by** :: (0..n)
Links a coded term to a single concept.

Attributes of: **Coded_term**

code : String
Code - the text string used within the code system to identify this concept.

definition : DescriptiveText
A textual representation of the meaning of this entry as it is represented in the coding system from which it came.

Any term may have a definition stored. For the terms that are referenced from external coding systems, the definition will not be included..

These terms may be used to maintain HL7 code tables, in which case, the definition is mandatory.:

displayName : String
A textual name for the term.

edit_note : DescriptiveText
A general purpose textual field for recording specific information about this code, or details about the rationale for creating, modifying, or deleting this particular table entry.

language_cd : Enumerated
ISO Language - the language (English, German, French, Italian, etc.) that is used in the description. The language codes come from the vocabulary domain of Language and are specified by ISO 639:1988 (E/F) Code for the representation of names of languages.

mapping_quality : Enumerated
Map Relationship - the closeness or quality of the mapping between the HL7 concept (as represented by the HL7 concept identifier) and the source coding system. The values for the relationship come from the MapRelationship domain and are patterned after the similar relationships used in the UMLS Metathesaurus. Examples of values for map relationship are: exact, broader than, narrower than, etc. Because the HL7 coding system is the master reference for the definition of the concept, the map relationship for HL7 coding system entries will always be Exact.

status : CodedElement
HL7 Status - the status of this entry within this table. The values for Status come from the vocabulary domain EditStatus. Some values for status are Proposed, Rejected, Active, Obsolete, and Inactive.

system_version_in : String
The code system version in which this term was first introduced.

system_version_out : String
The code system version in which this term was first deleted or changed.

version_out : Integer
Code System Vout - the version number of the code system at the time the code was retired or deleted.

Class: **Communication_wrapper**

Is part of: **Model**

Associated with: **Interaction_type**
Message_type

Description of: **Communication_wrapper**
A message type intended to act as the outer communication wrapper when transmitting a message.

Composition for: **Communication_wrapper**

in (1,1) :: Model :: **has (0,n)**

Associations for: **Communication_wrapper**

may_wrap_interactions_of_type :: (1..1) Interaction_type :: **supports_wrapper** :: (1..1)
Communication wrappers may wrap interactions of a given type.

acts_as :: (1..1) Message_type :: **implemented_by** :: (0..1)
Communication wrappers have their content communicated by a message type.

Attributes of: **Communication_wrapper**

history : CompoundHx
This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

identifier : IdentifierString
A unique identifier used to reference the communication_wrapper.

name : NameString
A unique name for the interaction type.

Class: **Composite_aggregation**

Subtype of: **Association**

Description of: **Composite_aggregation**
An association between classes that depicts the relationship between a composite aggregate class and its component parts. The source_multiplicity of the composite aggregation is constrained to be "1..1".

Class: **Composite_data_type**

Subtype of: **Data_type**

Composite of: **Data_type_component**

Description of: **Composite_data_type**
A composite data type consists of one or more named and typed components.

Composition for: **Composite_data_type**

contains (1,n) :: Data_type_component :: **belongs_to (1,1)**

Class: **Concept_relationship**

Is part of: **Domain_version**

Associated with: **Vocabulary_concept**
Vocabulary_concept

Description of: **Concept_relationship**
There are three different types of concept-to-concept relationships. Each is identified by a relationship type code, and each relates a contained or included concept to a container concept.

The Value Set Relationship Table records the relationship between HL7 maintained value sets and the concepts that are values within the value set. An HL7 maintained value set can be composed from individual concepts, other value sets, or both.

Composition for: **Concept_relationship**

in_version (1,1) :: Domain_version :: **has (0,n)**

Associations for: **Concept_relationship**

has_container :: (1..1) Vocabulary_concept :: **is_containing_concept** :: (0..n)

links_content :: (1..1) Vocabulary_concept :: **is_contained_concept** :: (0..n)

Attributes of: **Concept_relationship**

edit_note : DescriptiveText
Editor's notes for the domain. A general purpose textual field for recording specific information about the entry, or details about the rationale for modifying this particular table entry.

generality : Enumerated
Generality - indicates whether the concept that is the target should be interpreted as itself, or whether it should be expanded to include its child concepts, or both when it is included in the source domain/valueset.. Possible values are: Abstract, Specializable, and Leaf. Leaf means that only the concept itself is included in the domain. Abstract means that only descendents of the concept are included in the domain, and Specializable means that the concept itself and its descendents are included in the domain. The values for the Generality column come from the ConceptGenerality domain.

operator : Enumerated
Operator - the name of the relationship that exists between the value set and the concept that is being included. In the initial implementation, the only relationship is Include, which means that the domain contains the concept that is linked as the content. Relationship names come from the ValueSetOperator domain.

sequence : Integer
Sequence number operates within a given version to sequence the operations for a given "container" value set. So long as the only operator used is "include," the sequence is not needed. However, the "exclude" and "union" operators will produce sequence dependent results if they are used in conjunction with other operators such as "include."

status : CodedElement
Status - the status of the item. The values for Status come from the vocabulary domain EditStatus. Some values for status are Proposed, Rejected, Active, Obsolete, and Inactive.

type_cd : Enumerated
The relationship types are:

DV - Domain contains value set. In this relationship, the container is a domain and the target is a value set whose concepts provide the content of the domain.

VI - Value set inclusion. In this relationship the container is a value set and the target is either a value set or concept to be part of the content in the value set. The nature of the inclusion is determined by the generality code.

version_out : Integer
Vout - the version number of the domain specification database at the time this entry was updated or deleted. A blank Vout value means that the entry continues to exist in the current version of the table.

Class: **Control_event**

Associated with: **Interaction**
Message_type

Description of: **Control_event**

A control event provides information about an action in the health-care domain which causes information to be exchanged or which must be tracked and have a record of the activity persisted.

Associations for: **Control_event**

wraps_interaction :: (0..n) Interaction :: **has_control_event** :: (1..1)
:: (0..n) Interaction :: :: (1..1) Control events wrap the payload message type for interactions.
implemented_by :: (1..1) Message_type :: acts_as :: (0..1) Control events have their content represented by a message type.

acts_as :: (1..1) Message_type :: **implemented_by** :: (0..1)
Control events have their content represented by a message type.

Attributes of: **Control_event**

history : CompoundHx
This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

identifier : IdentifierString
A unique identifier used to reference the communication_wrapper.

name : NameString
A unique name for the interaction type.

Class: **Data_type**

Supertype of: **Composite_data_type**
Generic_type_parameter

Is part of: **Model**

Associated with: **Attribute**
Attribute_type
Data_type_category
Data_type_component
Data_type_generalization
Data_type_generalization
Generic_type_parameter
Generic_type_parameter
Generic_type_parameter

Description of: **Data_type**
Datatypes are used to express the type of an attribute or of a data type component. A data type may be composite, primitive, an alias or a generic type parameter.

A generic type parameter contains a parameter that is part of the definition of a generic data type. Each generic type parameter is part of the definition for a single generic type.

A composite data type contains one or more components.

An alias provides an alternative name, alternate type code and additional description for another data type.

A primitive data type is a data type that is defined entirely by its specification. A primitive data type may have generic type parameters.

A generic data type is a type that has one or more generic type parameters. It provides a pattern for instantiating a specific, usually composite, data type.

Composition for: **Data_type**

in (1,1) :: Model :: **has (0,n)**
The relationship between data types and the models in which they are first defined.

Associations for: **Data_type**

types :: (0..n) Attribute :: **is_of_type** :: (0..1)
A link between an attribute and the datatype that has been assigned to it.

implements :: (1..n) Attribute_type :: **implemented_by** :: (0..1)
Each Attribute type may be implemented by one or more data types.

resides_in :: (0..n) Data_type_category :: **contains** :: (0..n)

types :: (0..n) Data_type_component :: **is_of_type** :: (1..1)
Each component is linked to a single type either directly or through a generic type parameter.

is_subtype :: (0..n) Data_type_generalization :: **has_subtype** :: (1..1)
Each data type generalization includes a single sub-type.

is_supertype :: (0..n) Data_type_generalization :: **has_supertype** :: (1..1)
Each data type generalization provides sub-types for a single super-type..

allowed_for :: (0..n) Generic_type_parameter :: **has_allowed_types** :: (0..n)
Determines the set of types that a generic type parameter may implement.

defined_by :: (0..n) Generic_type_parameter :: **defines** :: (1..1)
Relationship between a Generic Type Parameter and the Generic type for which it is a parameter.

types :: (0..n) Generic_type_parameter :: **has_instance_type** :: (0..1)
This relationship defines the particular instantiation type for a generic instance.

Attributes of: **Data_type**

description : DescriptiveText
A detailed description or specification for the data type. All such descriptions are assumed to also reference a broader specification of data types.

history : CompoundHx
This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

is_internal : Boolean
A data type may be defined as being "internal". An internal type is used only to define other composite data types. Internal types shall not be used in messages. For example, we define a type Binary that contains pure raw data bits, and that is used only by Multimedia Enabled Free Text.

name : NameString
The formal name for the data type.

type_code : String

The formal code assigned by the Control/Query Committee for this datatype. This is the representation of the data type that appears as the data type for attributes of the information model and data type components in the data type model.

Class: **Data_type_category**

Is part of: **Model**

Associated with: **Data_type**
Data_type_category
Data_type_category
HL7_committee

Description of: **Data_type_category**

A data type category collects data types that represent similar real world concepts, or are represented in a similar fashion.

Composition for: **Data_type_category**

in (1,1) :: Model :: **has (0,n)**

The relationship between data type categories and the models of which they are a part.

Associations for: **Data_type_category**

contains :: (0..n) Data_type :: **resides_in** :: (0..n)

is_nested_in :: (0..1) Data_type_category :: **nests** :: (0..n)

nests :: (0..n) Data_type_category :: **is_nested_in** :: (0..1)

maintained_by :: (1..1) HL7_committee :: **maintains** :: (0..n)

Attributes of: **Data_type_category**

description : DescriptiveText

Th description of the data type category expresses the unifying concept that causes a set of data types to be included in this category.

history : CompoundHx

This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

name : NameString

The name given to the data type category.

Class: **Data_type_component**

Is part of: **Composite_data_type**

Associated with: **Data_type**

Description of: **Data_type_component**

A component is an element of a data type that may be valued when the data type is used in HL7 communications. The component takes its type from a data type that is any of - primitive, composite, or generic type parameter.

A component of a composite data type is like a variable, i.e. it has a name and a type. The type can be declared to be included by reference instead of by value. This is useful if you know such a component mentions an instance that is already mentioned elsewhere in the communication. In languages such as Java, where objects are always handled through references this does not make any difference.

Composition for: **Data_type_component**

belongs_to (1,1) :: Composite_data_type :: **contains (1,n)**

Associations for: **Data_type_component**

is_of_type :: (1..1) Data_type :: **types** :: (0..n)
Each component is linked to a single type either directly or through a generic type parameter.

Attributes of: **Data_type_component**

description : DescriptiveText
The description of a component should include its role within the composite of which it is a part and its relationships, if any, to other components of the same composite.

history : CompoundHx
This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

is_reference : Boolean
The component type can be declared to be included by reference instead of by value. This is useful if you know such a component mentions an instance that is already mentioned elsewhere in the communication. In languages such as Java, where objects are always handled through references this does not make any difference.

name : NameString
The formal name of the data type component.

