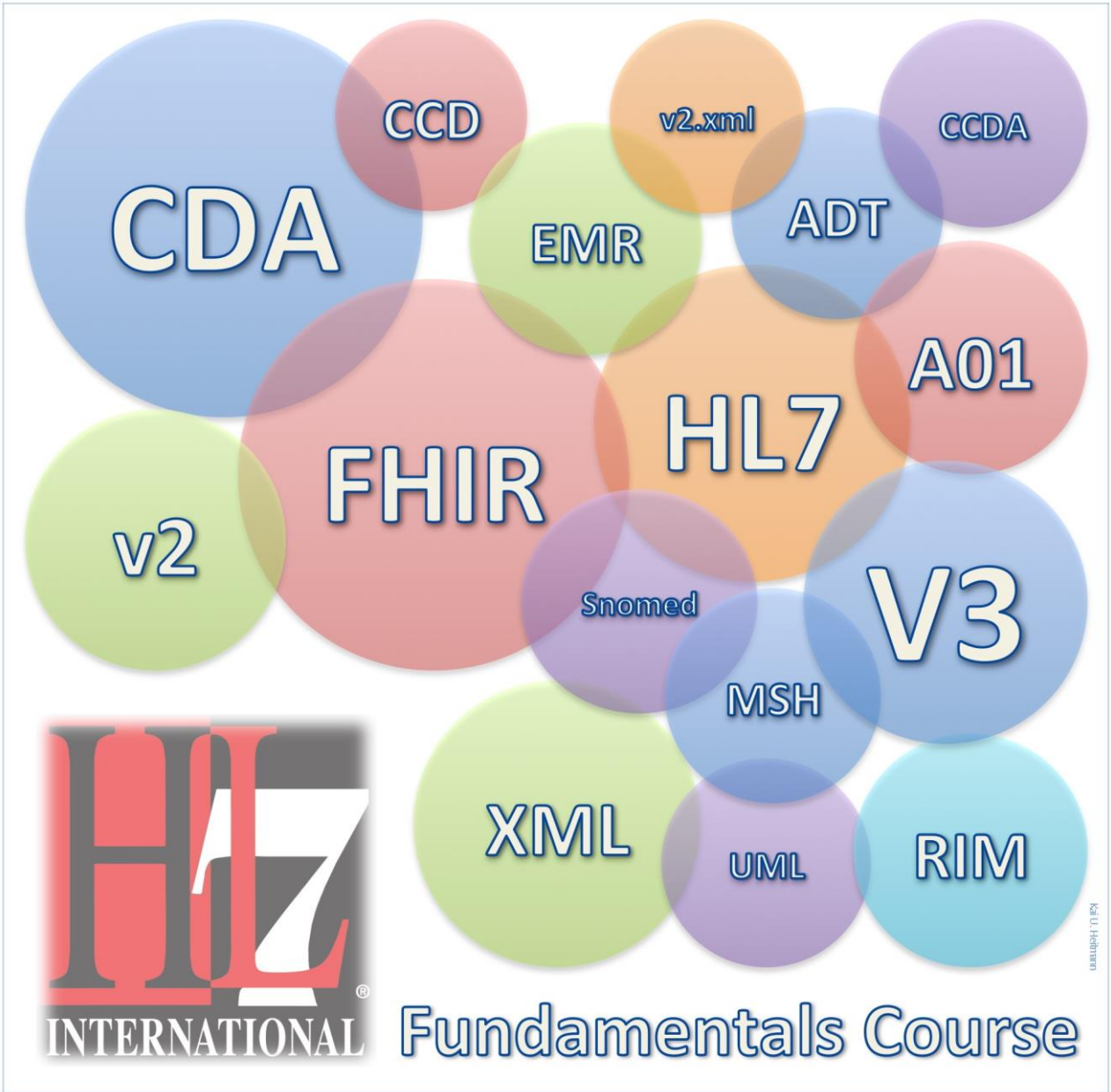


V3 Module

Introduction to the HL7 V3, RIM and datatypes

Reading Material

Table of Contents

1. Introduction to HL7 Version 3

HL7 Version 3 ("V3) represents an approach to clinical information exchange using a model driven methodology that produces messages and electronic documents expressed in XML syntax. Like the other HL7 standards, V3 allows implementers to work with the full set of messages, data types and terminologies needed to build healthcare computing systems implementations.

Some of the HL7 V3 products include:

- **Information Model:** Focuses on semantic interoperability by specifying that information be presented in a complete clinical context that assures that the sending and receiving systems share the meaning (semantics) of the information being exchanged. Examples include the HL7 Version 3: Reference Information Model (RIM).
- **Data Types:** This product provides a set of global representations for data used in the presentation and communication of healthcare information. Examples includes the HL7 Version 3 Standard: Implementation Technology Specification R2 - ISO Harmonized Datatypes.
- **Domain Models:** Use cases and models of common clinical scenarios related to clinical encounters and conditions. Examples include the HL7 Version 3 Domain Analysis Model: Emergency Medical Services.
- **Structured Documents**: Markup standard for the exchange of structured clinical documents. Examples include the HL7 Clinical Document Architecture (CDA) for sharing the vast array of clinical documentation.
- **Services:** Service-Oriented capabilities for health information resources management and standardizes the way in which the resources are exposed and consumed, independently from the nature of the resources. Examples include the HL7 Version 3 Standard: Retrieve, Locate, and Update Service (RLUS), Common Terminology Service (CTS2) and Entity Identification Service (EIS), etc...

As the main use of HL7 V3 specs globally is as the base for structured documents, we will focus on the RIM (Reference Information Model) and its datatypes. If you are interest in HL7 V3 messaging, you can find specific material for that in this unit's additional materials.

2. The HL7 V3 RIM

RIM Classes

Figure 1 shows the foundation classes of the RIM - so you will be able to recognize them when we look at the full RIM:

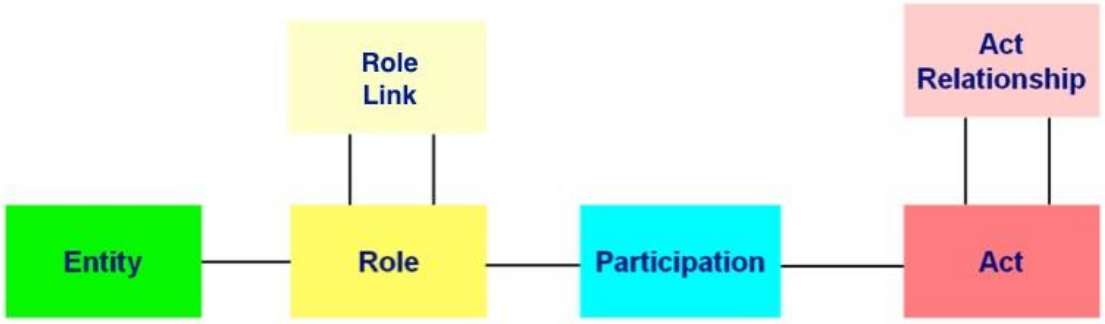


Figure 1: The six Foundation Classes of the HL7 Reference Information Model (RIM)

The RIM is very abstract. All the objects in the health-care domain can be expressed by combining these foundation classes:

- **Act:** Any action in the health-care domain ("medical acts" are only one type of act)
- **Participation:** The act participation context: Who participated and where did it take place.
- **Entity:** living and non-living entities or things, subjects or targets of the act
- **Role:** The role that each entity plays in its participation
- **ActRelationship:** Relationships between acts
- **RoleLink:** Links between roles

Reducing this model down to only six main classes (Act, Participation, Entity, Role, ActRelationship and RoleLink) was a complex process spanning several years. Before reaching this level of abstraction, early rudimentary models of the RIM had hundreds of classes.[1]

In addition to the foundation classes, the RIM includes two infrastructure classes related to messaging and documents.

The complete RIM

In Figure 1 we introduced the six foundation and infrastructure classes of the RIM. Each class group is visually identified by its color. Before getting into details about the various classes, take a

[1] For previous versions of the RIM, see www.hl7.org/implement/standards/product_brief.cfm?product_id=30

look at Figure 2 below which shows the complete RIM. Please take this opportunity to become familiar with the color codes used in the 'RIM:

Entity = green
Role and RoleLink = yellow
Participation = light blue
Act = red
ActRelationship = purple
Infrastructure = blue

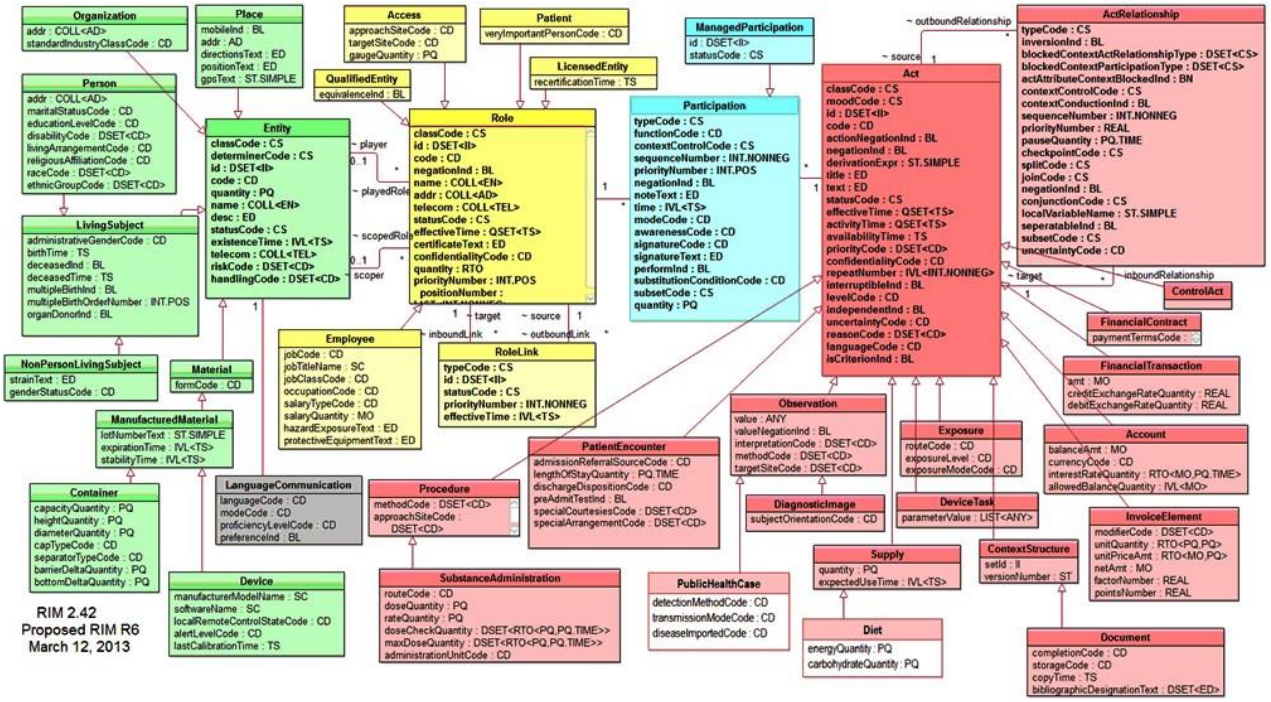


Figure 2: The complete HL7 Reference Information Model (RIM)[2]

How to Read a Class Diagram

The RIM is composed of **Class Diagram** models as defined by the OMG's UML specification.