Class: **Data_type_generalization**

Associated with: **Data_type**
Data_type

Description of: **Data_type_generalization**
Types can maintain an inheritance relationship with each other. We explicitly allow (and use) "multiple inheritance". However, we do use inheritance as a way to specialize subtypes from general super-types. Rather we go the other way. Abstract generalized types are used to categorize the concrete types in different ways. Thus one can get hold of all types that have a certain property of interest.

For instance, we define the generalized type Quantity to subsume all quantitative types. This is used to define one type Ratio as a ratio of any two quantities.

Similarly, we define a data type Interval that is a continuous subset of any type with an order relation. All types with an order relation are subsumed under OrderedType. Note that not all quantities are ordered (e.g. vectors are not) and there may be non-quantities that have an order relationship (ordinals, e.g. military ranks).

Associations for: **Data_type_generalization**

has_subtype :: (1..1) Data_type :: **is_subtype** :: (0..n)
Each data type generalization includes a single sub-type.

has_supertype :: (1..1) Data_type :: **is_supertype** :: (0..n)
Each data type generalization provides sub-types for a single super-type..

Attributes of: **Data_type_generalization**

description : DescriptiveText
A statement of the nature of the generalization relationship.

history : CompoundHx
This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

stereotype : Enumerated
The DT generalization is of one of two stereotypes. "Extends" implies that the sub-type has a "value" that is of the super type and then adds components of its own. "Restricts" implies that the sub type is a constrined version of the super type.

Class: **Design_annotation**

Associated with: **Application_role**
Interaction
Note
Trigger_event

Associations for: **Design_annotation**

for :: (0..1) Application_role :: **has** :: (0..n)

for :: (0..1) Interaction :: **has** :: (0..n)

has_content :: (1..1) Note :: **content_for** :: (0..n)

for :: (0..1) Trigger_event :: **has** :: (0..n)

Attributes of: **Design_annotation**

target_type : Enumerated
Indicates the type of element that is the target or focus of the annotation.

Class: **Design_category**

Is part of: **Model**

Associated with: **Application_role**
Design_category
Design_category
HL7_committee
Interaction
Storyboard
Trigger_event

Description of: **Design_category**
A major category of information represented in the design of HL7 information structures model. A category allows portions of a large model to be viewed as a whole thereby eliminating some complexity involved in understanding a large model.

As these categories are nested, they create a hierarchy of design information that supports the publication and interpretation of the standards.

Composition for: **Design_category**

in (1,1) :: Model :: **has (0,n)**
The relationship between interaction model categories and the models of which they are a part.

Associations for: **Design_category**

includes :: (0..n) Application_role :: **defined_in** :: (1..1)

nested_in :: (0..1) Design_category :: **nests** :: (0..n)
Interaction model categories may be nested.

nests :: (0..n) Design_category :: **nested_in** :: (0..1)
Interaction model categories may be nested.

maintained_by :: (0..1) HL7_committee :: **maintains** :: (0..n)

includes :: (0..n) Interaction :: **defined_in** :: (1..1)

includes :: (0..n) Storyboard :: **defined_in** :: (1..1)

includes :: (0..n) Trigger_event :: **defined_in** :: (1..1)

Attributes of: **Design_category**

description : DescriptiveText
Short informative text describing the scope of the content for the interaction model category.

For sections and subsections this may include a description of boundaries and the relationship itself and other subsections.

For domains, it may include a description of boundaries and the relationship of itself and other modeling domains. From reading the scope it should be clear what type of interactions, application roles, trigger events and information models will be defined within the domain.

history : CompoundHx
This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

identifier : IdentifierString
A unique identifier for this category. The structure for this identifier will be defined by HL7 for each "level" of category.

level : Enumerated
As design categories are nested, they form a hierarchy whose levels are designated by this attribute. Examples levels are:

Section - A categorization of the deliverables of the HL7 organization. Identifies a high-level content area in which work is performed by HL7 members. It is usually comprised of one or more subsections.

Subsection - A sub-categorization of information within a section. Reflects one of the primary focal areas of the HL7 organization. It contains individual domains.

Domain - A major focus area around which messages and other HL7 deliverables are built. A modeling domain is the primary unit of work for a technical committee. The modeling domain represents a cohesive view of an area of healthcare, including the events that can occur, the types of applications used and the information exchanged and recorded.

name : NameString
The name given to the interaction model category. The identifier for the committee defining this category is prepended to the name as Cnn.

Class: **Design_information_model**

Is part of: **Model**

Composite of: **DIM_row**

Associated with: **Design_information_model**
Design_information_model

Description of: **Design_information_model**

The Design Information Model (DIM) is the representation of the information that must be recorded, managed and/or transmitted to satisfy the requirements of a particular domain or set of specifications to be published by HL7.

The DIM has two major forms that are distinguished by the models from which the DIM draws its classes, attributes, relationships, etc. These are a simple Design Information Model DIM, and the Constrained Design Information Model IC-DIM).

A DIM is the representation of the information that needs to be recorded and transmitted to satisfy the requirements of a particular domain. The content for the DIM is drawn directly from the containing information model. The information is represented through the cloning and restricting of the classes, attributes and associations of the containing Model. That is classes, attributes and associations from a Model may be represented more than once in a DIM. This is done to allow the messages to be tailored to the specific needs of different instances of a class. For example, the Person class has roles for both patients and doctors. By and large, the attributes and associations of Person that are important to the patient role are different from those important to the doctor role. Cloning allows these differences to be represented explicitly in the DIM.

When a class is cloned, the clone must be given a unique name. If there are multiple clones derived from the same Model class, each clone may be constrained independently. Constraints involve: removing attributes, tightening association cardinality (increasing minimum/decreasing maximum), discarding associations, constraining datatypes, making an attribute or association mandatory, specifying association or attribute conformance, reducing vocabulary domains, specifying a default or fixed value for an attribute and adding constraint notes.

The DIM is displayed both as a UML diagram, and in a two-level tabular format in which each class and clone is a primary row, and the attributes and associations of those classes comprise the second-level rows. Selected attributes of the meta-model provide 'lay-out' information for the tabular format.

A Constrained Design Information Model (C-DIM) is derived from a DIM or another C-DIM. That is, C-DIM is design information model that is developed with the further constraint that its content, while being drawn from the containing information model, may only include elemtns that are a constrained subset of the DIM or C-DIM from which it is derived.

Composition for: **Design_information_model**

contains (1,n) :: DIM_row :: **part_of (1,1)**

is_part_of (1,1) :: Model :: **contains (0,n)**

Associations for: **Design_information_model**

derives_from :: (0..1) Design_information_model :: **further_constrains** :: (0..n)
Each DIM or C-DIM may be further constrained by multiple C-DIMs that derive from it.

further_constrains :: (0..n) Design_information_model :: **derives_from** :: (0..1)
Each DIM or C-DIM may be further constrained by multiple C-DIMs that derive from it.

Attributes of: **Design_information_model**

first_node_id : IdentifierString
Identifies the first class node in the tabular display of the R-MIM.

history : CompoundHx
A compound data element that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

identifier : IdentifierString
A unique identifier for the R-MIM.

name : NameString
A unique formal name for the DIM.

type : Enumerated
A Design information model may be of one of two types.

A simple design information model (designated DIM) may draws its information and creates its clones from the containing Model without further restrictions.

A constrained design information model (designated C-DIM) applies further constraints to an existing DIM or C-DIM upon which it is dependent.

walkThrough : DescriptiveText
A description explaining use and meaning of the information constructs represented in a DIM.

Class: **DIM_attribute_domain_constraint**

Associated with: **DIM_attribute_row**
Vocabulary_concept

Description of: **DIM_attribute_domain_constraint**
Constrains a coded DIM attribute row to a particular vocabulary domain.

For any class, the special attribute status_cd has as its domain all of the states of the class.

Associations for: **DIM_attribute_domain_constraint**

constrains :: (0..n) DIM_attribute_row :: **constrained_by** :: (0..1)

links_domain :: (1..1) Vocabulary_concept :: **is_constraint** :: (0..n)

Attributes of: **DIM_attribute_domain_constraint**

strength : Enumerated

The strength of the constraint is either CWE (coded with exceptions) or CNE (coded, no exceptions). If no value is given, CWE is the default.

Class: **DIM_attribute_row**

Subtype of: **DIM_row**

Associated with: **Attribute**
DIM_attribute_domain_constraint
HMD_attribute_row

Description of: **DIM_attribute_row**
Expresses the presence of selected attributes in the DIM.

Associations for: **DIM_attribute_row**

based_on :: (1..1) Attribute :: **has_dependent** :: (0..n)
Each R-MIM attribute is based on one MIM attribute.

constrained_by :: (0..1) DIM_attribute_domain_constraint :: **constrains** :: (0..n)

defines :: (0..n) HMD_attribute_row :: **defined_by** :: (1..1)

Class: **DIM_class_row**

Subtype of: **DIM_row**

Associated with: **Class**
DIM_relationship_row
DIM_row
DIM_row
HMD_class_row

Description of: **DIM_class_row**
Expresses the presence of selected classes or clones thereof in the DIM.

Associations for: **DIM_class_row**

is_based_on :: (1..1) Class :: **has_dependent** :: (0..n)

is_related_by :: (0..n) DIM_relationship_row :: **has_distal_class** :: (0..1)

is_active_parent :: (0..n) DIM_row :: **has_active_parent** :: (1..1)
Shows the active parent relationship for an inherited row. Reflects the combined effects of inheritance and cloning.

is_true_parent :: (0..n) DIM_row :: **has_true_parent** :: (1..1)
Establishes the true parent for each association and attribute. Is required because the act of cloning precludes determining this association through the information model or MIM.

defines :: (0..n) HMD_class_row :: **defined_by** :: (1..1)

Attributes of: **DIM_class_row**

first_attribute_row_id : IdentifierString

Pointer to the first child attribute row for this class row.

first_relation_row_id : IdentifierString
Pointer to the first child relationship row for this class row.

Class: **DIM_notation**

Associated with: **DIM_row**
Note

Associations for: **DIM_notation**

annotates :: (1..1) DIM_row :: **has_notation** :: (0..n)

links_note :: (1..1) Note :: **is_notation** :: (0..n)

Attributes of: **DIM_notation**

type : Enumerated
Indicates the type of the note. Types include:

RV : Required value

CP : Conditional Presence

CN : Constraint

DM : Domain

CT : Comment

Class: **DIM_other_row**

Subtype of: **DIM_row**

Associated with: **DIM_row**
HMD_other_row

Description of: **DIM_other_row**
Expresses the presence of special rows in the DIM. There is one type of such additional row:

stc : Represents the presence of a sub-component of a data type in the message. These components are exposed in to allow the expression of constraints against them.

Associations for: **DIM_other_row**

has_parent :: (1..1) DIM_row :: **is_parent** :: (0..n)
Each 'other' node arises as a result of some other node, its parent.

defines :: (0..n) HMD_other_row :: **defined_by** :: (1..1)

Attributes of: **DIM_other_row**

otherType : Enumerated

Coded value for the type of the other row. See definition of the RMIM_other_row class for the code values and their meaning.

Class: **DIM_relationship_row**

Subtype of: **DIM_row**

Associated with: **DIM_class_row**
DIM_relationship_row
DIM_relationship_row
HMD_relationship_row
Relationship

Description of: **DIM_relationship_row**

Expresses the presence of selected association nodes (UML roles) in the DIM. Each relationship in the RIM and DIM produces two rows in the tabular DIM, one for the appearance of each end of the relationship in one of the DIM classes or clones.

Associations for: **DIM_relationship_row**

has_distal_class :: (0..1) DIM_class_row :: **is_related_by** :: (0..n)

has_other_half :: (1..1) DIM_relationship_row :: **is_other_half** :: (0..1)

Each RMIM relationship row may be paired with a second such row to comprise a complete relationship.

is_other_half :: (0..1) DIM_relationship_row :: **has_other_half** :: (1..1)

Each RMIM relationship row may be paired with a second such row to comprise a complete relationship.

defines :: (0..n) HMD_relationship_row :: **defined_by** :: (1..1)

is_based_on :: (1..1) Relationship :: **has_dependent** :: (0..n)

Attributes of: **DIM_relationship_row**

blocked : Boolean

Expresses whether this half-relationship can be followed in building an HMD. When the R-MIM is diagrammed in UML, the presence of a blocked path is shown by making the UML role for the other end of the relationship unnavigable.

Class: **DIM_row**

Is Abstract Class

Supertype of: **DIM_attribute_row**
DIM_class_row
DIM_other_row
DIM_relationship_row
DIM_state_row

Is part of: **Design_information_model**

Associated with: **DIM_class_row**
DIM_class_row
DIM_notation
DIM_other_row

DIM_row
DIM_row

Description of: **DIM_row**

The DIM is modeled by the Rows that make up its tabular expression.

Composition for: **DIM_row**

part_of (1,1) :: Design_information_model :: **contains (1,n)**

Associations for: **DIM_row**

has_active_parent :: (1..1) DIM_class_row :: **is_active_parent** :: (0..n)

Shows the active parent relationship for an inherited row. Reflects the combined effects of inheritance and cloning.

has_true_parent :: (1..1) DIM_class_row :: **is_true_parent** :: (0..n)

Establishes the true parent for each association and attribute. Is required because the act of cloning precludes determining this association through the information model or MIM.

has_notation :: (0..n) DIM_notation :: **annotates** :: (1..1)

is_parent :: (0..n) DIM_other_row :: **has_parent** :: (1..1)

Each 'other' node arises as a result of some other node, its parent.

derives_from :: (0..1) DIM_row :: **further_constrains** :: (0..n)

Each row of a C-DIM further constrains a specific row of the "parent model" (a DIM or C-DIM) from which it derives. Moreover, each row of a C-DIM must derive from some row in the "parent model."

further_constrains :: (0..n) DIM_row :: **derives_from** :: (0..1)

Each row of a C-DIM further constrains a specific row of the "parent model" (a DIM or C-DIM) from which it derives. Moreover, each row of a C-DIM must derive from some row in the "parent model."

Attributes of: **DIM_row**

cardinality : MultiplicityString

Expresses the minimum and maximum number of occurrences for an association or an attribute in the context of the R-MIM.

conformance : Enumerated

Expresses the constraints on this row for conformance testing. Possible values are:

R: required for conformance

(blank): not required for conformance

NP: not permitted to appear in his message variant (only used in message row controls, not in R-MIM).

constraint : DescriptiveText

Captures constraints and notes for a given row in the R-MIM and HMD. The type of note and its contents must both be expresses. The types provided for include:

Required value : states a value and, in the presence of repetition, how many times the value can/must appear

Conditional Presence : states a value that must or must not be present based on the value of another element or sub-component that is higher in the HMD

Constraint : a verbal expression of a constraint

Domain : a domain specification, as described in the vocabulary chapter

Comment : any general comment; this label should not be used for items that can be described with any of the other labels.

default_update_mode : String

A coded value drawn from the possible modes of updating that occur when an attribute is received by a system that already contains values for that attribute. The update modes and their codes are:

R : replace (this is the default)

D : delete

I : ignore

NA : not applicable

V : verify: confirm that it exists

K : key: when creating an element store it; when updating an element confirm that it exists.

The following codes apply when updating individual items in a set:

ESA : edit set: add item

ESC : edit set: change item

ESD : edit set: delete item

ESAC : edit set: add or, if the item exists, change item

default_value : DescriptiveText

The default value for this attribute. If this field is blank or Null, the default is 'NULL'. If the field contains 'No" then no default specified. Otherwise, the field contains the value of the default and if the value is an integer it must be enclosed in quotes.

history : CompoundHx

A compound data element that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

mandatory : Boolean

If this attribute has a value of "True" then this message element must have a non-null value in order for the receiver to process the message.

name : NameString

The name of the row in the R-MIM. Rows representing attributes should not be renamed.

next_sibling_ID : IdentifierString

Points to the next sibling node in the tabular R-MIM.

previous_sibling_id : IdentifierString

Points to the previous sibling node in the tabular R-MIM.

short_name : NameString

A shortened version of the name for the row.

update_mode_set : String
A set of utterances from the list of values for 'default_update_mode.'. The sender may change the update mode instance by instance to any of the values in this list.

Class: **DIM_state_row**

Subtype of: **DIM_row**

Associated with: **State**

Description of: **DIM_state_row**
Expresses the presence of selected states in the DIM.

Associations for: **DIM_state_row**

is_based_on :: (1..1) State :: **has_dependent** :: (0..n)

Class: **Domain_version**

Composite of: **Coded_term**
Concept_relationship
Observation_id_link
Vocabulary_concept

Description of: **Domain_version**
Captures each update of the vocabulary domain tables in a version, including the when the editing took place, who performed it, and comments as to what was done.

Composition for: **Domain_version**

has (0,n) :: Coded_term :: **in_version (1,1)**

has (0,n) :: Concept_relationship :: **in_version (1,1)**

has (0,n) :: Observation_id_link :: **in_version (1,1)**

has (0,n) :: Vocabulary_concept :: **in_version (1,1)**

Attributes of: **Domain_version**

comment : DescriptiveText
Comment - a summary of why the edits were made, and what was done.

edit_dttm : DateTime
The date and time that the edit session began

editor_id : String
Who - an identifier of the person who actually edited the database. People are identified by reference to a directory where each person is assigned a unique identifier, and where locating information about the person can be found.

for_whom_id : String

For whom - an identifier of the person or organization for whom the edit was made. For example, a person may be making edits as authorized by the Vocabulary TC, or on behalf of the TC for which they are the facilitator. People and organizations are identified by reference to a directory where each person or organization is assigned a unique identifier.

version : Integer

Version - the version number of the edit session. This number is incremented by 1 each time a new edit session takes place. The version number is used as the value of Vin and Vout as appropriate to track which table entries in the domain specification database were added, modified, or deleted during the session.

For specific tables, Vin provides the version number of the domain specification database at the time that this entry was added or updated.

Class: **Generalization_relationship**

Subtype of: **Relationship**

Description of: **Generalization_relationship**

Generalization is a relationship between a class and subtypes of the class. A supertype can be associated with more than one subtype. Each of the subtypes associated with a single supertype is mutually exclusive. In HL7 information models, a subtype may be associated with only one supertype. The hierarchy or lattice of generalizations is called a generalization relationship. The subtype inherits the attributes, and associations, and of all of its supertypes.

The supertype is the "source" of the relationship, and the subtypes are the "destinations" of this relationship.

Class: **Generic_type_parameter**

Subtype of: **Data_type**

Associated with: **Data_type**
Data_type
Data_type

Description of: **Generic_type_parameter**

A generic type parameter contains a parameter that is part of the definition of a generic data type. Each generic type parameter is part of the definition for a single generic type.

The most common form of generic type parameter provides an instantiation type, drawn from two or more types.

Other generic type parameters constrain the generic type, such as the collection type and multiplicity for a Collection.

Associations for: **Generic_type_parameter**

defines :: (1..1) Data_type :: **defined_by** :: (0..n)

Relationship between a Generic Type Parameter and the Generic type for which it is a parameter.

has_allowed_types :: (0..n) Data_type :: **allowed_for** :: (0..n)

Determines the set of types that a generic type parameter may implement.

has_instance_type :: (0..1) Data_type :: **types** :: (0..n)

This relationship defines the particular instantiation type for a generic instance.

Attributes of: **Generic_type_parameter**

value : String
Establishes the content for a generic type parameter that defines a property of a generic type other than an instantiation type.

Class: **Hierarchical_message_description**

Is part of: **Model**

Composite of: **HMD_row**
Message_type

Description of: **Hierarchical_message_description**
A structure that completely defines the structure of a set of messages, and reflects the relationship of the elements of these messages to components of the Refined Message Information Model from which it derives and the Message Element Types that it defines or uses.

Composition for: **Hierarchical_message_description**

contains (1,n) :: HMD_row :: **is_part_of (1,1)**
A reference to the HMD that contains each HMD row.

contains (1,n) :: Message_type :: **is_part_of (1,1)**
Each message structure is contained in a single HMD.

in (1,1) :: Model :: **has (0,n)**
The relationship between hierarchical message descriptions and the models in which they are first defined.

Attributes of: **Hierarchical_message_description**

description : DescriptiveText
A short textual description of the messages covered in the HMD..

history : CompoundHx
This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

identifier : IdentifierString
Arbitrary identifier assigned by Technical Steering Committee. Committees may assign an interim identifier that starts with the committee's identifier, as "Cnn_<identifier>" so long as this composite identifier is unique.

name : NameString
The HMDs in the model are each given a unique, formal name.

shortName : NameString
The HMDs in the model are each given a short, common use name that is familiar to domain experts. The short name does not need to be unique.

Class: **HL7_committee**

Associated with: **Data_type_category**
Design_category
Model
Project
Subject_area
Use_case_category

Description of: **HL7_committee**

Unique identifier assigned to each of the Technical Committees and Special Interest Groups of the HL7 Working Group.

Associations for: **HL7_committee**

maintains :: (0..n) Data_type_category :: **maintained_by** :: (1..1)

maintains :: (0..n) Design_category :: **maintained_by** :: (0..1)

prepares :: (0..n) Model :: **prepared_by** :: (1..1)

Links a model to the committee that prepared it.

responsible_for :: (0..n) Project :: **responsibility_of** :: (1..1)

Establishes the relationship between committees and the projects for which that committee is responsible.

maintains :: (0..n) Subject_area :: **maintained_by** :: (0..1)

maintains :: (0..n) Use_case_category :: **maintained_by** :: (0..1)

Attributes of: **HL7_committee**

facilitator : String

Name of the individual who facilitates modeling for this committee.

id : IdentifierString

Assigned committee identifier.

mission : DescriptiveText

The approved mission statement or charter for this committee.

name : String

The name of the Technical Committee or Special Interest Group.

Class: **HMD_attribute_row**

Subtype of: **HMD_row**

Associated with: **DIM_attribute_row**

Description of: **HMD_attribute_row**

An attribute row represents a single attribute in the R-MIM.

Associations for: **HMD_attribute_row**

defined_by :: (1..1) DIM_attribute_row :: **defines** :: (0..n)

Class: **HMD_class_row**

Subtype of: **HMD_row**

Associated with: **DIM_class_row**

Description of: **HMD_class_row**

There is one class row in an HMD. This row is the root of the HMD.