Those familiar with this notation will note that these diagrams lack "methods" (functions that classes can perform). This is because the RIM uses a STATIC DIAGRAM; it only explains the class attributes and the relationship between classes.

Figure 3 shows a fragment of the RIM. We will use it to gain an understanding of how to read the RIM:

[2] For an on-line image of the RIM, go to www.hl7.org/v3ballot/html/infrastructure/rim/Graphics/RIM_billboard.pdf

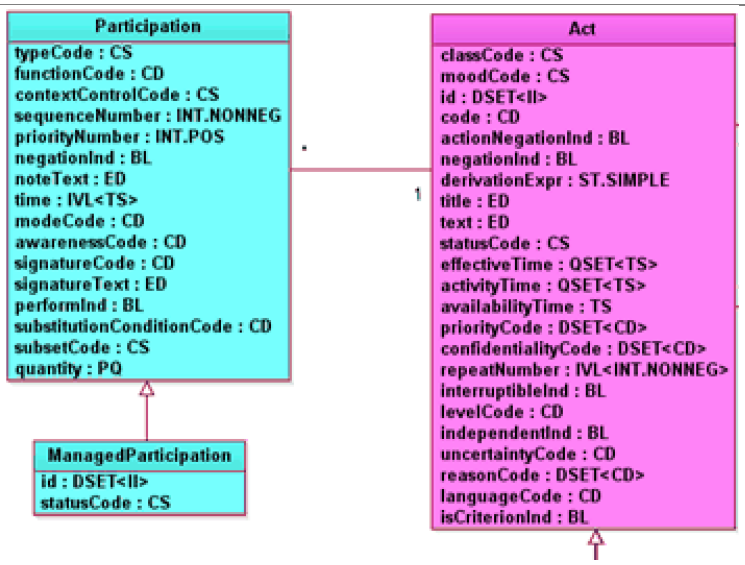
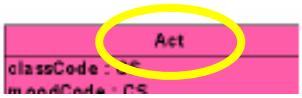


Figure 3: A RIM Fragment[3]

At this point, you should recognize each colored rectangle as a "class".

Class Name

The text above the line in each rectangle is the class name; as shown for the **Act** class:



Attributes

The **attributes** or characteristics of a class appear below the line.

The attributes of the **Act** are **classCode**, **moodCode**, **id**, **code**, **negationInd**, etc.

It is important to distinguish "attributes" of a class from "attributes" in XML. Class attributes are generally represented as elements in XML. See the section about V3 Implementation Technology Specification (ITS) below for more about this.

[3] For an on-line image of the RIM, go to www.hl7.org/v3ballot/html/infrastructure/rim/Graphics/RIM_billboard.pdf

Data Types

Each attribute has an associated data type immediately following the attribute name.

Some examples from Figure 3 above:

- Attribute classCode of class Act is of type CS
- Attribute **title** of class **Act** is of type **ED**
- Attribute quantity of class Participation is of type PQ

The different data types will be explained in the next Unit.

Specialization

The arrow pointing up in Figure 4 below indicates that both the **Procedure** and **Observation class** are "specializations of" **Act.**

As a specialized form of **ACT**, **OBSERVATION** inherits all of the attributes of **ACT** plus a few that are specific to **OBSERVATION**. Similarly, we would treat **Procedure** as an **Act** with additional attributes: **methodCode, approachSiteCode,** and **targetSiteCode.**

In a full UML class diagram this specialization concept also applies to the "methods", but because the RIM is a static model, that does not apply.

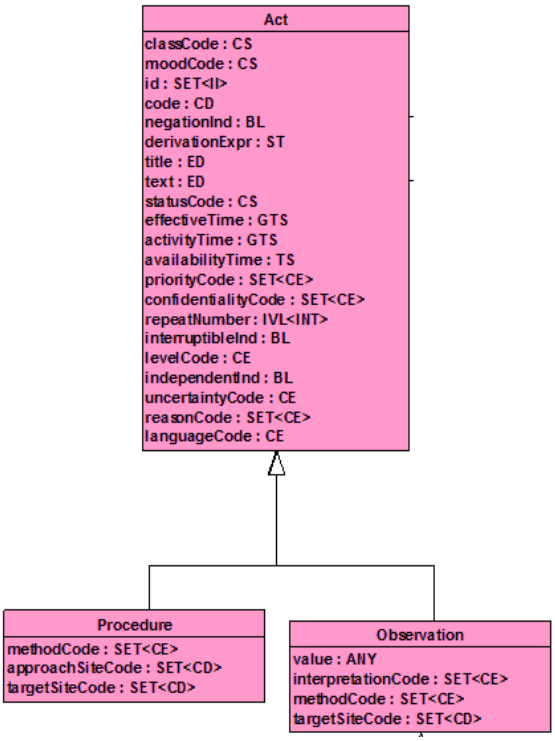


Figure 4: A RIM Specialization

Relationships between Classes

The lines connecting the classes **Entity** and **Role** in Figure 5 below denote a **relationship between classes.**

The cardinality (the numbers under the line) is declared in the **relationship** between classes **Role** and **Entity -** it tells us that each **Entity** could have many (*) **Roles,** and that each **Role** could have zero or one (0..1) **Entities**.

The name above the line is the role played by the relationship.

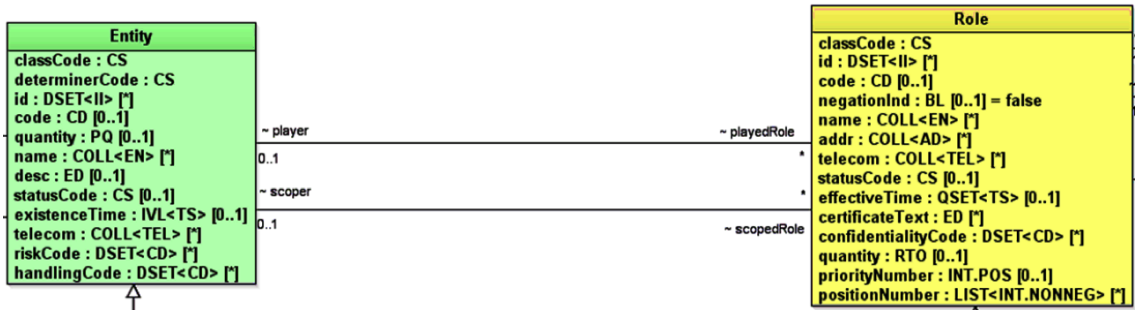


Figure 5: Relationships between RIM Classes

Relationship cardinality may be:

*	Zero to many
1	One and only One
0..1	Zero to one
1..n	One to many

3. The Six Foundation Classes of the RIM

Now take a look at each of the foundation classes and some of their attributes.

Act

An HL7 V3 "act" is defined as "A record of something that is being done, has been done, can be done, or is intended or requested to be done".

""Examples (or we might say "specializations") of Act include: patient encounter, procedure, diet, diagnostic image, observation, substance administration, financial transaction, account, and, document.

You may ask: "*Wait a minute, how can a diagnostic image be an Act*?" Keep in mind that the RIM is **act-centric** and what matters is not the "physical image" (a stream of DICOM or JPEG bytes), but the fact that we must, will or have obtained the diagnostic image. Of course, we will have to transfer/store the physical image, but the image itself is an attribute of the Act.

The attributes of **Act** are shown in Figure 6:

Figure 6 : Attributes of an Act

classCode: defines the kind of Act with each code representing a specialization.

moodCode: reflects the "intent" of the Act versus its status. The section on "difference between state and mood" should help clarify this statement.

classCode *and* ***moodCode*** *are known as STRUCTURAL ATTRIBUTES and are the only ones represented in XML as attributes and not as elements.*

id: a unique identifier for the Act (account number, order number, transaction number, etc.). This is not a description or classification of the Act, but a globally unique identifier for the Act issued by the responsible application.

code: defines the Act in controlled vocabulary terms such as ICD-9 for diagnostics or LOINC for lab results.

negationInd: expresses the negative of the Act in question. For example: if the Act is a diagnosis of diabetes, then if negationInd is "true" then this would express "No diagnosis of Diabetes".

statusCode: defines the "state" of the Act according to the state diagram associated with the Act. The following section "Difference between state and mood" provides an explanation.

effectiveTime: date/time the Act state becomes effective.

Difference between State and Mood

The state diagram is a tool for documenting the possible states of an Act.

For each act specialization, V3 methodology requires a state diagram to define its possible states. These states are expressed through the Act's **statusCode** attribute.

For any Act, the possible states will be a subset of the diagram shown in Figure 7:

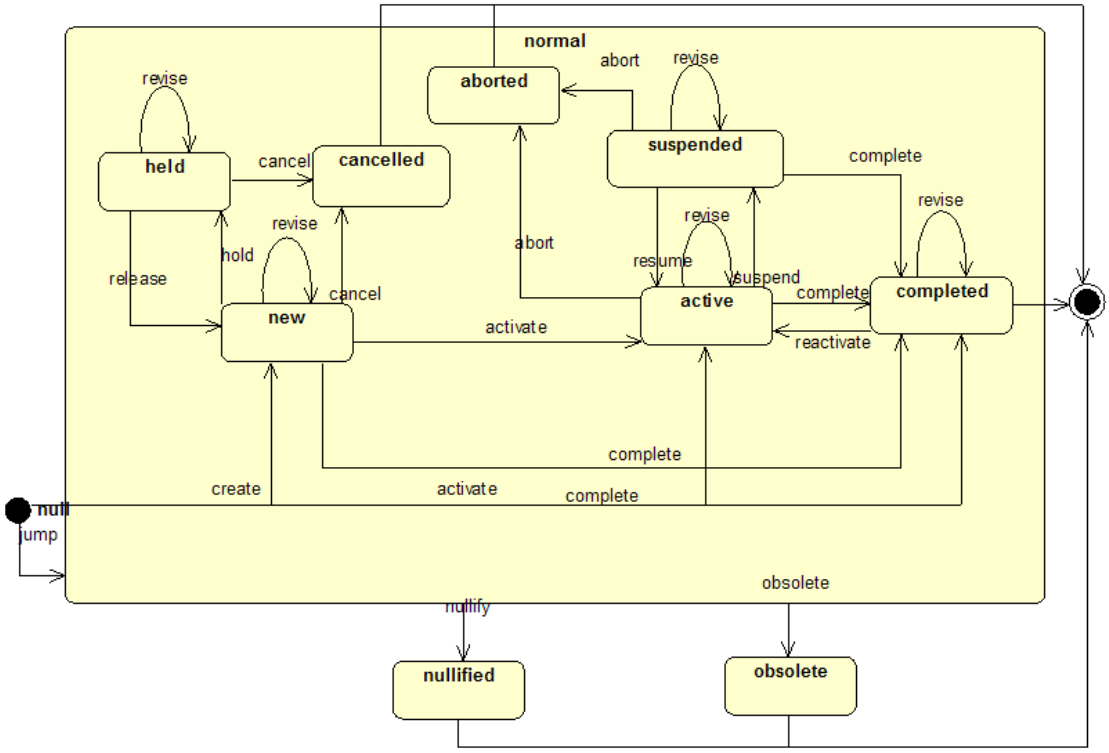


Figure 7: ACT State Diagram

In order to position an Act in relation to the lifecycle of an activity (from defining, through ordering, scheduling and finally performing), V3 has defined the concept of "**Mood**".

***A moodCode** is a code that distinguishes whether an act has already happened, is a possibility, a plan, a goal, an order, a result, etc.*

An example, using **Blood Glucose,** will help to explain this novel concept:

moodCode = "DEF" (Definition)

The concept of a blood serum level, a form of **Observation** (specialized from **Act),** must be defined somewhere in order for it to be ordered and performed (with at least a code, description and some other attributes).

Therefore, we would have the following Act:

```
Observation
classCode="OBS"   (a specialization of act)
moodCode="DEF" (definition)
code="14749-6"    (LOINC code for serum glucose)
```

statusCode="active" (it is active in the master file; it can be ordered)

For this example, the moodCode = "RQO" (Request/Order)

Once it is defined, we can place an order for someone to perform it.

Observation
classCode="OBS"
moodCode="RQO" (request/order)
id=... (The order identifier, issued by the ordering application)
code="14749-6"
statusCode="active" (The order is active)
effectiveTime=... (The date/time we want the order to be performed)
priorityCode=... (The priority for the order)

For this example, the moodCode = "EVN" (Event)

Finally, we report the event of the test result.

Observation
classCode="OBS"
moodCode="EVN" (Event)
id=... (The order identifier)
code="14749-6"
statusCode="completed" - The serum glucose test has been completed. (Please note that we can nullify the result in case of an error, with statusCode="nullified.")
The result of the test will go into the attribute...
value = ... (From the observation specialization in Figure 5.)

What is important in this example is that by using **moodCode** and **statusCode,** we can represent the various stages of an activity (definition, request, event, etc.) and the state of each stage having the same class and almost the same attributes.

Two final points about **statusCode** and **moodCode** to remember:

*A **Mood** is not a **State**!*

*An Act instance can only have a **single** moodCode value*

A Small Digression ...

You may ask: "*I have a LIS, we enter the order and it returns the observation. Aren't they the same object?*" Perhaps they are in your application; however, consider that you may have to identify the ORDER NUMBER, DATE, AUTHOR from the ordering application and your FILLER NUMBER, DATE, AUTHOR for the observations subsequently generated by your Laboratory Information System (LIS) application.

However, V3 definitely does not consider them the same object; they are two separate instances of an Act. However, there is no question that they are related. Information about this relationship is provided by the **ActRelationship** class. The order is related to the observation. We will see more about act relationships later.

Usually, each application will have some responsibilities for the object **in some mood**. For example, a Laboratory Information System (LIS) will use the EVN moodCode to represent the results of lab orders while CPOEs will use the RQO moodCode to represent the same lab order.

4. Participation

Participation is defined as an association between an Act and a Role. The Entity playing the Role is the "actor".

Participation attributes are shown in Figure 8:

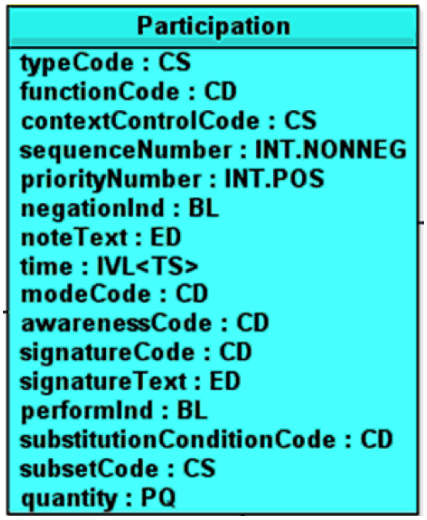


Figure 8 : Participation Attributes

typeCode: the type of participation. For example, consultant, escort, device, subject, specimen, receiver, data enterer, informant, witness, referrer, location, target product, donor, performer.

functionCode: the specialization of the participant's function. For example, admitting physician, anesthetist, nurse assistant, surgeon assistant, etc.

time: the time of the participation.

signatureCode: the status of the signature. For example, signed, intends to sign, requires a signature.

negationInd: If set to true, then the Participation is negated**.**

modeCode: the participation modality. For example, electronic, verbal, written, physically present, etc.

5. Role

Defined as "a competency of the Entity that plays the Role as identified, defined, guaranteed or acknowledged by the Entity that scopes the Role".

Each Entity participates in a given Act in a defined Role. Roles are **played** by some entities and **scoped** by others.

This may seem a little complex, so let us try an example:

Dr. Eric Woo works as a clinician in the Good Health Clinic.

So for the role of "Clinician" we have:"

- The entity playing the role is "Eric Woo" (a person).
- The entity scoping the role is the Good Health Clinic.

But later Dr. Eric Woo stumbles into a bear trap in the woods and was attended by Dr. Karen Carpenter at the Stone Mountain Satellite Clinic.

For the role of "Patient" we have:

- The entity playing the role is "Eric Woo" (a person).
- The entity scoping the role is the "Stone Mountain Satellite Clinic".

Note that Dr. Carpenter has a role as well: attending physician scoped by Stone Mountain Satellite Clinic.

The attributes for **Role** are shown in Figure 9:

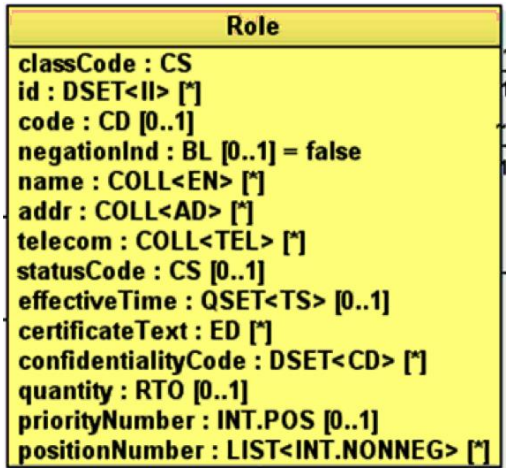


Figure 9 : Role Attributes

classCode: the major class of Role. For example, patient, employer, employee, etc.
code: the specific kind of Role.
id: a unique identifier for the player Entity in this Role; For example, the patient identifier assigned to this person by a given hospital.

name: the name of the Role
addr: the address of the Role
telecom: the phone or e-mail address of the Role
statusCode: the status code of the Role (see Figure 9 above).

6. Entity

Defined as "A physical thing, group of physical things or an organization capable of participating in Acts while in a role".

Some specializations include living subject (person, animal), material, organization, place. The Entity does not include events, acts or roles.

The **Entity** attributes are shown in Figure 10:

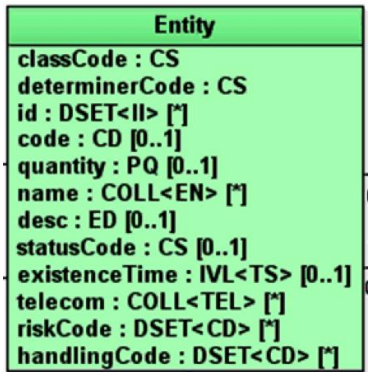


Figure 10 : Entity Attributes

classCode: the Entity class specialization code. For example, person, microorganism, organization, place, city, country, group, state, building, public institution, facility, etc.
determinerCode: specifies whether an Entity is representing only one thing or several.
id: the identifier of the Entity (for example, the person identifier issued by a country).
code: a coded representation of the Entity. For example, the HL7 vocabulary codes for sample types.
name: the Entity name.
desc: the Entity description.
statusCode: the status code using a specialized state diagram (see Figure 16).

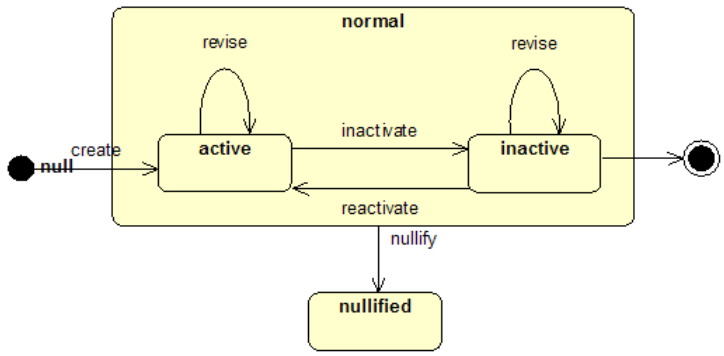


Figure 1: Entity State Diagram

Another Small Digression …

Question: If there is an entity state called "nullify" - what does that mean? If you nullify a person, have they ceased to exist?

No, it is not about **the real "you";** it is about **the application's record of "you".** If someone mistakenly entered some information into the application and then realized that everything entered was incorrect, it could be subsequently **nullified** and a correct entry created.

The application will behave as if the record never existed. Audit logs can be used to track the events which will say, "somebody entered this information and then they nullified it".

If this application sends the correct V3 message to another application, and the other application understands the V3 message, your record will cease to exist in the other application as well (for non-auditing operations).

7. RoleLink

Defined as "A connection between two roles expressing a dependency between those roles and permitting the authorization or nullification of a dependent role based on status changes in its causal or directing role. The RoleLink may be operated over time and thus whose state and identity must be managed."

Examples: Hierarchical relationship in an organization, relationships between an employee and their assigned duties.

RoleLink attributes are shown in Figure 12:

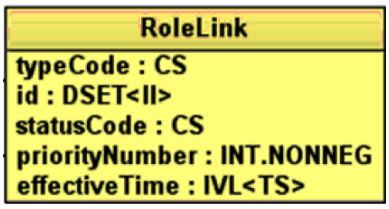


Figure 12: RoleLink Attributes

typeCode: the kind of connection represented by this RoleLink. For example, has-part, has-authority.

effectiveTime: an interval of time specifying the period during which the connection between Roles is in effect.

8. ActRelationship

The ActRelationship is defined as "A directed association between a source Act and a target Act".

The types of act relationships fall under a number of categories. Some examples, in some instances, including both the main category and additional classifications.

1. **Composition**, with composite (source) and component (target). One of the most commonly used ActRelationship types is "has component" to describe the composition and decomposition of Acts. The relationship type allows specifying detail of Acts to varying degrees.
   With the composition relationship, the detail of Acts can be revealed to different levels for different purposes without the structure of the Act hierarchy needing to be rearranged. This allows supporting multiple viewpoints on the same business processes. For instance, a billing viewpoint of a laboratory test battery may be as a single billable act. A clinician's view of the same laboratory test battery is as a set of individual observations, where the ordering among the observations is irrelevant.
2. **Sequel**, which includes; follow-up, fulfillment, instantiation, replacement, transformation, etc., for which source and target are Acts of essentially the same kind but with variances in mood, and where the target exists before the source.
3. **Pre-condition**, trigger, reason, contraindication, with the conditioned Act ("give aspirin") at the source and the condition or reason Act ("if fever above threshold") at the target.
4. **Post-condition**, outcome, goal and risk, with the Act at the source having the outcome or goal at the target.
5. **Workflow** - The composition and sequence relationships can be arranged in a sequence to form temporal and conditional (non-temporal) action plans (e.g., care plan, critical path, clinical trials protocols, drug treatment protocols). There is a group of attributes in both Act and ActRelationship called the "workflow Control suite of attributes" that allow the detailed specification of executable action plans.
6. A host of **functional** relationships including support, cause, derivation, etc. generalized under the notion of "pertinence."