Associations for: **HMD_class_row**

defined_by :: (1..1) DIM_class_row :: **defines** :: (0..n)

Class: **HMD_domain_constraint**

Associated with: **Message_row_control**
Vocabulary_concept

Description of: **HMD_domain_constraint**

Constrains a coded HMD attribute row to a particular vocabulary domain. Links each coded attribute in an HMD to the code domain that may be used to value it.

For any class, the special attribute status_cd has as its domain all of the states of the class. In an HMD domain specification, the special domain name '@state' can substitute for the domain name. If held is a valid state, <@state> and <@state - (held)> are valid domain specifications.

Associations for: **HMD_domain_constraint**

constrains :: (0..n) Message_row_control :: **has_domain** :: (0..1)

links_domain :: (1..1) Vocabulary_concept :: **is_constraint** :: (0..n)

Attributes of: **HMD_domain_constraint**

realm : String

May specify the realm of applicability for this attribute row.

strength : String

The strength of the constraint is either CWE (coded with exceptions) or CNE (coded, no exceptions). If no value is given, CWE is the default.

Class: **HMD_notation**

Associated with: **Message_row_control**
Note

Associations for: **HMD_notation**

annotates :: (1..1) Message_row_control :: **has_notation** :: (0..n)

links_note :: (1..1) Note :: **is_notation** :: (0..n)

Attributes of: **HMD_notation**

type : Enumerated

Indicates the type of the note. Types include:

RV : Required value

CP : Conditional Presence

CN : Constraint

DM : Domain

CT : Comment

Class: **HMD_other_row**

Subtype of: **HMD_row**

Associated with: **DIM_other_row**

Description of: **HMD_other_row**
Links an 'other' R-MIM row into a message.

Associations for: **HMD_other_row**

defined_by :: (1..1) DIM_other_row :: **defines** :: (0..n)

Class: **HMD_relationship_row**

Subtype of: **HMD_row**

Associated with: **DIM_relationship_row**
Message_type

Description of: **HMD_relationship_row**
An relationship row represents a single association, aggregation or inheritance relationship traversed in the R-MIM graph walk.

Associations for: **HMD_relationship_row**

defined_by :: (1..1) DIM_relationship_row :: **defines** :: (0..n)

typed_by_CMET :: (0..1) Message_type :: **types** :: (0..n)
Common message element type linkage. Class links the CMET defined by a particular message type (in a particular HMD) to the HMD row (allways a relationship row) that it types as a CMET.

Class: **HMD_row**

Supertype of: **HMD_attribute_row**
HMD_class_row
HMD_other_row
HMD_relationship_row

Is part of: **Hierarchical_message_description**

Associated with: **HMD_row**
HMD_row
Message_row_control

Description of: **HMD_row**

The rows of an HMD.

Composition for: **HMD_row**

is_part_of (1,1) :: Hierarchical_message_description :: **contains (1,n)**

A reference to the HMD that contains each HMD row.

Associations for: **HMD_row**

has_parent :: (1..1) HMD_row :: **is_parent** :: (0..n)

is_parent :: (0..n) HMD_row :: **has_parent** :: (1..1)

controlled_by :: (0..n) Message_row_control :: **controls** :: (1..1)

Each message row control controls the presence of one unsubsumed HMD row in the message structure of which the message row control is a part.

Attributes of: **HMD_row**

base_MET_name : String

Captures the base MET name that a particular row possesses (row is "of" type MET).

choice_set : String

Captures the set ("|" delimited) of METs that this row is "of."

history : CompoundHx

This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

MET_source : Enumerated

Indicates the source of the MET - reuse a definition from this HMD, CMET or data type.

nest_level : Integer

Indicates the nesting level of the row in the HMD.

Class: **Interaction**

Is part of: **Model**

Associated with:
Application_role
Application_role
Control_event
Design_annotation
Design_category
Interaction_sequence
Interaction_type
Message_type
Receiver_responsibility
Receiver_responsibility
Trigger_event

Description of: **Interaction**

An association between a specific message (information transfer), a particular trigger event that initiates or triggers the interaction, and the roles that send and receive the interaction. An interaction is a single, one-way transfer of information. Within itself, an interaction may not specify a return message. An

interaction may, however, establish a responsibility for the receiver of its message, and this responsibility may require that the receiver initiate a particular trigger event/interaction subsequent to the receipt. That follow-on interaction may have the effect of continuing or completing a transaction that requires two or more linked message exchanges.

Composition for: **Interaction**

in (1,1) :: Model :: **has (0,n)**
The relationship between interactions and the models of which they are a part.

Associations for: **Interaction**

received_by :: (1..1) Application_role :: **receives** :: (0..n)
A reference to the application role that is responsible for receiving the message involved in this interaction. The receiving role must be prepared to accept the message and to fulfill the receiver responsibility.

sent_by :: (1..1) Application_role :: **sends** :: (0..n)
The sending role has responsibilities to recognize the trigger event for the interaction and to cause the appropriate message to be sent.

has_control_event :: (1..1) Control_event :: **wraps_interaction** :: (0..n)
:: (0..n) Interaction :: :: (1..1) Control events wrap the payload message type for interactions.
implemented_by :: (1..1) Message_type :: acts_as :: (0..1) Control events have their content represented by a message type.

has :: (0..n) Design_annotation :: **for** :: (0..1)

defined_in :: (1..1) Design_category :: **includes** :: (0..n)

is_linked_by :: (0..n) Interaction_sequence :: **links** :: (1..1)

has_type :: (1..1) Interaction_type :: **is_type_for** :: (0..n)
:: (0..n) Interaction :: :: (1..1) Each interaction type may be instantiated by many interactions.

transfers :: (1..1) Message_type :: **transferred_by** :: (1..n)
Each interaction shall include a link to a single message structure that the interaction will transfer.

has_responsibility_option :: (0..1) Receiver_responsibility :: **is_responsibility_for** :: (1..1)
An application may have a responsibility to perform specific actions upon receiving an interaction. If receiver responsibilities are listed, the receiving application must perform the actions indicated in one of the identified responsibilities.

invokes :: (0..n) Receiver_responsibility :: **initated_by_receiver** :: (0..1)
A reference to an interaction that the receiver of the message must initiate once receipt of the message is acknowledged. This is an optional element in that there may no follow-on responsibility. Transactions can be established through a chain of receiver responsibilities for individual interactions.

initiated_by :: (1..1) Trigger_event :: **initiates** :: (0..n)
A reference to the trigger event that triggers or initiates this interaction.

Attributes of: **Interaction**

description : DescriptiveText
Provides a description of the data content of the interaction, usually expressed in terms of the class instances that are expected to be part of the message sent by the interaction.

history : CompoundHx
This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

identifier : IdentifierString
An identifier assigned to the interaction. The identifier should be unique within the scope of the model in which the interaction is defined. In HL7, committees manage the unique identifiers for their interactions, and concatenate the committee identifier as "Cnn_<identifier>."

name : NameString
A unique, formal name describing the interaction.

shortName : NameString
The short, meaningful name describing the interaction. Short names are not required to be unique.

Class: **Interaction_sequence**

Is part of: **Storyboard**

Associated with: **Interaction**

Description of: **Interaction_sequence**
Captures the sequence in which a particular interaction is included in a storyboard.

Composition for: **Interaction_sequence**

is_part_of (1,1) :: Storyboard :: **contains (0,n)**
Each storyboard is made up of a sequence of interactions, a sequence of use cases, or both.

Associations for: **Interaction_sequence**

links :: (1..1) Interaction :: **is_linked_by** :: (0..n)

Attributes of: **Interaction_sequence**

history : CompoundHx
This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

receiving_system_name : NameString
The name assigned to the system in the Storyboard which transmits this interaction. Necessary for constructing diagrams of the interaction.

sending_system_name : NameString
The name assigned to the system in the Storyboard which transmits this interaction. Necessary for constructing diagrams of the interaction.

sequence_number : Integer
The order in which the interaction participates in the Storyboard.

Class: **Interaction_type**

Associated with: **Communication_wrapper**
Interaction
Interaction_type

Interaction_type

Description of: **Interaction_type**

A categorization assigned to interactions to indicate how they behave and how they are used.

Associations for: **Interaction_type**

supports_wrapper :: (1..1) Communication_wrapper :: **may_wrap_interactions_of_type** :: (1..1)

Communication wrappers may wrap interactions of a given type.

is_type_for :: (0..n) Interaction :: **has_type** :: (1..1)

:: (0..n) Interaction :: :: (1..1) Each interaction type may be instantiated by many interactions.

has_allowed_receiver_responsibility_type :: (0..n) Interaction_type :: **receiver_responsibility_type_for** :: (0..n)

Each interaction type is permitted to have receiver responsibilities of the specified type. If no receiver responsibility types are indicated, the interactions of this type must not have receiver responsibilities.

receiver_responsibility_type_for :: (0..n) Interaction_type :: **has_allowed_receiver_responsibility_type** :: (0..n)

Each interaction type is permitted to have receiver responsibilities of the specified type. If no receiver responsibility types are indicated, the interactions of this type must not have receiver responsibilities.

Attributes of: **Interaction_type**

description : DescriptiveText

Describes the purpose and use of interactions having this type.

history : CompoundHx

This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

identifier : IdentifierString

A unique identifier used to reference the interaction type.

name : NameString

A unique name for the interaction type.

stateBasedTriggerAssociation : Enumerated

Indicates whether the trigger event associated with the interaction should be associated with a state transition. Possible values are:

R - Trigger must have a state transition

N - Trigger must NOT have a state transition

A - Trigger event is permitted to be associated with a state transition, but is not required to be.

Class: **Message_row_control**

Is part of: **Message_type**

Associated with: **HMD_domain_constraint**
HMD_notation
HMD_row

Description of: **Message_row_control**

An element of a message type that controls the use of a particular HMD row in messages defined by the parent message type. The message row control has one subtype that relates to HMD attribute rows.

Composition for: **Message_row_control**

included_in (1,1) :: Message_type :: **includes (0,n)**

A reference to the message structure of which each message row control is a part.

Associations for: **Message_row_control**

has_domain :: (0..1) HMD_domain_constraint :: **constrains** :: (0..n)

has_notation :: (0..n) HMD_notation :: **annotates** :: (1..1)

controls :: (1..1) HMD_row :: **controlled_by** :: (0..n)

Each message row control controls the presence of one unsubsumed HMD row in the message structure of which the message row control is a part.

Attributes of: **Message_row_control**

conformance : Enumerated

Describes the requirement of information systems to send, or receive and process, this message element in order to claim conformance to the HL7 messaging standard defined by this message type. Values are: Required (R) and Not Required (N).

constraint : String

Expresses a particular constraint for this row in this type.

default_value : String

A notation that captures the default value for this row in the message type. It provides a value that a sending system may insert when creating a message instance if it has no other value to use.

defaultUpdateMode : Enumerated

history : CompoundHx

This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

inclusion : Enumerated

Shows whether a message element may appear and if it may be null.. A mandatory or required message element may appear within an optional message element. If the outer message element, which is optional, actually appears in a message instance, any mandatory inner element must appear Possible values are: Mandatory (M), and Optional (O)..

repetitions : MultiplicityString

Describes whether the message element may repeat. Shows the minimum and maximum number of repetitions for this row (and its subordinates) in this message structure. It is not consistent to have a minimum number of repetitions of zero unless the Inclusion value is Optional.

updateModeSet : Enumerated

Class: **Message_type**

Supertype of: **Union_message_type**

Is part of: **Hierarchical_message_description**

Composite of: **Message_row_control**

Associated with: **Communication_wrapper**
Control_event
HMD_relationship_row
Interaction
Union_message_type

Description of: **Message_type**

A message type is part of an HMD. It defines the specific information transfer that occurs in an interaction to meet the requirements of use cases. It is a set of constraints applied to the message elements defined in the HMD. These are represented by a set of columns in the HMD. The content for those columns is specified by the message row control instances that are parts of the message type.