ActRelationship attributes are shown in Figure 13:

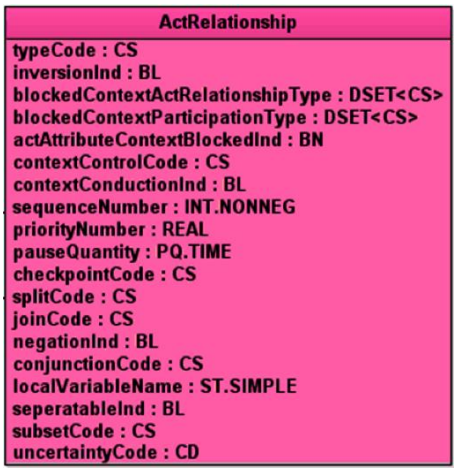


Figure 13 : RoleLink Attributes

typeCode: the category of the relationship: reason, component, sequence.
sequenceNumber: ordering number in a sequence.
checkpointCode, splitCode, joinCode: workflow control attributes. They allow specifying precisely how to follow a care plan or a drug treatment plan.
negationInd: negates the specific relationship between the acts.

Static Models derived from the RIM

The good news is that V3 derives a specific model for each domain in health care from the RIM using processes and tooling. These are beyond the scope of this introductory Course.

The not so good news is that to read these domain models, we will have to learn another notation. HL7 decided that classic UML was not sufficient and defined a notation based on UML to create them.

To understand the outcome of these processes we must first understand the concepts of CONSTRAINING, REFINING and LOCALIZATION.

Refining

The V3 Domain Message Information Model (D-MIM) is a refinement of the RIM. **Refining** means selecting only those classes and attributes appropriate to that domain. Other classes and attributes are omitted from the domain model.

Constraining

Creating a D-MIM also means constraining the RIM. **Constraining** means restricting attributes to a specific vocabulary.

Localization

In order to fulfill the goal of being valid worldwide, the V3 methodology allows affiliates, representing regions or nations, to further refine and constrain D-MIMs to accommodate the requirements of their health-care systems. This is called **localization**.

Figure 14 shows how HL7 refines and constrains the RIM to produce the D-MIM:

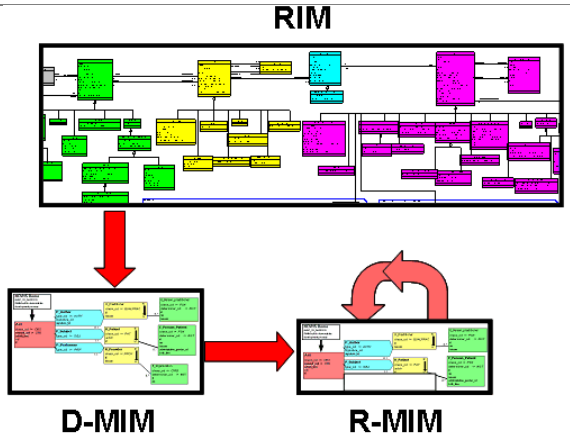


Figure 14: The HL7 V3 Domain Message Information Model

The D-MIM

The first level of models derived from the RIM is a model expressing the reality of a given health-care domain (i.e. **it only shows the Acts that can be recorded, and all the relevant Entities, Roles and Participations**). We are now moving from an abstract notion of the global health-care environment to the reality of healthcare information exchange.

The R-MIM

However, the model that is most relevant is the Refined Message Information Model (R-MIM). The R-MIM is a subset of the D-MIM, used to show all the objects involved in ONE possible interaction in the domain.

The R-MIM is a graphic representation of the messages associated with the interaction. It shows:

- Which objects will need to be transmitted and in which order
- The relationship between these objects
- Which attributes to include, their data types and vocabulary

The R-MIM is a graphical model or diagram of an artifact.

If we want the R-MIM to be useful, we will have to learn how to read it just as we learned to read a UML diagram.

How to Read a R-MIM

Color-Coding

We need to remember the color-coding of the RIM:

Entity = green	Act = red
Role and RoleLink = yellow	ActRelationship = purple
Participation = light blue	Infrastructure = blue

Almost all R-MIMs have the format shown in Figure 15. They are usually read outwards from the center to the left and right, and they contain **ENTITIES playing ROLES, PARTICIPATING in ACTS, RELATED to other ACTS.**

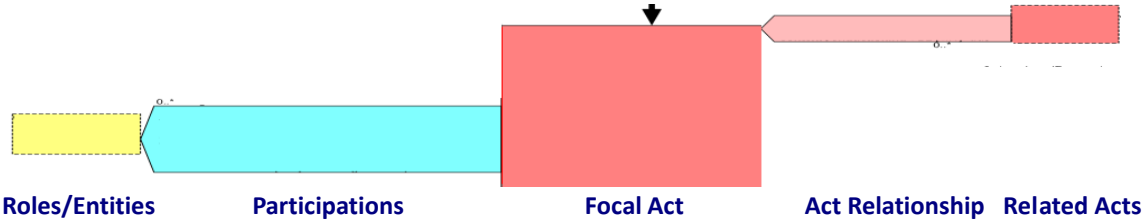


Figure 15: The HL7 V3 R-MIM Structure

Question: Why is there no entity in Figure 15? It is because they are combined with their respective roles, forming CMETs. This concept will be discussed later in the Unit.

The Components of an R-MIM

Figure 16 below shows an example of the components of an R-MIM:

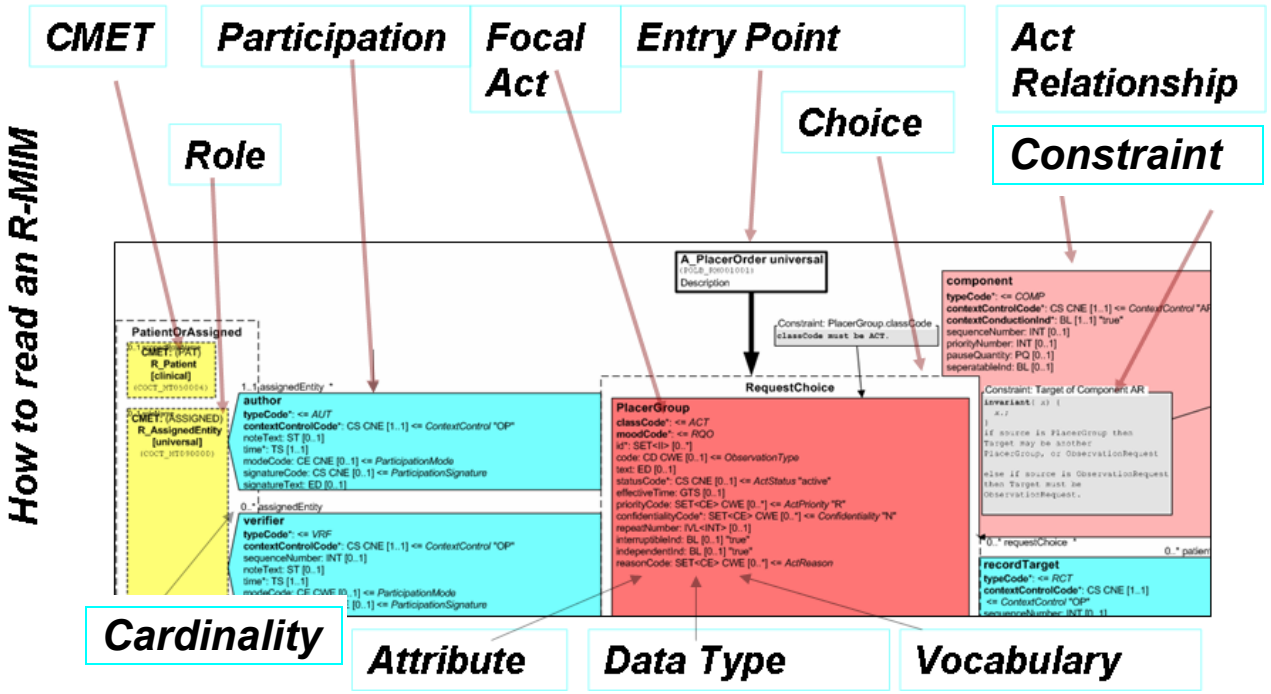


Figure 16 : Example of a V3 R-MIM

Entry Point

The first thing we have to find, much like arriving at a building, is the door. Otherwise, we will not know where to begin or start into the model. An R-MIM door is called an "Entry Point".

Entry points (an R-MIM has only one, but there can be several entry points in a D-MIM) contain a name, code and a brief description. They point to the Focal Act. Figure 17 below shows the three Entry Points in a D-MIM:

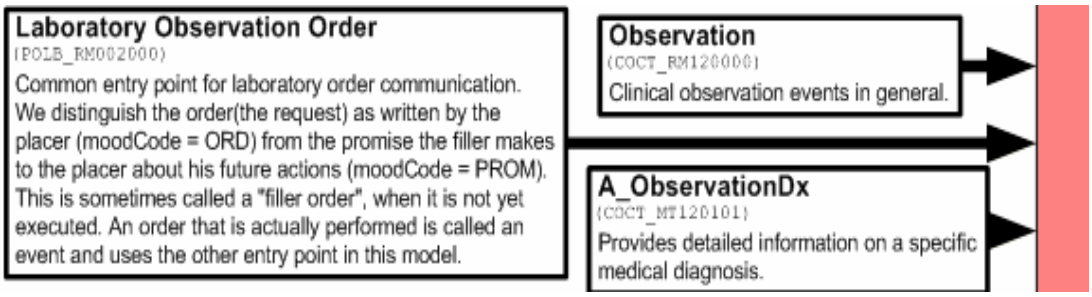


Figure17: Example of a V3 R-MIM

Focal Act

The focal act is the R-MIM's main purpose - the Act containing the information we want to transmit.

The focal act is an Act (highlighted in red) and it is the main focus (as signaled by the entry point).

Figure 18 below shows an example of an entry point (SupportingClinicalInformation) and focal act (SubstanceAdministration):

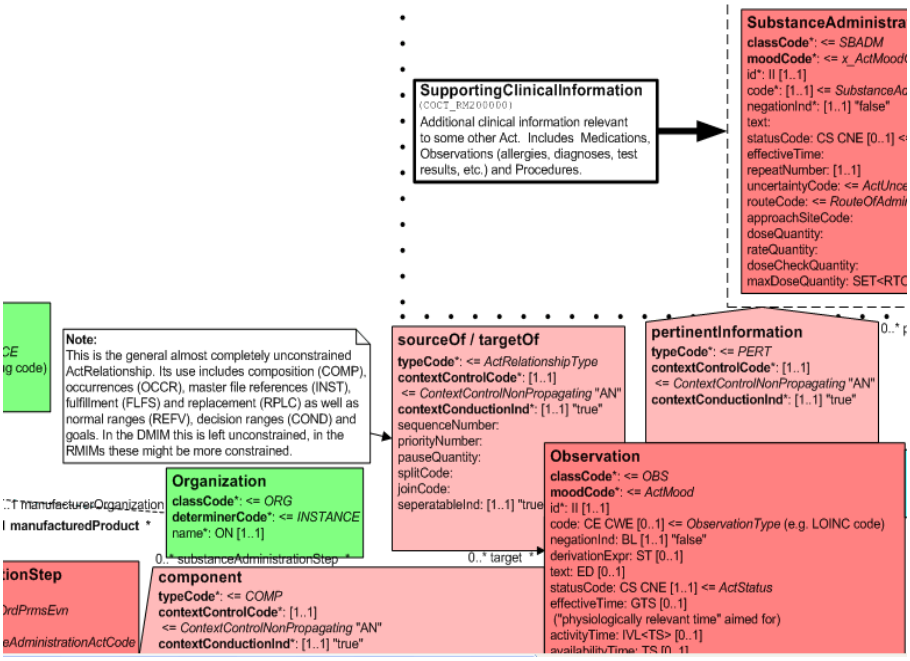


Figure18: Example of a V3 R-MIM

Sometimes the focus of an interaction is not an Act but another class (an Entity or Role) - this usually occurs in the registry domains.
In that case, instead of talking about focal acts, we talk about Focal Classes. However, it is the same mechanism - begin reading from the object indicated by the entry point arrow.