The message type is an atomic unit in that the entire information content defined by the message type will be sent in an interaction, or no part of it will be sent. No further decomposition is possible.

Composition for: **Message_type**

is_part_of (1,1) :: Hierarchical_message_description :: **contains (1,n)**
Each message structure is contained in a single HMD.

includes (0,n) :: Message_row_control :: **included_in (1,1)**
A reference to the message structure of which each message row control is a part.

Associations for: **Message_type**

implemented_by :: (0..1) Communication_wrapper :: **acts_as** :: (1..1)
Communication wrappers have their content communicated by a message type.

implemented_by :: (0..1) Control_event :: **acts_as** :: (1..1)
Control events have their content represented by a message type.

types :: (0..n) HMD_relationship_row :: **typed_by_CMET** :: (0..1)
Common message element type linkage. Class links the CMET defined by a particular message type (in a particular HMD) to the HMD row (allways a relationship row) that it types as a CMET.

transferred_by :: (1..n) Interaction :: **transfers** :: (1..1)
Each interaction shall include a link to a single message structure that the interaction will transfer.

combined_in :: (0..1) Union_message_type :: **combines** :: (1..n)

Attributes of: **Message_type**

description : DescriptiveText
Informative text describing the intended use or purpose of a particular message type.

history : CompoundHx
This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

identifier : IdentifierString
Arbitrary, unique identifier assigned by Technical Steering Committee. Committees may assign an interim identifier that starts with the committee's identifier, as "Cnn_<identifier>" so long as this composite identifier is unique.

isCommonType : Boolean
Identifies that this message type structure is common to all of the message types in this HMD.

Class: **Model**

Composite of: **Actor**
Application_role
Application_role_relationship
Class
Communication_wrapper
Data_type
Data_type_category
Design_category
Design_information_model
Hierarchical_message_description
Interaction
Receiver_responsibility
Storyboard
Subject_area
Trigger_event
Use_case
Use_case_category

Associated with: **HL7_committee**

Description of: **Model**

This class in the meta-model contains the elements necessary to uniquely define an aggregate model, to establish its provenance and scope, and to link it to each of the elements that make up that model.

A model is a collection of subject areas, scenarios, classes, attributes, use cases, actors, trigger events, interactions, etc. that depict the information needed to specify HL7 Version3 messages. This model is further divided into four specific models - a use case model, an information model, an interaction model, and a message design model..

Composition for: **Model**

has (0,n) :: Actor :: **in (1,1)**
The relationship between actors and the models of which they are a part.

has (0,n) :: Application_role :: **in (1,1)**
The relationship between application roles and the models of which they are a part.

has (0,n) :: Application_role_relationship :: **in (1,1)**
The relationship between application role relationships and the models of which they are a part.

has (0,n) :: Class :: **in (1,1)**
The relationship between classes and the models of which they are a part.

has (0,n) :: Communication_wrapper :: **in (1,1)**

has (0,n) :: Data_type :: **in (1,1)**
The relationship between data types and the models in which they are first defined.

has (0,n) :: Data_type_category :: **in (1,1)**
The relationship between data type categories and the models of which they are a part.

has (0,n) :: Design_category :: **in (1,1)**
The relationship between interaction model categories and the models of which they are a part.

contains (0,n) :: Design_information_model :: **is_part_of (1,1)**

has (0,n) :: Hierarchical_message_description :: **in (1,1)**
The relationship between hierarchical message descriptions and the models in which they are first defined.

has (0,n) :: Interaction :: **in (1,1)**
The relationship between interactions and the models of which they are a part.

has (0,n) :: Receiver_responsibility :: **in (1,1)**

has (0,n) :: Storyboard :: **in (1,1)**
The relationship between scenarios and the models of which they are a part.

has (0,n) :: Subject_area :: **in (1,1)**
The relationship between subject areas and the models of which they are a part.

has (0,n) :: Trigger_event :: **in (1,1)**
Sets relationship between model elements and the models of which they are a part.

has (0,n) :: Use_case :: **in (1,1)**
The relationship between use cases and the models of which they are a part.

has (0,n) :: Use_case_category :: **in (1,1)**
The relationship between use case model categories and the models of which they are a part.

Associations for: **Model**

prepared_by :: (1..1) HL7_committee :: **prepares** :: (0..n)
Links a model to the committee that prepared it.

Attributes of: **Model**

description : DescriptiveText
A short narrative describing the scope and intent of the model.

developing_org : String
A short form identifier of the organization responsible for the publication and maintenance of the model. . For HL7, this name shall be "HL7."

last_modified_date : Date
The date the model was last modified by the model developing organization.

modelID : NameString
A unique identifier assigned to the model by the developing organization. In HL7, these identifiers will be assigned by the Modeling and Methodology Committee.

name : NameString
A descriptive title for the model. The name in combination with the version number shall be unique within the set of models developed by any particular model developing organization.

version_number : VersionNumber
A number showing the release level of the model. The version number, in combination with the name, shall be unique for all public releases of the model.

Class: **Note**

Supertype of: **Reference_note**

Associated with: **Design_annotation**
DIM_notation
HMD_notation

Description of: **Note**
Provides additional descriptive information about the associated item.

Associations for: **Note**

content_for :: (0..n) Design_annotation :: **has_content** :: (1..1)

is_notation :: (0..n) DIM_notation :: **links_note** :: (1..1)

is_notation :: (0..n) HMD_notation :: **links_note** :: (1..1)

Attributes of: **Note**

number : Integer
Allows for reference numbering of annotations when the models are published.

source : Enumerated
An encoded indication of where in the analytic and design process the note was first documented,

text : DescriptiveText
Provides additional descriptive detail about the associated item.

type : Enumerated
Identifies the type of annotation. Supported types are: Design_notes, Implementation_notes, and External_standard_reference.

Class: **Observation_id_link**

Is part of: **Domain_version**

Associated with: **Coded_term**
Vocabulary_concept

Description of: **Observation_id_link**
The Observation Identifier to Value Set Linking Table is used to link an observation identifier, like a LOINC code (Logical Observation Identifier Names and Codes), with a value set. This table is used when it is desirable to specify the exact value set that should be associated with a coded observation identifier as used in a service event, assessment, or observation instance.

Composition for: **Observation_id_link**

in_version (1,1) :: Domain_version :: **has (0,n)**

Associations for: **Observation_id_link**

links_obs_term :: (1..1) Coded_term :: **linked_to_set** :: (0..n)
Each linked observation identifier is drawn from a particular code system. Initially these will all be drawn from LOINC.

links_domain :: (0..1) Vocabulary_concept :: **equates_to** :: (0..n)

Attributes of: **Observation_id_link**

edit_note : DescriptiveText
A general purpose textual field for recording specific information about this code, or details about the rationale for creating, modifying, or deleting this particular table entry.

status : CodedElement
Status - the status of this entry within this table. The values for Status come from vocabulary domain EditStatus. Some values for status are Proposed, Rejected, Active, Obsolete, and Inactive.

version_out : Integer
The version number of the table at which this entry was deleted. A blank Vout value means that the row continues to exist in the current version of the table.

Class: **Project**

Associated with: **HL7_committee**

Description of: **Project**
The specification of a particular, coherent set of messages and events based around one (or a few) Subject Classes. A project is undertaken to address a current need for standardizing information that flows between a number of parties in healthcare. A project also defines a ballot package that might be advanced independently of other HL7 ballot packages.

Associations for: **Project**

responsibility_of :: (1..1) HL7_committee :: **responsible_for** :: (0..n)
Establishes the relationship between committees and the projects for which that committee is responsible.

Attributes of: **Project**

ANSI_PINS_date : Date
The date on which the project scope was published in the ANSI PINS system.

id : IdentifierString
Arbitrary identifier assigned by Technical Steering Committee.

name : NameString
Brief descriptive name for the project.

scope : DescriptiveText
The approved scope statement for the project. It defines the area of healthcare functionality that needs to be supported by HL7 messaging and is a high level use case that encompasses the entire project.

TSC_approval_date : Date
Date on which the project was approved by the TSC.

Class: **Receiver_responsibility**

Is part of: **Model**

Associated with: **Interaction**
Interaction

Trigger_event

Description of: **Receiver_responsibility**

An interaction may have a list of possible receiver responsibilities. On receipt of an interaction having a receiver responsibility, the receiving application is required to perform the interaction and trigger event (if any) identified in one of the possible receiver responsibilities. If no interaction or trigger event is identified by a receiver responsibility, nothing is available.

Composition for: **Receiver_responsibility**

in (1,1) :: Model :: **has (0,n)**

Associations for: **Receiver_responsibility**

initated_by_receiver :: (0..1) Interaction :: **invokes** :: (0..n)

A reference to an interaction that the receiver of the message must initiate once receipt of the message is acknowledged. This is an optional element in that there may no follow-on responsibility. Transactions can be established through a chain of receiver responsibilities for individual interactions.

is_responsibility_for :: (1..1) Interaction :: **has_responsibility_option** :: (0..1)

An application may have a responsibility to perform specific actions upon receiving an interaction. If receiver responsibilities are listed, the receiving application must perform the actions indicated in one of the identified responsibilities.

invokes :: (0..1) Trigger_event :: **initated_by_receiver** :: (0..n)

A reference to a trigger event that the receiver of the message must fire within their own system once receipt of the message is acknowledged. Firing this trigger event may result in interactions being transmitted to the sender of the original message and/or to other systems. The trigger may also result in interactions being triggered by application roles other than the receiving application role. This is an optional element in that there may no follow-on responsibility.

Attributes of: **Receiver_responsibility**

history : CompoundHx

This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

identifier : IdentifierString

A unique identifier used to reference the receiver responsibility.

reason : DescriptiveText

Identifies the reason this particular receiver responsibility would be invoked (as opposed to other receiver responsibility options).

Class: **Reference_note**

Subtype of: **Note**

Description of: **Reference_note**

A type of annotation used when referencing external standards. (Reference_annotation.type must be External_standard_reference.)

Attributes of: **Reference_note**

standard_name : NameString

Identifies the name and version standard being referenced.

Class: **Relationship**

Supertype of: **Association**
Generalization_relationship

Associated with: **Class**
Class
DIM_relationship_row

Description of: **Relationship**

A relationship between classes. In HL7 information models, three types of relationship are accomodated: generalization-specialization; association; and composite aggregation.

Subject to constraints expressed for the sub-types of relationship, the relationship may exist between classes, or between instances (objects) of the same or different classes. When the association is defined one of the two classes is designated the "source class" and the other the "target" class. These designations are further defined in the sub-types of Relationship.

Associations for: **Relationship**

has_destination :: (1..1) Class :: **is_destination** :: (0..n)
A reference to the class that is the target of the association.

has_source :: (1..1) Class :: **is_source** :: (0..n)
A reference to the class from which the association perspective is captured.

has_dependent :: (0..n) DIM_relationship_row :: **is_based_on** :: (1..1)

Attributes of: **Relationship**

description : DescriptiveText
A short informative description of the Generalization relationship.

history : CompoundHx
This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

Note, the version data must fit within the range of the applicable versions for both classes to which this element is attached.