Associations

We see the associations linked to the focal act: participants, roles and act relationships.

Participants

The participants are at both sides of the focal act (highlighted in light blue) and point to the roles.

Roles

The roles are on the left side (highlighted in yellow) usually merged in CMETs with the entities playing them.

Act Relationships

Act relationships (highlighted in purple) point to other Acts or in some cases back to the focal act (see Figure 15) - indicating a possible recursive relationship (the focal act can be related to another instance of the same act class.)

Association Cardinality

The cardinality defines how many times (minimum and maximum) an association can exist in a specific model.

Attributes, Data Types, Vocabulary Domain and Qualifier

Figure 19 below provides examples of the expression of attributes, their data type and the specific vocabulary for coded attributes:

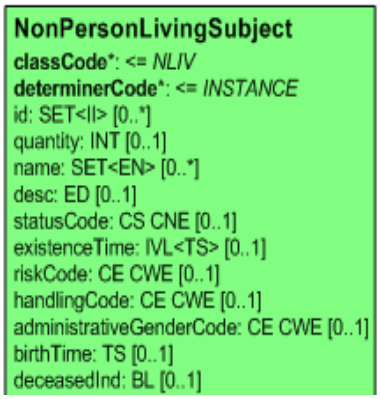


Figure 19: Example of Attributes

Class Name: appears at the top of the box, in this example "NonPersonLivingSubject".

Required Attributes: are shown in bold, in this example "classCode" and "determinerCode".

Data Type: appears after the colon (:) for each attribute.

Examples:

- quantity: INT (integer data type)
- riskCode: CE (coded element data type)
- existenceTime: IVL<TS> (interval data type expressed as time stamp data types)

Cardinality: follows the data type and expresses the number of instances of the attribute that can be included.

Vocabulary Domain: the vocabulary domain for an attribute is expressed following the symbols "<=" or "=".

It may be a fixed value, as shown for **classCode** in Figure 20, or expressed as a domain, in which case the domain is italicized. For example, **statusCode <= *ActStatus*** means that the vocabulary domain (the valid set of values) for **statusCode** is defined by ***ActStatus*** (see "vocabulary and registries" section below).

Vocabulary domain qualifier: In addition to a Vocabulary domain, coded attributes have an Extensibility Qualifier; CNE or CWE.

The CNE value (coded no extensions) means that the code set is fixed and cannot be extended. If the field cannot be valued from the specified domain, then the field cannot be placed in the message. If a CNE field is required in a message, but the field cannot be valued from the specified domain, then no valid message can be created.

The CWE value (coded with extensions) means that the code set can be expanded to meet local implementation needs. When a coded attribute is sent in a message, local concepts or free text may be sent in place of a standard code if the desired concept is not represented in the standard vocabulary domain.

Choice

A choice is used to indicate that we can choose a class from among several options of the same type (role, act, entity).

Choices are presented as groupings of the available classes within a dashed frame - the actual "choice" artifact - which is given a name.

It is important to note that any associations connected to the Choice box apply to all classes within it. Associations connected to a specific class within the choice box apply only to that class:

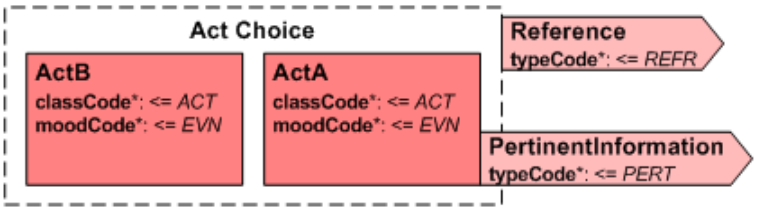


Figure 20 : Example of a Choice Box

9. Vocabulary and Registries

Globally Unique Identifiers

As stated in the HL7 V3 goals, instance identifiers (for persons, buildings, orders, devices, etc.) shall be globally unique. This means that a V3 identifier used anywhere in the world will always refer to the same instance. Vocabularies, code tables or coding systems have unique universal identification.

HL7 International recommends the use of OID or UUID for instance identifiers and uses OIDs to identify coding systems or vocabularies. The component structure of these identifiers is not relevant at this point, just the fact that they are globally unique identifiers.

What is an OID?

An Object Identifier (OID) is a unique ISO identifier represented as a string of positive numbers delimited by periods. HL7 uses the OID as the recommended identifier for all coding systems and registries.

Once assigned by ISO, the HL7 OID is incremented hierarchically and assigned as identifiers for HL7 Affiliates, HL7 Members, regulatory agencies, etc.

The hierarchical nature of an OID is evident in these examples:

OID	Description	Assigned by
2.16.840.1.113883	HL7 International Inc	ISO
2.16.840.1.113883.6	External Coding Systems (assigned by HL7)	HL7 International
2.16.840.1.113883.6.1	LOINC	HL7 International
2.16.840.1.113883.6.2	ICD-9-CM	HL7 International
2.16.840.1.113883.2	HL7 Affiliates	HL7 International
2.16.840.1.113883.2.10	HL7 Argentina	HL7 International
2.16.840.1.113883.2.10.1	Hospital Italiano de Buenos Aires	HL7 Argentina
2.16.840.1.113883.2.10.1.1	Document Identifier	HL7 Argentina

How can I verify if an OID is assigned to a given entity?

HL7 OIDs can be searched and registered here: www.hl7.org/oid/index.cfm

Since the OID is a hierarchical identifier, we can usually trace responsibility upwards, but higher levels of the OID structure **do not necessarily know** how lower levels are assigning their OIDs.

In our example, "HL7 International Inc." does not know how "HL7 Argentina" subsequently assigns its OIDs. There is no a global OID registry and each hierarchical branch of the OID is responsible for maintaining its own registry.

Furthermore, OIDs are hierarchical by nature, but the hierarchy is not intended to be used semantically. They are used solely to guarantee that identifiers are globally unique.

Universally Unique Identifiers (UUID)

The Universally Unique Identifier (UUID) is a unique identifier comprised of five groups of hexadecimal digits. It is' generated using the MAC address, date, time and some random values from computing systems.

This data combination (MAC address + date + time + random) generates an UUID that is 128 bits long and is guaranteed unique across space and time.[4]

It is not the preferred HL7 identifier method for vocabularies, but it can be used for instance identifiers (e.g. documents, etc.).

An example UUID: 6FED4EBC-A6B1-11E3-80BD-63D01D5D46B0

[4] You can generate a UUID on this web site: www.somacon.com/p113.php

10. The V3 XML Implementation Technology Specification (ITS)

An implementation technology specification (ITS) defines how objects derived from the RIM are represented for exchange.
Models derived from the RIM are defined abstractly, without reference to a particular ITS; therefore various ITS may be defined (including additional XML-based ITS).

The **XML Implementation Technology Specification, Release 2** describes how HL7 V3 compliant instances can be expressed using XML. The specification supports the definition of objects, attributes and data types.

Figure 21 below is an example R-MIM and its subsequent Hierarchical Message Descriptor (HMD) - all derived from the RIM; in Figure 21 we can see the mapping to XML elements and attributes:

Figure 21: Example R-MIM and HMD

No	Element Name (Link to tabular view)	Rim Source	of Message Element Type
4	**author**	Act	Author
5	typeCode	Participation	CS
6	**assignedEntity**	Participation	AssignedEntity
7	classCode	Role	CS
8	id	Role	II
9	code	Role	CE

Figure 22 : R-MIM mapping to XML elements and attributes

```
<hl7:author xmlns:hl7="urn:hl7-org:v3" hl7:typeCode="AUT">
  <hl7:assignedEntity>
  <hl7:classCode code="ASSIGNED"/>
  <hl7:id root="2.16.840.1.113883.2.1.4.2"
                                        extension="G0000017"/>
  <hl7:code codeSystem="99.999.99.9.9" code="008"
                                       displayName="Doctor"/>
  </hl7:assignedEntity>
</hl7:author>
```

The XML standard specifies the wire format for messages. It defines how to represent documents as 8-bit byte strings and how to use tags to create "well-formed" XML documents. The XML Document Object Model (DOM) defines how to parse XML documents.

The message specification provides definitions for all the HL7 components: data types, class names, associations, and attribute names.

HL7 developed XML Schemas use a single namespace: **urn:hl7-org:v3**

The XML Schemas provide support to validate HL7 XML instance messages. The specification also supports the message wrapper (transmission and control act), CMETs, etc.

The XML Schemas include specifications for representing Data Types and Structures (classes, associations, choices, interactions, attributes and CMETs) as XML instances.

Steps for V3 Exchanges between a Sender and a Receiver

The steps defined by this specification for object exchange between a sender and a receiver include:

1. The sender extracts the information from its local database.
2. The sender generates a RIM-derived object representing the information.
3. The sender generates a DOM tree (XML instance) representing the object.
4. This XML instance is sent to the receiver.
5. The receiver parses the XML instance (message).
6. The receiver generates a RIM-derived object representing the information received.
7. The receiver stores the information in its local database.

The ITS defines how to get from a message definition and data to the XML instance that represents that message over the wire and how the receiving system takes that information for local use.

Unit Content and Learning Objectives

At the Zoo, next to each animal enclosure there is usually a sign with information about the animal: its natural habitat, family roots, diet and physical characteristics:

African Lion

Scientific Name: *Panthera leo*

Fast Fact:
Lions may rest or sleep about 20 hours each day to conserve their energy.

Lions are perfect hunting machines. They have great binocular vision to judge distances. In bright light, the pupils constrict to round points so the lion can hunt during the day without being blinded by the sun. At night, their hunting ability is enhanced because the pupils of their eyes let in more light by dilating to three times the size of a human's pupils. Keen hearing and sense of smell enable lions to detect hidden prey with ease.

STATUS: African lions are considered vulnerable to extinction in the wild. This situation is mainly due to persecution by humans, over-hunting or -harvesting, and habitat destruction.

HABITAT: Lions roam habitats just south of the Sahara Desert down to South Africa. They have adapted to live in the open grasslands (or savannas), arid woodlands, and semi-desert regions of the continent.

DIET: Lions are carnivores and, therefore, eat meat. Their diet includes prey animals such as impala, wildebeest, zebra, buffalo, rodents, birds, fish, and even ostrich eggs.

PHYSICAL CHARACTERISTIC: The light, golden-brown coats of these big cats provide excellent camouflage in the tall, dry

(The example above was excerpted from the Los Angeles Zoo website www.lazoo.org)

What we want to achieve in this Unit is to explain how each data type is **composed**, **where** it is used, its **relationships** with its **HL7 V2 ancestor** and review an example.

The goals of this Unit are to:

- Understand the HL7 V3 CDA data types and have a general idea of their use
- Be able to distinguish between them merely by seeing their XML representation
- Be able to write a V3 element in XML after seeing the D-MIM.
- Know the equivalence to V2 data types (if they exist)
- Know where to find the XML-ITS specifications for the datatypes used to express each attribute (structures, elements) of a V3 Message or Document and how and where each datatype is commonly used.

In this Unit, we will not study the operations, the formal description or the abstract representation for each datatype. You will not require them for implementation purposes. If you are interested in this information, please follow the links in the references or view the additional reading material.

11. What is a Data Type?

A data type is the minimum size block to build any artifact: messages, documents, etc.

What is the difference between a data type and any other object?

Data types have no state or identity: data types only have values.

Messages or documents are built using elements (if you are thinking of a UML class: attributes) and each element has a data type.

The HL7 V3 data type specification is purely semantic: independent from any representation. This allows better scalability and independence from implementation.

Why Develop a New Data Type Standard?

HL7 needed a new data type standard completely independent from implementation technology. Very few of the existing implementations of data types covered healthcare informatics aspects to such precision and range required for clinical documents including that certain information is not even available to be documented.

Some of the data types presented here are solving old interoperability and localization problems faced by HL7 in previous standards, like:

- How do we know that our **identifiers** for persons, places and applications are **universally unique**?
- How do we know that all variations of **person naming** in all countries can be commonly expressed in a message?
- How do we know that all variations for defining a person's or an organization's **address** can be exchanged without losing meaning or structure?
- How can we leverage **advanced coding systems or classifications** such as SNOMED CT?
- How can we **accurately express dates, times and related conditions** when exchanging information about prescribed medications?
- How can we ensure **true interoperability regarding coding for the units** that are used to express observations?

12. Methodology

HL7 International defines data types and possible operations. In this Unit we will only show you:

Elements or static attributes: components of each data type and a brief description for each component.

Equivalence with V2 data types: there is some equivalence but often V3 data types are more powerful and expressive, so this equivalence may not be bidirectional.

XML samples: simple examples to show how each data type is used. Our samples are always extracted from real CDA documents or V3 messages. We include an explanation of the sample so you will know what each sample means to the system receiving it (and yes, it has the same meaning as the system sending it).

Where is it used: the names of elements or constructions where the data type is usually found in messages or documents.

13. Basic Data Types

The basic data types include the core primitive data types: Boolean, alphanumeric strings, pointers or encapsulated data.

Boolean (BL)

Logical TRUE or FALSE value.

This data type allows null values.

XML Sample

```
<id root="2.16.840.1.113883.19.1" extension="10219201" assigningAuthorityName="ADMISSION SYSTEM" displayable="true" />
```

V2.x equivalent

Not available.

Closest equivalent is an ID value based in table 0532 Expanded Yes/No Indicator.

Boolean not Null (BN)

Logical TRUE or FALSE value,

This data type doesn't allow null values.

XML Sample

```
<id root="2.16.840.1.113883.19.1" extension="10219201" assigningAuthorityName="ADMISSION SYSTEM" displayable="true" />
```

V2.x equivalent

Not available.

The closest equivalent is an ID value based in table 0136 Yes/No Indicator.

Encapsulated Data (ED)

Allows the transmission of binary data as ASCII text. ED can be used for image, audio and video content etc.

Components / Attributes

Component	Description
Data	Text or Base64 conversion of Text (e.g. for binary content)
Representation	Literal: "TXT" or "B64"

Component	Description
mediaType	Data type and representation or interpretation method according to RFC2046: Required (application SHALL support these) "text/plain" "image/png" "image/jpg" "audio/basic" "audio/mpeg" Recommended (application MAY support these) "text/html" "text/x-hl7-ft" "text/x-hl7-text+xml" "image/g3fax" "model/vrml" "application/pdf" "application/dicom" "multipart/x-hl7-cda-level1"
charset	The charset is identified by an Internet Assigned Numbers Authority (IANA) Charset Registration. In XML, derived by default from what is declared in the instance, for example: "US-ASCII" (ASCII) "UTF-8" (UNICODE)
language	The human language defined by RFC 3066. Examples: en-US = English (United States) en-CA = English (Canada) en-AU = English (Australia) fr = French
compression	The compression algorithm, if any, used on the raw byte data. Examples: "DF": Deflate (RFC1951)
reference	A URL the target of which is taken as the binary content of the ED

XML Sample 1 (GIF)

```
<value mediaType="image/gif">
  <reference value=" http://example.com/lefthand.gif"/>
</value>
```

XML Sample 2 (DICOM)

```
<value mediaType="application/dicom">
   <reference value="http://example.com/retrieve.asp?q=882309F930FA3D"/>
</value>
```

XML Sample 3 (GZIP compressed PDF document, base64)

```
<text mediaType='application/pdf'' representation='B64' compression='GZ'>
  omSJUEdmde9j44zmMiromSJUEdmde9j44zmMirdMDSsWdIJdksIJR3373jeu83
  6edjzMMIjdMDSsWdIJdksIJR3373jeu83MNYD83jmMdomSJUEdmde9j44zmMir
  ...
  MNYD83jmMdomSJUEdmde9j44zmMir6edjzMMIjdMDSsWdIJdksIJR3373jeu83
  4zmMir6edjzMMIjdMDSsWdIJdksIJR3373jeu83==
```

```
</text>
```

XML Sample 4 (an image with a reference value, integrity check, and thumbnail)

```
<text mediaType='image/png' representation='B64'
integrityCheck='aA5mb7c8TXtu392KMsaSa2MKkAwL5LKAo2d99azAs3MdUdw'>
  <reference value='http://example.org/xrays/128s8d9ej229se32s.png'>
    <useablePeriod xsi:type='IVL_TS'>
      <low value='201401200845'/>
      <high value='201401290545'/>
    </useablePeriod>
  </reference>
  <thumbnail mediaType='image/jpeg' representation='B64'>
    MNYD83jmMdomSJUEdmde9j44zmMir6edjzMMIjdMDSsWdIJdksIJR3373jeu83
    6edjzMMIjdMDSsWdIJdksIJR3373jeu83MNYD83jmMdomSJUEdmde9j44zmMir
    ...
    omSJUEdmde9j44zmMiromSJUEdmde9j44zmMirdMDSsWdIJdksIJR3373jeu83
    4zmMir6edjzMMIjdMDSsWdIJdksIJR3373jeu83==
  </thumbnail>
</text>
```

V2.x equivalent

Limited equivalence; ED data type for including in line, RP for reference pointers.

Where is it used?

You will find them in **text** or **value** elements.

Character String (ST)

Allows the transmission of text or pointers to text, but only represented as plain text. Mainly used for small strings intended for use by applications.

Components / Attributes

Component	Description
Data	Text to transmit
representation	Literal: "TXT"
mediaType	Literal: "text/plain"
charset	The charset is identified by an Internet Assigned Numbers Authority (IANA) Charset Registration. In XML, derived by default from what is declared in the instance, for example: "US-ASCII" (ASCII) "UTF-8" (UNICODE)
language	The human language defined by RFC 3066. Examples: en-US = English (United States) en-CA = English (Canada) en-AU = English (Australia) fr = French

In the XML ITS, ST strings can be expressed as elements or attributes of more complex elements.

XML Sample

Can be represented as an element or attributes (if one of the attributes is of type ST)

As an attribute: displayName datatype is "ST"

```
<code code="86290005" codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED CT"
  displayName="Respiratory rate"/>
```

As an element: Value datatype is "ST"

```
<value language="en">Chest pain</value>
```

V2.x equivalent

Data Type: ST

Where is it used?

In **displayName** or coded elements, as an attribute or some elements.

Null Flavor

Allows the expression of more details about why a value has not been included.

How is it expressed (when allowed): Using the attribute **nullFlavor** = "value"; with "value" being one of the values in this table:

Value	Meaning
NI	No Information - the value is missing, omitted, incomplete, improper This is the default value
INV	The value as represented in the instance is not a member of the set of permitted data values
OTH	The value is not in the domain for the variable
NINF	Negative infinite
PINF	Positive infinite
DER	Derived - An actual value may exist, but it must be derived from the provided information
UNC	Un-encoded - No attempt has been made to encode the information
UNK	Unknown - A proper value is applicable, but not known.
ASKU	The value was asked for, but is unknown.
NASK	The value was not asked for.
NAV	Temporarily not available. Can be known later.
QS	Sufficient Quantity - the specific quantity is not known, but is known to be non-zero
TRC	Content is greater than zero, but it cannot be quantified.
MSK	The information exists but cannot be revealed based on business rules (policy, privacy, etc.)
NA	Not applicable

XML Sample

```
<representedCustodianOrganization>
  <id extension="1120456789" root="2.16.840.1.113883.19.1.2"
       assigningAuthorityName="HC PAYOR INC" />
   <name>CHH - HIV Unit</name>
   <telecom nullFlavor="ASKU" />
   <addr nullFlavor="MSK" />
</representedCustodianOrganization>
```

Sample Explained: "*We know that the telecom and address for the HIV Unit of the CHH organization must be included but we asked for the phone number but it was not provided and we cannot send the address due to privacy regulations.*"

V2.x equivalent

The CWE data type that was introduced in HL7 V2.5 can specify a null flavor instead of the expected code, identifying the value as coming from table HL7 0353.

```
OBX|1|CWE|883-9^Blood Group^LN|1|NAV^Not
Available^HL70353^^^^2.3.1|||N||F<cr>
```

Where is it used?

In any element that allows null values to provide more information about why a null value is sent.

14. Coded Values

Overview

HL7 V3 and CDA data types are defined to allow different levels of constraints in the expression of coded values, from including just the code (in this case the code system is implicit and cannot be changed), to including several code systems and using qualifiers to change the meaning of the included code:

Type	Name	Description
CS	CODED SIMPLE	Only the code as the coding system is fixed
CV	CODED VALUE	The code and coding system
CO	CODED ORDINAL	Same as CV but the codes are ordered
CE	CODED WITH EQUIVALENTS	Same as CV but allows the term to be coded in a number of ways
CD	CONCEPT DESCRIPTOR	Same as CE but allows qualifiers to support post-coordinated expressions

Coded Simple Value (CS)

CS allows only the inclusion of a coded value in its simplest form. All other components (code system, version) are fixed and given from the context.

It is used when the only reference table or domain for the concept is defined by HL7 or taken only from a given standard (ISO, IETF, etc.)

XML Samples

```
<signatureCode code="S"/>
<languageCode code="en-US"/>
```

Sample Explained: the codes allowed in these attributes are defined by HL7 (the first one) and ISO (the second one). No other code system can be the source for these codes.

V2.x equivalent

CNE - Coded with no exceptions

The code is mandatory, the text is not important. The table is 'HL7 defined'

Where is it used?

In **code** elements.

Coded Value (CV)

It allows the inclusion of a code, code system, code system version and the original text.

Components / Attributes

Component	Description
code	Code value of the concept from the Code System
codeSystem	The global identifier (e.g. OID) for the Code System where the code is defined.
codeSystemName	Name of the Code System.
codeSystemVersion	Version of the Code System.
displayName	The human readable name of the code. It cannot change the meaning of the code, and is included only as a display aid to a person interpreting the message.
originalText	Original text or reference (if available) that was coded. If an element is coded based on free text written by a provider, the free text would be included here.