Class: **State**

Is part of: **Subject_class**

Associated with: **DIM_state_row**
State
State
State_transition
State_transition

Description of: **State**

The identification of a unique combination of attribute value(s) and connections which are of interest about a subject class.

The enumerated States for each Subject class must include an "Initial state" from which some designated state transition moves the class to one or more active states.

Composition for: **State**

in (1,1) :: Subject_class :: **has (0,n)**
This relationship between states and the subject classes of which they are a part.

Associations for: **State**

has_dependent :: (0..n) DIM_state_row :: **is_based_on** :: (1..1)

has_substate :: (0..n) State :: **is_substate_of** :: (0..1)
States may be defined as sub-states of a parent, provided that all of the states for a given class have unique names.

is_substate_of :: (0..1) State :: **has_substate** :: (0..n)
States may be defined as sub-states of a parent, provided that all of the states for a given class have unique names.

ends :: (0..n) State_transition :: **ends_in** :: (1..1)

is_start_of :: (0..n) State_transition :: **starts_from** :: (1..1)

Attributes of: **State**

description : DescriptiveText
A short informative description of the State.

history : CompoundHx
This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

Note, the version data must fit within the range of the applicable versions for the class of which this element is a part.

name : NameString
The name of this particular state which shall be unique within this class.

predicate : String
The condition or set of conditions that when true about the attribute(s) and connections of the parent subject class identifies that the subject class is in the declared state.

Class: **State_transition**

Associated with: **State**
State
Trigger_event
Use_case

Description of: **State_transition**
Captures the semantics of transitions from state-to-state for the Subject classes. Note that a legal transition may return to the same state from which it started.

State transitions also are a critical link between leaf-level use cases from which the transitions stem, and the trigger events identified with the transitions.

Associations for: **State_transition**

ends_in :: (1..1) State :: **ends** :: (0..n)

starts_from :: (1..1) State :: **is_start_of** :: (0..n)

identified_by :: (0..1) Trigger_event :: **identifies** :: (0..n)

This connection from state transition to trigger event is only optional because we do not expect all possible trigger events to be defined in HL7. Nevertheless, any state transition that is described in a use case must be linked to a trigger event.

captured_in :: (0..n) Use_case :: **describes** :: (0..1)

A leaf-level use case describes the events that result in a single state transition of the subject class.

Although the instance connection from use case to state transition is optional, this is only because not all use cases are leaf-level. At the leaf-level, each use case should describe one and only one state transition.

Attributes of: **State_transition**

description : DescriptiveText

A short, informative description of the state transition.

history : CompoundHx

This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

Note, the version data must fit within the range of the applicable versions for both of the states to which this transition links.

label : NameString

A word or phrase that reflects the event that causes the transition. The name shall be unique with respect to the state from which the transition starts. This label may be the same as the name of the trigger event identified with the Transition. The name of a transition should reflect the state in which the transition ends. It is acceptable (and common) for multiple transitions to end in the same state, and therefore have the same name.

Class: **Storyboard**

Is part of: **Model**

Composite of: **Interaction_sequence**
Storyboard_example
Use_case_sequence

Associated with: **Design_category**

Description of: **Storyboard**

A storyboard is a statement of health care relevant events defined as a sequence of leaf-level use cases or interactions. The storyboard provides one set of use case instances that the modeling committee expects will typically occur in the domain.

A storyboard may also be expressed as a subset of the interaction model in which case the representation includes all interactions that are implied by the trigger events associated with the sequence of use cases or are implied by the sender and receiver responsibilities of those interactions. Usually, an interaction diagram is constructed to show a group of interactions for a single storyboard.

The collection of storyboards that HL7 may publish does not limit the ways in which HL7 can be applied; other combinations of trigger events, interactions, and application roles that are consistent with the interaction model may also be used.

Composition for: **Storyboard**

contains (0,n) :: Interaction_sequence :: **is_part_of (1,1)**
Each storyboard is made up of a sequence of interactions, a sequence of use cases, or both.

in (1,1) :: Model :: **has (0,n)**
The relationship between scenarios and the models of which they are a part.

exemplifies (0,n) :: Storyboard_example :: **is_part_of (1,1)**
Each storyboard example provides a real world example for a single storyboard..

contains (0,n) :: Use_case_sequence :: **contains (1,1)**
Each storyboard is made up of a sequence of interactions, a sequence of use cases, or both.

Associations for: **Storyboard**

defined_in :: (1..1) Design_category :: **includes** :: (0..n)

Attributes of: **Storyboard**

history : CompoundHx
This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

identifier : IdentifierString
An identifier assigned to the storyboard to simplify references to it. The identifier should be unique within the scope of the model in which it is defined. In HL7, committees manage the unique identifiers for their storyboards using an HL7-defined naming convention.

name : NameString
A short phrase that provides a descriptive title for the storyboard.

purpose : DescriptiveText
The purpose for which the storyboard was created. Frequently it describes the generic set of actions that the storyboard represents.

Class: **Storyboard_example**

Is part of: **Storyboard**

Description of: **Storyboard_example**
Provides a real-world example of the sequence of events captured in a Storyboard.

Composition for: **Storyboard_example**

is_part_of (1,1) :: Storyboard :: **exemplifies (0,n)**
Each storyboard example provides a real world example for a single storyboard..

Attributes of: **Storyboard_example**

description : DescriptiveText

A narrative example from the real world that describes a set of events represented by the sequence of use cases that make up the Storyboard which this example exemplifies.

history : CompoundHx
This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

identifier : IdentifierString
A unique identifier assigned to the storyboard example.

Class: **Structural_attribute**

Subtype of: **Attribute**

Associated with: **Vocabulary_concept**

Description of: **Structural_attribute**
An attribute of a Class that provides a linkage between that Class and the Concepts (codes) that are used to further define or qualify the class's function within the model and/or to extend the class's generalization hierarchy.

The structural attribute must have a coded data type, and the allowable set of code values is limited to those that have been formally adopted by HL7.

Associations for: **Structural_attribute**

structure_defined_by :: (0..n) Vocabulary_concept :: **defines_structure_for** :: (0..1)

Class: **Subject_area**

Is part of: **Model**

Associated with: **Class**
Class
HL7_committee
Subject_area
Subject_area

Description of: **Subject_area**
A major category of information represented in the information model. An aggregation of interrelated classes. A subject area allows portions of a large model to be viewed as a whole thereby eliminating some complexity involved in understanding a large model.

Subject areas will also be used by the Methodology and Modeling Committee of HL7 to designate class stewardship responsibilities and classes of interest to a particular committee.

Composition for: **Subject_area**

in (1,1) :: Model :: **has (0,n)**
The relationship between subject areas and the models of which they are a part.

Associations for: **Subject_area**

holds :: (1..n) Class :: **primarily_resides_in** :: (0..1)

The linkage between a Class and the Subject area that is its primary residence. This must be established if a Class resides in more than one Subject area.

includes :: (1..n) Class :: **appears_in** :: (0..n)
The linkage between a Subject area and each of the Classes that are in that Subject area.

maintained_by :: (0..1) HL7_committee :: **maintains** :: (0..n)

is_nested_in :: (0..1) Subject_area :: **nests** :: (0..n)
The linkage between two subject areas where one of the two is nested within the other.

nests :: (0..n) Subject_area :: **is_nested_in** :: (0..1)
The linkage between two subject areas where one of the two is nested within the other.

Attributes of: **Subject_area**

description : DescriptiveText
Short informative text describing the subject area so as to be clear what type of Classes it includes.

history : CompoundHx
This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

name : NameString
The name given to the subject area. A subject area name is often the plural form of the name of the central or dominant class within the subject area. Subject area names shall be unique within a given model.

Class: **Subject_class**

Subtype of: **Class**

Composite of: **State**

Associated with: **Attribute**
Use_case

Description of: **Subject_class**
A specialization of Class that is used in HL7 to identify those classes that are the focus for a set of Use Cases, Trigger events and/or Application Roles.

Composition for: **Subject_class**

has (0,n) :: State :: **in (1,1)**
This relationship between states and the subject classes of which they are a part.

Associations for: **Subject_class**

has_state_attribute :: (1..1) Attribute :: **is_state_attribute_for** :: (0..1)
The state attribute of a class contains a value indicating the current state of the class. In the event that the class has concurrent states, the attribute must be a set of state values.

subject_of :: (0..n) Use_case :: **describes** :: (0..1)
Links a leaf-level use case to its Subject Class.

Class: **Trigger_event**

Is part of: **Model**

Associated with: **Design_annotation**
Design_category
Interaction
Receiver_responsibility
State_transition

Description of: **Trigger_event**

An occurrence in the health care domain, or within the systems that support this domain, that causes information to be exchanged in the domain or between systems. Trigger events are initiators of Interactions.

Each Trigger event is tied to one of the following: aone State transition for one of the Subject classes in the model, the receiving of a message (interaction) from another system, or some external 'real-world' action that does not involve a state transition (e.g. User submits query).

Composition for: **Trigger_event**

in (1,1) :: Model :: **has (0,n)**

Sets relationship between model elements and the models of which they are a part.

Associations for: **Trigger_event**

has :: (0..n) Design_annotation :: **for** :: (0..1)

defined_in :: (1..1) Design_category :: **includes** :: (0..n)

initiates :: (0..n) Interaction :: **initiated_by** :: (1..1)

A reference to the trigger event that triggers or initiates this interaction.

initated_by_receiver :: (0..n) Receiver_responsibility :: **invokes** :: (0..1)

A reference to a trigger event that the receiver of the message must fire within their own system once receipt of the message is acknowledged. Firing this trigger event may result in interactions being transmitted to the sender of the original message and/or to other systems. The trigger may also result in interactions being triggered by application roles other than the receiving application role. This is an optional element in that there may no follow-on responsibility.

identifies :: (0..n) State_transition :: **identified_by** :: (0..1)

This connection from state transition to trigger event is only optional because we do not expect all possible trigger events to be defined in HL7. Nevertheless, any state transition that is described in a use case must be linked to a trigger event.

Attributes of: **Trigger_event**

dependency : String

If the occurrence of the trigger event is dependent upon the state of one or more objects in the domain or upon the prior occurrence of a different trigger event, this dependency will be expressed in this textual component.

description : DescriptiveText

The text that describes the trigger event. When viewed along with the description of the State transitions which the event identifies, this description must have sufficient detail that the event can be reliably recognized when it occurs.

history : CompoundHx
This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

identifier : IdentifierString
An identifier assigned to the trigger event. The identifier is unique within the scope of the model in which the trigger event is defined. In HL7, committees manage the unique identifiers for their trigger events, and concatenate the committee identifier as "Cnn_<identifier>."

name : NameString
A name assigned to the trigger event. The name is unique within the scope of the model in which the trigger event is defined.

Class: **Union_message_type**

Subtype of: **Message_type**

Associated with: **Message_type**

Associations for: **Union_message_type**

combines :: (1..n) Message_type :: **combined_in** :: (0..1)

Class: **Use_case**

Is part of: **Model**

Associated with: **Actor**
State_transition
Subject_class
Use_case_category
Use_case_relationship
Use_case_relationship
Use_case_sequence

Description of: **Use_case**
A use case is a summary of health care relevant events and related information system events that reflect the usage of the information in the information model and related business models. Use cases describe the interactions and information interchanges that occur in the healthcare domain and the events that cause these interchanges.

Use cases may be expressed at various levels. A use case may be a parent to several child use cases. In this circumstance, the interactions ascribed to all of the children constitute the complete interaction of the parent. The decomposition to child use cases should stop when the resulting use case involves a single actor and a single interaction in response to a single stimulus - an atomic or leaf level use case. Note that a use case may be a child of more than one parent, but must not be defined such that a trace up the parental tree from a child will run into the same child (recursion).

At the lowest level of decomposition, each leaf level use case should link to only a single state transition which, in turn, links to a trigger event that initiates interactions. If the linkage is to multiple such transitions or events, further decomposition should be considered.

Composition for: **Use_case**

in (1,1) :: Model :: **has (0,n)**

The relationship between use cases and the models of which they are a part.

Associations for: **Use_case**

involves :: (1..n) Actor :: **participates_in** :: (0..n)

describes :: (0..1) State_transition :: **captured_in** :: (0..n)
A leaf-level use case describes the events that result in a single state transition of the subject class.

Although the instance connection from use case to state transition is optional, this is only because not all use cases are leaf-level. At the leaf-level, each use case should describe one and only one state transition.

describes :: (0..1) Subject_class :: **subject_of** :: (0..n)
Links a leaf-level use case to its Subject Class.

included_in :: (0..n) Use_case_category :: **includes** :: (0..n)

is_source_for :: (0..n) Use_case_relationship :: **links_source** :: (1..1)
Links a use case relationship to the source of the relationship.

is_target_in :: (0..n) Use_case_relationship :: **links_target** :: (1..1)
Links the use case relationship to its target or destination.

is_linked :: (0..n) Use_case_sequence :: **links** :: (1..1)

Attributes of: **Use_case**

description : DescriptiveText
The text that describes the use case and provides the details necessary to understand the events that are involved in the use case.

history : CompoundHx
This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

identifier : IdentifierString
An identifier assigned to the use case. The identifier is unique within the scope of the model. In HL7, committees manage the unique identifiers for their use cases, and concatenate the committee identifier as "Cnn_<identifier>."

name : NameString
A short phrase that provides a descriptive name for the use case. The name should be unique within the scope of use cases defined by a particular committee.

Class: **Use_case_category**

Is part of: **Model**

Associated with: **Actor**
HL7_committee
Use_case
Use_case_category
Use_case_category

Description of: **Use_case_category**

A major category of information represented in the use case model. An aggregation of interrelated actors and use cases. A category allows portions of a large model to be viewed as a whole thereby eliminating some complexity involved in understanding a large model.

Composition for: **Use_case_category**

in (1,1) :: Model :: **has (0,n)**
The relationship between use case model categories and the models of which they are a part.

Associations for: **Use_case_category**

includes :: (0..n) Actor :: **included_in** :: (0..n)

maintained_by :: (0..1) HL7_committee :: **maintains** :: (0..n)

includes :: (0..n) Use_case :: **included_in** :: (0..n)

nested_in :: (1..1) Use_case_category :: **nests** :: (0..n)
Use case categories may be nested in a hierarchy.

nests :: (0..n) Use_case_category :: **nested_in** :: (1..1)
Use case categories may be nested in a hierarchy.

Attributes of: **Use_case_category**

description : DescriptiveText
Short informative text describing the use case category so as to be clear what type of use cases it includes.

history : CompoundHx
This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

name : NameString
The name given to the use case category. The identifier for the committee defining this category is prepended to the name as Cnn.

Class: **Use_case_relationship**

Associated with: **Use_case**
Use_case

Description of: **Use_case_relationship**
Use cases maintain a variety of relationships. This class captures all such flavors. See the use case model chapter of the MDF for details of the stereotypical relationships.

Associations for: **Use_case_relationship**

links_source :: (1..1) Use_case :: **is_source_for** :: (0..n)
Links a use case relationship to the source of the relationship.

links_target :: (1..1) Use_case :: **is_target_in** :: (0..n)
Links the use case relationship to its target or destination.

Attributes of: **Use_case_relationship**

history : CompoundHx

This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

stereotype : String
Identifies the stereotype for the use case relationship. A blank or null value is a simple generalization relationship, meaning that the target use case Use Case 1 adds additional behavior to the source use case. A value of "extends" means that the target use case adds additional behavior to the source use case at a specified Variation Point. A value of "includes" means that the target uses the source as part of its execution.

Class: **Use_case_sequence**

Is part of: **Storyboard**

Associated with: **Use_case**

Description of: **Use_case_sequence**
Captures the sequence in which a particular use case is enacted in a storyboard.

Composition for: **Use_case_sequence**

contains (1,1) :: Storyboard :: **contains (0,n)**
Each storyboard is made up of a sequence of interactions, a sequence of use cases, or both.

Associations for: **Use_case_sequence**

links :: (1..1) Use_case :: **is_linked** :: (0..n)

Attributes of: **Use_case_sequence**

history : CompoundHx
This is a compound data type that holds the id and previous_ID for history, and the first_ver and last_ver for versioning.

sequence : Integer
The order in which the use case participates in the Storyboard.

Class: **V23_data_type**

Associated with: **Attribute**
V23_fields

Description of: **V23_data_type**
The definition of the data type as specified in Figure 2.2, Chapter 2 of HL7 Version 2.3.

Associations for: **V23_data_type**

typed :: (0..n) Attribute :: **had_V23_type** :: (1..1)
Provides an indication of the data type used in Version 2.x for a particular attribute, if such prior usage has been identified.

types :: (0..n) V23_fields :: **is_of_type** :: (1..1)
Expresses that each field in V2.3 is typed with a data type.

Attributes of: **V23_data_type**

data_type_category : String
The type of data type, such as alphanumeric, or chapter-specific.

data_type_code : String
The two- or three-character code for the data type. e.g. CM

data_type_name : String
The name for the data type as specified in the HL7 Chapter 2 table.

notes_format : String
The format of the data type, particularly for compound data types.

Class: **V23_field_segment**

Associated with: **V23_fields**
V23_segments

Description of: **V23_field_segment**
Links the fields in HL7 V2.3 to the segments in which those fields appear. Source is the HL7 data dictionary.

Associations for: **V23_field_segment**

positions :: (1..1) V23_fields :: **populate** :: (0..n)
Links each field to the segment(s) in which it is used and the sequential position in which it appears.

is_in :: (1..1) V23_segments :: **contains** :: (1..n)
Aggregates the fields that make up each segment.

Attributes of: **V23_field_segment**

sequence : Integer
The sequence number at which the element or field appears in the segment.

Class: **V23_fields**

Associated with: **Attribute**
V23_data_type
V23_field_segment

Description of: **V23_fields**
Contains all of the fields (data elements) specified in HL7 Version 2.x, as captured in the data dictionary published by HL7.

Associations for: **V23_fields**

is_source_for :: (0..n) Attribute :: **based_on** :: (0..n)
Provides a linkage for an information model attribute to its equivalent version 2.x field, if such linkage exists and has been identified.

is_of_type :: (1..1) V23_data_type :: **types** :: (0..n)
Expresses that each field in V2.3 is typed with a data type.

populate :: (0..n) V23_field_segment :: **positions** :: (1..1)
Links each field to the segment(s) in which it is used and the sequential position in which it appears.

Attributes of: **V23_fields**

description : DescriptiveText
The description of this field.

element : Integer
The unique numeric identifier assigned by HL7 for this field.

field_name : String
The name of the field in the HL7 Data Dictionary.

table : Integer
The number of the HL7 table that provides values for this field, if the field is table-based.

Class: **V23_segments**

Associated with: **Attribute**
V23_field_segment

Description of: **V23_segments**
Contains all of the segments specified in HL7 Version 2.3, as captured in the data dictionary published by HL7.

Associations for: **V23_segments**

source_of :: (0..n) Attribute :: **stems_from** :: (0..n)
Many attributes are traced to equivalent content in HL7 Version 2.x. This connection is secondary to the path that traces an attribute to an HL7 field to a segment. It is provided for modelers who wish to specify particular segments for information model attributes.

contains :: (1..n) V23_field_segment :: **is_in** :: (1..1)
Aggregates the fields that make up each segment.

Attributes of: **V23_segments**

name : String
The name of the segment.

segment : String
The three-character segment identifier.

Class: **Vocabulary_concept**

Is part of: **Domain_version**

Associated with: **Attribute_domain_constraint**
Code_system
Coded_term
Concept_relationship
Concept_relationship
DIM_attribute_domain_constraint
HMD_domain_constraint
Observation_id_link
Structural_attribute

Description of: **Vocabulary_concept**

Aliases for this table include "Vocabulary domain" and "Vocabulary value set." The distinction among the three notions is carried in the type_cd attribute.

A concept is defined by ISO 1087 as a "unit of thought constituted through abstraction on the basis of characteristics common to a set of objects." A value domain consists of a set of concepts, not a set of words or codes. Any given concept may be represented using different coding systems.

This table describes the high level definition of a given vocabulary domain. It holds the name of the vocabulary domain, and anchors the relationship of the vocabulary domain to different realms of use, and to the various vocabularies and coding systems that are used to define the vocabulary domain. The structure of this table allows a vocabulary domain (or its value sets) to be defined by recursive reference to other vocabulary domains or value sets.

Composition for: **Vocabulary_concept**

in_version (1,1) :: Domain_version :: **has (0,n)**

Associations for: **Vocabulary_concept**

is_constraint :: (0..n) Attribute_domain_constraint :: **links_domain** :: (1..1)

has_basis_in :: (0..1) Code_system :: **is_basis_for** :: (0..n)
Each value set is based on a single code system.

is_represented_by :: (0..n) Coded_term :: **represents** :: (0..1)
Links a coded term to a single concept.

is_contained_concept :: (0..n) Concept_relationship :: **links_content** :: (1..1)

is_containing_concept :: (0..n) Concept_relationship :: **has_container** :: (1..1)

is_constraint :: (0..n) DIM_attribute_domain_constraint :: **links_domain** :: (1..1)

is_constraint :: (0..n) HMD_domain_constraint :: **links_domain** :: (1..1)

equates_to :: (0..n) Observation_id_link :: **links_domain** :: (0..1)

defines_structure_for :: (0..1) Structural_attribute :: **structure_defined_by** :: (0..n)

Attributes of: **Vocabulary_concept**

applies_to : String
Qualifies the application of the codes for RIM structural attributes, as to which moods, relationship types, etc. the code in question applies.

concept_id : Integer
A unique, sequentially assigned number that identifies a vocabulary concept. It stems from the following requirements in the MDF:

1) HL7 Concept Identifier - the unique item identifier assigned by HL7 to this concept. This concept identifier is globally unique to a concept throughout all HL7 tables, and it does not overlap any HL7 value set identifier. That is, if the concept male occurred in another vocabulary domain in addition to the Gender domain, it would again have an item identifier of "40001". If a universal terminology of medicine becomes available, the universal concept identifier from that terminology will be used in place of this HL7 assigned identifier.