XML Sample

```
<value code="493.91"
    codeSystem="2.16.840.1.113883.6.2"
    codeSystemName="ICD9-CM"
    displayName="Asthma">
    <originalText>Allergic asthma</originalText>
</value>
```

Sample Explained: The provider entered **'Allergic asthma'** into the system. The system coded **'493.91'** (**Asthma**) from ICD9-CM, which is identified as **2.16.840.1.113883.6.2**

Note that originalText is in the text of an element (<originalText>) and is not expressed as an attribute (like <displayName>).

V2.x equivalent

CWE

Where is it used?

In **code** elements.

Coded with Equivalents (CE)

CE allows the inclusion of a code, code system, code system version and the original text. It also allows the inclusion of translations in other coding systems.

Components / Attributes

Component	Description
code	Code value of the concept from the Code System
codeSystem	The global identifier (eg OID) for the Code System where the code is defined.

Component	Description
codeSystemName	Name of the Code System.
codeSystemVersion	Version of the Code System.
displayName	The human readable name of the code. It cannot change the meaning of the code, and is included only as a display aid to a person interpreting the message.
originalText	Original text or reference (if available) that was coded. If an element is coded based on free text written by a provider, the free text would be included here.
translation (SET)	Translations of the code to other coding systems

XML Sample

```
<code code="HNN.8" codeSystem="2.16.840.1.113883.2.12.1"
      codeSystemName="CISP"
      displayName="Méningite bactérienne">
  <originalText mediaType="text/xml">
     <reference value="antmed-1"/>
  </originalText>
  <translation code="G00.9" codeSystem="2.16.840.1.113883.6.3"
                            codeSystemName="CIM10"/>
</code>
```

Sample Explained: The provider entered some text referenced by '**antmed-1**', the code is **HNN.8** in the code system **CISP** (Classification dite Internationale des Soins Primaires) - note the OID is only an example - and this concept is coded as **G00.9** in code system **CIM10** the French name for **2.16.840.1.113883.6.3 = ICD10**).

V2.x equivalent

CWE, CNE (only up to 1 translation in V2.x)

Where is it used?

In **code** elements.

Concept Descriptor (CD)

CD allows the inclusion of a code, code system, code system version and the original text. It also allows the inclusion of translations in other coding systems and a list of qualifiers that further enhance the semantics of the concept (if the coding system allows the use of qualifiers, such as for post-coordination). Some coding systems that allow the inclusion of qualifiers are SNOMED-CT.

Components / Attributes

Component	Description
code	Code value of the concept from the Code System
codeSystem	The global identifier (e.g. OID) for the Code System where the code is defined.

Component	Description
codeSystemName	Name of the Code System.
codeSystemVersion	Version of the Code System.
displayName	The human readable name of the code. It cannot change the meaning of the code, and is included only as a display aid to a person interpreting the message.
originalText	Original text or reference (if available) that was coded. If an element is coded based on free text written by a provider, the free text would be included here.
translation (SET)	Translations of the code to other coding systems
qualifier (LIST)	List of qualifiers for the concept that provides more information.

XML Sample

```
<code code='396275006' codeSystem='2.16.840.1.113883.19.6.96'
    codeSystemName='SNOMED CT' displayName='Osteoarthritis'>
  <originalText>osteoarthritis of the right knee</originalText>
  <qualifier>
    <name code='363698007' codeSystem='2.16.840.1.113883.19.6.96'
      codeSystemName='SNOMED CT' displayName='finding site'/>
    <value code='6757004' codeSystem='2.16.840.1.113883.19.6.96'
      codeSystemName='SNOMED CT' displayName='right knee'/>
  </qualifier>
</code>
```

Sample Explained: The provider entered 'osteoarthritis of the right knee'. This can be fully expressed using SNOMED CT as the concept 'osteoarthritis' qualified by '**finding site**' valued as **'right knee'**.

V2.x equivalent

Not available.

Where is it used?

It is used in **code** elements.

15. Identifiers

Instance Identifier (II)

The Instance Identifier uniquely identifies an object, thing, place or person. Examples: patient, order, observation, etc.

Components / Attributes

Component	Description
root	The Identifier (OID or UUID) that guarantees that the instance identifier is globally unique, typically known as the namespace. The root alone may be the entire instance identifier.
extension	A string that identifies the object uniquely in the namespace defined by the root. This may be used in combination with the root to create a globally unique identifier. If the extension is not valued, the root alone must be globally unique.
assigningAuthorityName	A string that helps a human reading the message to know the Authority responsible for assigning of the identifier. It is not computable. It is only included to help understand the message.
displayable	Specifies whether the identifier is intended to be shown to human beings (true) or not (false).

Note: The root element must be always present. The use of extension is optional if the root alone defines the instance. This can be the case when identifying organizations that have their own assigned root. The assigningAuthorityName and displayable elements are optional too.

XML Sample

Sample 1 - the root is enough to identify an organization (with an OID)

```
<representedOrganization>
 <id root="2.16.840.1.113883.19.3"/>
</representedOrganization>
```

Sample 2

```
<participant typeCode="PRCP">
  <associatedEntity classCode="CON">
   <id root="2.16.840.1.113883.19.1.5" extension="222-77-7777"
       assigningAuthorityName="HEALTH AUTHORITY WEST "
       displayable="true"/>
   <associatedPerson>
    <name>
     <family>Intern</family>
     <given>Irving</given>
    </name>
```

```
</associatedPerson>
```

Sample Explained: The provider here is **Irving Intern**, identified by **Health Authority West** (namespace authority) with the id **222-777-7777**. This identifier **can be displayed** to humans.

V2.x equivalent

HD Hierarchical Designator (root)

EI Entity Identifier (root/extension)

Where is it used?

In id, templateId, typeId elements

16. Names

Entity Name Part (ENXP)

Entity Name Part is a string used to represent part of a name. It can be associated with a type code to specify which part of the name it is, and with a qualifier to further qualify the name alias, etc.

HL7 V3 allows us to define for each kind of name (i.e. persons, organizations, etc.) which parts are allowed to be included.

Type codes:

Entity Part Name	Description	Component Name
FAM	Last name	**family**
GIV	First name	**given**
PFX	Prefix (as in **Mr.**)	**prefix**
SFX	Suffix (as in **MD**)	**suffix**
DEL	Delimiter (as in **,**)	**delimiter**

Some example qualifiers:

Qualifier	Description
LS	Legal status
SP	Spouse last name
CL	Preferred name
IN	Only initials
TITLE	Title

Entity Name (EN)

Entity Name is the name for a person, organization, place or thing

It's a sequence of name parts (ENXP), and may have additional attributes like valid date range and usage information.

Components / Attributes

Component	Description
use	A set of codes advising which name in a set of names to select for a given purpose. Examples: legal, pseudonym, search, license, indigenous, etc.
validTime	An interval of time specifying the time during which the name is or was used for the entity.
name part (LIST)	The Entity Name with a standard ordering imposed on the parts (see ENXP above)

Trivial Name (TN)

A specialization of ENTITY NAME that only has ONE part name and no qualifiers.

Used especially for PLACES or THINGS.

XML Sample

```
<name>Sydney Harbour Bridge</name>
```

Person Name (PN)

A person name is a sequence of entity name parts (ENXP) along with any of the Entity Name Part Qualifiers applicable to Persons. . "". A PN can also have attributes like validTime and use.

Components / Attributes

Component	Description
use	A set of codes advising which name in a set of names to select for a given purpose. For example; pseudonym, search, indigenous, etc.
validTime	An interval of time specifying the time during which the name is or was used for the entity.
name part (LIST)	The Entity Name with a standard ordering imposed on the parts (see ENXP above)

XML Samples

Sample 1: The name is included as a whole (no part names)

```
<name>Bunion, Paul</name>
```

Sample 2: Family Name and Given Name

```
<name>
 <family>Bunion</family>
 <given>Paul</given>
</name>
```

Sample 3: Two family names, two given names (usual in Spain and Latin America)

```
<name>
 <family>García</family>
 <family>Márquez</family>
  <given>Gabriel</given>
  <given>José</given>
  <given>De la Concordia</given>
</name>
```

Sample 4: With delimiter, academic title and initials (the provider's name is **Robert H. Dolin, MD**)

```
<name>
 <family>Dolin</family>
 <given>Robert</given>
```

```
  <given qualifier="IN">H</given>
  <delimiter>,</delimiter>
  <suffix qualifier="AC">MD</prefix>
</name>
```

V2.x equivalent

XPN

Where is it used?

In **name elements** included in 'person', elements like assignedPerson, contactPerson, patientPerson, etc.

Organization Name (ON)

An organization name is a sequence of name parts to represent the name of an organization. It can also have attributes such as validTime and use.

Components / Attributes

Component	Description
use	A set of codes advising which name in a set of names to select for a given purpose (codes only relevant to organizations)
validTime	An interval of time specifying the time during which the name is or was used for the entity.
name part (LIST)	The Entity Name with a standard ordering imposed on the parts (see ENXP above) and only relevant parts to organization names.

XML Sample

```
<name>Good Health Clinic</name>
```

V2.x equivalent

XON

Where is it used?

In **name** elements included in information about organizations.

17. Addresses

Address Part (ADXP)

The Address Part is a string defining the role played by a part of a postal address.

Each country has its own variation depending on how it uses the different parts for postal addresses. The address parts include political and geographical regions:

Address Part Name	Description	Component Name
STR	Street Name	streetName
DIR	Direction (as in South, North, E)	direction
CNT	Country	country
STA	State or Province	state
ZIP	Postal code	postalCode
CTY	City	city
DEL	Delimiter	delimiter

Postal Address (AD)

The postal address for work or home, or belonging to an organization. Composed of address parts.

Components / Attributes

Component	Description
Use	A set of codes advising the purpose of the address. Examples: H=Home, WP=Workplace, etc.
useablePeriod	The address validity period.
address part (LIST)	List of address parts (from ADXP)

XML Sample

Sample 1

```
<addr use="H">
 <city>ANN ARBOR</city>
 <state>MICHIGAN</state>
 <country>USA</country>
 <streetName>Healthcare Drive</streetName>
 <buildingNumberNumeric>444</buildingNumberNumeric>
</addr>
```

Sample Explained: Home address is 444 Healthcare Drive, Ann Arbor, Michigan, USA

Sample 2

```
<addr use='WP'>
 555 INSURERS CIRCLE
 Ann Arbor, MI 48104
```

```
  USA
</addr>
```

V2.x equivalent

XAD

Where is it used?

In **addr** elements included with person or organization demographic information.

18. Communications

Universal Resource Locator (URL)

A communication address specified according to RFC2396.

The URLs will always be expressed as **scheme:address**

Address scheme can be: fax / file / ftp / http / mailto / mllp / modem / nfs / tel / telnet

The format of the address depends on the type.

XML Sample

Sample 1: MLLP (hl7 interface)

```
<telecom value="mllp://137.212.212.312:8888/ "> </telecom>
```

Sample 2: TEL (phone number)

```
<telecom value="tel:+13176307960"/>
```

Sample 3: TEL (phone number)

```
<telecom value="tel:+1(317)630-7960"/>
```

Note: The phone number in Sample 2 is the same as in Sample 3.

V2.x equivalent

XTN - Extended telecommunication number

Telecom Address (TEL)

You can express the useable period and the type of phone (home, workplace) besides including the URL for the address.

Components / Attributes

Component	Description
useablePeriod	Period of validity
use	One or more codes advising a system or user which telecommunication address in a set of like addresses to select for a given telecommunication need. Examples: H = Home WP = Workplace DIR = Direct EC = Emergency Contact
value	Type and address (see URL)

XML Sample

Sample 1 - Emergency phone

```
<telecom use="EC" value="tel:+13176307960"/>
```

V2.x equivalent

XTN - Extended telecommunication number

Where is it used?

In **telecom** elements.

19. Quantities

Integers, reals, ratios, quantities, etc. which are used to report measurements.

Integer (INT)

Expressed as a string of decimal digits without a decimal point. It can contain a sign at the beginning.

XML Sample

```
<versionNumber value="1"/>
```

V2.x equivalent

NM (without decimal point)

Where is it used?

It is found in versionNumber, priorityNumber, sequenceNumber elements.

You can also find it in **value** elements, e.g. as a result of an observation, we can include a value of this type using **xsi:type="INT"** which will ensure that the number is recognized as an Integer.

```
<value xsi:type="INT" value="-100"/>
```

Real (REAL)

Expressed as a string of decimal digits, with a decimal point. It can also include a sign (e.g. + or -).

XML Sample

```
<value value="1.32"/>
```

V2.x equivalent

NM (with decimal point)

Where is it used?

You can find it in value elements, e.g. as a result of an observation, we can include a value of this type using **xsi:type="REAL".**

```
<value xsi:type="REAL" value="12.32"/>
```

Ratio (RTO)

A quantity constructed as the quotient of a numerator quantity divided by a denominator quantity. The Unit attribute is used to describe the dimension measured.

XML Sample

```
<maxDoseQuantity>
 <numerator xsi:type='PQ' value='25' unit='mg'/>
 <denominator xsi:type='PQ' value='5' unit='mL'/>
</maxDoseQuantity>
```

V2.x equivalent

SN (structured numeric, with separator = ":")

Where is it used?

It is used in **value** elements. The name of the element is given by context.

Physical Quantity (PQ)

Physical Quantity is a numeric quantity, which includes the units of measurement associated with that quantity. Thus, it expresses a measurable quantity.

Components / Attributes

Component	Description
value	Quantity measured in the expressed unit.
unit	The unit of measure specified in the Unified Code for Units of Measure (UCUM). See www.unitsofmeasure.org
translation (PQR)	An alternative representation for the quantity, expressed in another system of units.

Translation (PQR type) is composed from:

Component	Description
value	Quantity measured in the expressed unit
code	Code for the unit in the alternative coding system
codeSystem	Identifier for the alternative code system
codeSystemName	Name for the alternative code system
codeSystemVersion	Version of the alternative code system
displayName	Human Readable Description for the code

XML Sample

Sample 1: Only quantity

```
<value xsi:type="PQ" value="12.34" unit="mmol/L"/>
```

Sample 2: Quantity in meters and inches

```
<value value='1.77' unit='m'>
 <translation value='69.7' code='[in_i]' codeSystem='2.16.840.1.113883.19.6.8'
  codeSystemName='UCUM'/>
</value>
```

Sample Explained: The value is 1.77 meters. It can be expressed also as 69.7 inches

V2.x equivalent

PQ

Where is it used?

Usually used in **value** element of observations, including xsi:type.

Monetary Amount (MO)

A quantity expressing an amount in a given currency.

Components / Attributes

Component	Description
value	Amount measured in the currency
currency	Unit according to ISO 4217

Currency examples:

EUR = Euros	MXN = Mexican Nuevo Peso
USD = American Dollars	AUD = Australian Dollers

XML Sample

```
<amount value="12.50" currency="EUR"/>
```

V2.x equivalent

MO

Where is it used?

It is used in **amt**, **balanceAmt** elements.

20. Collections

Collections are abstract collection of different data types, including bags, sets, and lists. These collection data types can be used with any of the other data types..

They are expressed as follows:

- SET (<II>) means a SET of elements of the II data type
- LIST (<CD>) means a LIST of elements of the CD data type

Set (SET)

A set of values with no specific order that cannot contain null values.

XML Sample:

Patient with two identifiers (set<ii>):

```
<patient>
 <patientRole>
       <id extension="401233-2" root="2.16.840.1.113883.19.35.2.4"/>
       <id extension="90492012" root="2.16.840.1.113883.19.35.2.5"/>
 </patientRole>
 ...
</patient>
```

V2.x equivalent:

Repetitive field

Where is it used?

In the elements defined as SET(<DATA TYPE >)

Sequence (LIST)

A set of values in the specified order that cannot contain null values. The order of the items is important and must be preserved.

. XML Sample: LIST(INT)

```
<regionOfInterest classCode='ROIOVL'>
  <code code='ELLIPSE'/>
  <value value='3'/>
  <value value='1'/>
  <value value='3'/>
  <value value='7'/>
  <value value='2'/>
  <value value='4'/>
  <value value='4'/>
```

```
  <value value='4'/>
</regionOfInterest>
```

V2.x equivalent:

Repetitive field

Where is it used?

In the elements defined as LIST(<DATA TYPE >)

Interval (IVL)

An Interval is a range from one point to another, expressed through variables.

Components / Attributes

Component	Description
low	Lowest boundary
high	Highest boundary
center	Interval center
width	Width of the interval

Variations in its use can include: range (**low**, **high**), only **low**, only **high** or (**center, width)**

IVL can also include attributes to indicate if the specified values are included in the interval:

- lowClosed - Specifies whether LOW is included in the IVL
- highClosed - Specifies whether HIGH is included in the IVL

The default value is true (i.e. closed).

XML Sample: IVL<PQ>

An interval of PQ between 2.8 meters, inclusive, and 4.6 meters, exclusive:

```
<example xsi:type='IVL_PQ' lowClosed='true' highClosed='false'>
  <low value='2.8' unit='m'/>
  <high value='4.6' unit='m'/>
</example>
```

V2.x equivalent:

Not available

Where is it used?

In the elements defined as IVL(<DATA TYPE >)

21. Time

These data types include point in time (TS), time interval (IVL_TS) and types for support of events that repeat during a period of time.

These last types include: periodical intervals, event related periodical interval, and a general timing specification. These data types describe the temporal distribution of the data.

Point in Time (TS)

A quantity specifying a point in time. Expressed as an alphanumeric string compliant with the ISO 8601 standard.

The valid format is YYYYMMDDHHMMSS.UUUU[+|-ZZzz]

Digits can be omitted from right to left to express less precision.

The most common formats are YYYYMMDD or YYYYMMDDHHMM, but all legal "precision" values are allowed. These range from specifying only the year (e.g. YYYY) to specifying a precision of 0.0001 seconds (YYYYMMDDHHMMSS.UUUU). The optional addition of [+|-ZZzz] specifies the time zone as an offset from universal coordinated time (UCT).

XML Sample:

```
<time value="20140410103000"/>
```

V2.x equivalent:

DTM

Where is it used?

In elements like **time.**

Interval of Point in Time (IVL<TS>)

An interval of time, expressed through variables.

Components / Attributes

Component	Description
low	Lower boundary (earliest point in time)
high	Upper boundary (latest point in time)
center	Interval center
width	Interval width

Variations in its use can include: date range (**low**, **high**), only **low**, only **high** or (**center, width)**.

XML Sample

Sample 1: From 15 Jan 2013 10:30 AM to 20 Jan 2013 10:50 AM

```
<effectiveTime>
 <low value="20130105103000"/>
 <high value="20130120105000"/>
</effectiveTime>
```

Sample 2: From 1 AUG 2013

```
<effectiveTime>
 <low value="20130801"/>
</effectiveTime>
```

Sample 2: Until 2 Aug 2013

```
<effectiveTime>
 <high value="20130802"/>
</effectiveTime>
```

Event Related Periodic Interval of Time (EIVL)

A time interval recurring periodically after a normal life event, e.g. lunch, sleeping, etc.

Components / Attributes

Component	Description
event	The code describing the activity. Examples: AC/PC = Before/After meal ACD/PCD = Before/after lunch ACV/PCV = Before/after dinner HS = Sleeping time IC = Between meals ICD = Between lunch and dinner
offset	Time that must pass before activating the activity

XML Sample

Sample: 30 minutes after dinner

```
<effectiveTime xsi:type='EIVL_TS'>
  <event code='PCV' codeSystem='2.16.840.1.113883.5.139'/>
  <offset>
    <center value='30' unit ='min'/>
  </offset>
</effectiveTime>
```

V2.x equivalent

TQ, GTS

Where is it used?

Usually these data types are used in Pharmacy domains (medications dispense and administration)

General Timing Specification (GTS)

A set of temporal elements SET(<TS>), which can include IVL_TS, PIVL, or EIVL

V2.x equivalent

GTS

Where is it used?

In activityTime and effectiveTime elements.

Usually these complex time-related data types are used in Pharmacy domains (e.g. in medication dispense and administration messages)

Periodic Interval of Time (PIVL)

An interval of time that recurs periodically. PIVL has two properties, phase and period/frequency. Phase specifies the "interval prototype" that is repeated every period or frequency

Components / Attributes

Component	Description
phase	An interval specifying the duration of each occurrence and anchoring the PIVL sequence at a certain point in time.
period	A time duration specified as a reciprocal measure of the frequency at which the PIVL repeats.
frequency	The number of times the PIVL repeats (numerator) within a specified time-period (denominator).
isFlexible	Indicates whether the exact timing is up to the party executing the schedule. For example; to distinguish "every 8 hours" from "3 times a day". *Note: this is sometimes referred to as "institution specified timing"*
alignment	Specifies if and how the repetitions are aligned to the cycles of the underlying calendar (e.g. to distinguish every 30 days from "the 5th of every month").

XML Sample

Twice a day (The actual time is at the discretion of the institution)

```
<example xsi:type='PIVL_TS' isFlexible='true'>
  <period value='12' unit='h'/>
</example>
```

Twice a day, 10 minutes each time:

```
<effectiveTime xsi:type='PIVL_TS'>
  <phase>
    <width value='10' unit='min'/>
  </phase>
  <period value='12' unit='h'/>
</effectiveTime>
```

V2.x equivalent

TQ, GTS

Unit Summary and Conclusion

This Unit gave a detailed introduction into the HL7 V3 RIM and data types.

Additional Reading Material

All references can be found in the HL7 V3 product suite:

HL7 Version 3 Standard: Data Types - Abstract Specification, Release 2 - download at: www.hl7.org/implement/standards/product_brief.cfm?product_id=264

HL7 Version 3 Standard: XML Implementation Technology Specification R2: ISO-Harmonized Data Types, Release 1 - download at: www.hl7.org/implement/standards/product_brief.cfm?product_id=48

More about the RIM and ITS

The HL7 RIM can be downloaded from www.hl7.org/implement/standards/rim.cfm
The HL7 ITS can be downloaded from www.hl7.org/implement/standards/product_brief.cfm?product_id=30

Note: You can download the HL7 V3 standards free from the Health Level 7 International web site (www.HL7.org) after you create a free log-in account.

More about OIDs and Registries

Detail of OIDs registered by HL7 International Inc.: www.hl7.org/oid/index.cfm
Whitepaper on the use of OIDs in the European Union: www.ringholm.de/docs/00900_en.htm