2) HL7 Value Set Identifier - a unique, sequential number assigned by HL7 that identifies a value set. Each unique combination of a Realm, a Domain Name, and a Code System is given a unique value set identifier. For HL7 tables that exist in version 2.X, the value set identifier is the same as the version 2.X table number. An HL7 value set will never have the same identifier as an HL7 concept.

define_sequence : Integer

The sequence number used when the concept was defined for an HL7 table. Attribute has no semantic import, but does allow reconstruction of the submission form at a later date.

defining_expression : String

Value Set Definition Expression - an expression that defines how the value set is derived from other pre-existing value sets. The expression refers to value sets using a name enclosed in double quotes. The name enclosed in quotes is a concatenation of the domain name, the realm, and the code system. When a value set expression is for the HL7 code system, the expression includes set operators that indicate how a given value set can be derived from pre-existing HL7 maintained value sets. The value sets referenced in the expression can be either primitive or composite. A composite value set is a value set that contains other value sets. The allowed set operators are:

Symbol Description + Union - Difference * Intersection

Parentheses are allowed in the expression when they are needed to create the proper ordering of the operations. If the value set definition is for any system other than HL7, then there must be a valid expression in the expression field that refers to a value set provided by the terminology source named in the code system column of this table. For non-HL7 vocabularies, operators other than the usual set operators are allowed. For example, ChildrenOf might be used as a keyword to indicate that all children of a given hierarchical node are included in the value set. It is the responsibility of the given terminology provider to define the operators, keywords, and syntax that are supported by their terminology system, and to state how hierarchical structures (nodes) should be referenced.

description : DescriptiveText

Value Set Description - a textual description of the value set as it is known within the code system. When possible, the principle upon which concepts are either included or excluded from the domain should be stated.

Concept Definition - Description - a textual representation of the meaning of this entry.

edit_note : DescriptiveText

Editor's notes for the domain. A general purpose textual field for recording specific information about the entry, or details about the rationale for modifying this particular table entry.

how_applies : String

In conjunction with 'applies_to' tells how the concept in question applies to a particular structural element.

name : String

All names in this class will be unique, regardless of whether they are for domains, concepts or value sets.

For Domains, a unique textual name for the vocabulary domain. The name is created using mixed case object oriented style names, without the use of white space or special punctuation. The name is generally singular. This name is used when the vocabulary domain is referenced by other vocabulary domain definitions. Examples of acceptable names are: Gender, OrderType, PatientType, AbnormalFlag, etc.

The domain name is retained across all realms. Thus, the vocabulary domain name implies a different set of values depending on the Realm of use and the code system. The name may be determined by the organization defining the system which includes this domain.

For Value sets, A unique textual name assigned by HL7 for the value set. The value set name implies a different set of concepts within each realm of use and code system. The name is generally singular. This name is used when the value set is used in HL7 specifications or when the value set is referenced by other vocabulary domain definitions.

For Concepts, A unique textual name assigned by HL7 for the concept. This name need not be the same as the "print name" of any of the systems.

open_issue : DescriptiveText

Captures issues raised and entered into the record as part of vocabulary harmonization.

preference : Integer

Each coded field has only one preferred value set. That is, there is a single preferred coding scheme for a given domain-realm combination. Other value set definition are allowed, but they are of lower rank. The rank is expressed as an integer, with the preferred rank being one ("1").

realm_of_use : Enumerated

Realm - the realm of use of a value set.(This attribute does not apply directly to either a domain or a concept. A given vocabulary domain will have a new row for each different realm of use and code system. The jurisdiction or realm within which the domain will be used. A realm might be a country, a group of countries, a region of the world, or an organization. The values for the realm column come from the RealmOfUse vocabulary domain.

status : CodedElement

Status - the status of the item. The values for Status come from the vocabulary domain EditStatus. Some values for status are Proposed, Rejected, Active, Obsolete, and Inactive.

type_cd : Enumerated

A code indicating the type of entity being represented as a concept. This occurs because the class represents a generalization hierarchy that has been collapsed into a single class. Values for this code and their meanings include:

D - Domain -- Within the HL7 message framework, a vocabulary domain is the set of all concepts that can be taken as valid values in an instance of a coded field or attribute. A domain is the complete set of all concepts that are valid values for an attribute in the RIM, an HMD, a CMET, or a template,

C - Simple Concept - Represents a terminal or leaf-level concept. It is represented as a single term in a code system.

- Value Set - A domain that has been constrained to a particular realm and vocabulary is called a value set. Value sets (domains that have been specialized by realm and code system) are defined from concepts from a single vocabulary. A vocabulary domain that has been specialized in the context of a specific message and placed in the context of a specific Realm and Code System becomes a value set. A value set is the subset of concepts from the global domain that are applicable in a given Realm and Code System.

H - HL7 Value Set - A value set that is based on an HL7-defined code system. E - External Value set

version_out : Integer

Vout - the version number of the domain specification database at the time this entry was updated or deleted. A blank Vout value means that the entry continues to exist in the current version of the table.

Infrastructure classes in: **HL7_V3_Meta-Model**

Data type definitions in: **HL7_V3_Meta-Model**

Data type: **Boolean : Boolean**

Description of: **Boolean**

Boolean data

Data type: **CodedElement : CodedElement**

Description of: **CodedElement**

Coded data

Data type: **CompoundHx : CompoundHx**

Description of: **CompoundHx**

This set of components is assigned to one attribute of most meta-model elements. These components serve to track the history of each element and designate the models in which the element is valid.

Components of: **CompoundHx**

firstVer : String

This component contains the model unique identifier (modelID) of the first model version in which this element was defined.

hxID : Integer

This component is used to track the version history of each element of the model. It contains the unique element identifier assigned to each model element. The values are assigned in the repository. Modelers should never change these values or assign new ones, but they may copy them to indicate element history.

lastVer : String

This component contains the model unique identifier (modelID) of the model for which this element ceased to be valid. A blank lastVer value means that the element is valid in the most recent HL7 models. If this value is valued, the element is no longer a member of the current RIM. Since model identifiers are monotonically increasing, a given element is valid from the model identified by firstVer up to but not including the model identified by lastVer.

prevHxID : Integer

If an element of the meta-model derives from a previously defined element, this component will be valued. It contains the unique identifier of the element's predecessor,

Data type: **Date : Date**

Description of: **Date**

Date data.

Data type: **DateTime : DateTime**

Description of: **DateTime**

DateTime data.

Data type: **DescriptiveText : DescriptiveText**

Description of: **DescriptiveText**

In most instances the information to be kept about model components includes provision for a textual description of the component. Experience in documenting such models has shown the value of structuring these descriptions in order to provide for reference to external documents, identification of open issues, explanation of modeling rationale, etc. Therefore, descriptions of model components shall be of type DescriptiveText, as follows.

A paragraph that is part of the regular description shall not begin with one of the reserved phrases. Paragraphs that begin with a reserved phrase are used to capture comments about the rationale for modeling, to capture open issues and to express external references. The reserved phrases and their usage are shown below:

Reserved phrase "Rationale:" allows the modeler to document the rationale or justification for the specification of a particular element. It may occupy one or more paragraphs, but only one modeling rationale component should appear for any given model element. The first paragraph of the rationale must begin "Rationale:" The rationale will continue to the end of the description or until another reserved phrase is encountered.

Reserved phrase "OpenIssue:" allows the modeler to identify and discuss any open issues that remain to be resolved with respect to the model element. It may occupy one or more paragraphs, and there may be multiple open issues for a model element. The first paragraph of each open issue must begin with "OpenIssue:" The open issue will continue to the end of the description or until another reserved phrase is encountered.

Reserved phrase "ExtRef:" provides the specification of a reference to an external document, either by name or by a URL reference. Multiple external references may be contained in a given description. The external reference must be a single paragraph that starts with "ExtRef:" and must either be the final paragraph of the description or it must be followed by another reserved phrase paragraph.

Data type: **Enumerated : Enumerated**

Description of: **Enumerated**

Enumerated data

Data type: **IdentifierString : IdentifierString**

Description of: **IdentifierString**

Various data model elements in the use case model and interaction model are required to have identifiers that are unique throughout the model. These elements will be specified as being represented by an IdentifierString.

Elements that use these identifiers are: application role, interaction, message, scenario, scenario example, trigger event, and use case.

An IdentifierString is a string that contains no embedded spaces and that is built from a limited character set. The IdentifierString may include any number of the following characters: upper or lower case alphabetic characters (A-Z and a-z); the digits (0-9); the dot character (.); the hyphen character (-); and the underscore character (_). These characters may be in any order, except that the first character of the IdentifierString shall be either a digit or an upper case alphabetic character.

Note: Because the dot character (.) is an allowed member of an IdentifierString, the IdentifierString cannot be used in defining fully qualified names for elements.

Data type: **Integer : Integer**

Description of: **Integer**

Integer data.

Data type: **MultiplicityString : MultiplicityString**

Description of: **MultiplicityString**

A set of values and value ranges including the minimum and maximum occurrence are required for associations and aggregations in the meta-model. A MultiplicityString shall be used to represent this set in both the literary and graphical expressions of the model. The MultiplicityString is a constrained string. It is built according to the following rules:

1. A MultiplicityString shall have at least a minimum and a maximum value.

2. A MultiplicityString may also include an open ended range at the upper end.

3. A MultiplicityString shall be expressed either as the single element "1" (the numeral one) or as a pair of elements separated by an ellipsis (..).

4. The elements making up a MultiplicityString shall be either zero, a positive integer, or the character "*."

5. If the character "*" appears in a MultiplicityString, there must be only a single occurrence, and that occurrence shall represent the set of all positive integers that are greater than the largest of the other integers in the same MultiplicityString.

6. The minimum value for the multiplicity shall be the smallest integer in the MultiplicityString, and may not be the character "*."

7. The maximum value for the multiplicity shall be the largest integer in the MultiplicityString, and must be greater than zero.

8. The elements making up a MultiplicityString should be ordered in ascending order, but are not required to be.

Data type: **NameString : NameString**

Description of: **NameString**

The NameString is a string that contains no embedded spaces and that is built from a limited character set. The NameString may include any number of the following characters: upper or lower case alphabetic characters (A-Z and a-z); the digits (0-9); the hyphen (-), and the underscore character (_). These characters may be in any order, except that the first character of the NameString shall be an alphabetic character.

The appropriate use of upper and lower case characters, and the inclusion of special characters allow data modelers to create easily readable strings for the noun- and verb-phrases required for many of these elements. (Examples might include "is_ordered_by" or "HealthCarePractitioner" or NameString.)

For clarity of reading, the initial character of a NameString should be lower case when used for names of attributes, labels of relationships, and labels for state transitions, and should be upper case in all other uses.. No matter what conventions are used with respect to capitalization, when NameStrings are compared for uniqueness, all alphabetic characters shall be treated as though they are lower case.

Data type: **String : String**

Description of: **String**

String data.

Data type: **VersionNumber : VersionNumber**

Description of: **VersionNumber**

A version number is a string comprised solely of the digit characters and the dot (.) character.

Stewardship & DIMs in: **HL7_V3_Meta-Model**

Data type categories for: **HL7_V3_Meta-Model**

Data type category: **MET_Metamodel_data_types**

Specifies particular data types used in the meta-model. Other data types such as String, Boolean, Integer and Enumerated are not listed here.

Contains data types:
Boolean
CodedElement
CompoundHx
Date
DateTime
DescriptiveText
Enumerated
IdentifierString
Integer
MultiplicityString
NameString
String
VersionNumber

Stewardship & DIMs in: **HL7_V3_Meta-Model**

