

Federal Health Information Model (FHIM)
FHIM Datatypes
Draft Document

Contents

Acknowledgments.....	5
Revision History.....	7
 Chapter 1: INTRODUCTION.....	 9
Overview.....	10
Approach.....	10
Implementation Guide Scope.....	10
Audience.....	10
Organization of This Guide.....	10
Conventions Used in This Guide.....	10
 Chapter 2: DATATYPES.....	 11
Address.....	13
Address Type.....	13
Any.....	14
Birth Address.....	14
CD.....	15
Code.....	15
Coded Ordinal.....	15
HL72 Coded Element.....	15
HL72 Entity Identifier.....	16
HL72 Extended Composite Id.....	16
HL72 Hierarchic Designator.....	18
HL72 Organization Identifier Extended.....	18
HL72 Person Location.....	19
HL73 Coded Simple Value.....	20
HL73 Concept Descriptor.....	20
HL73 Id.....	21
Id.....	21
Integer Interval.....	22
Integer Ratio.....	22
Monetary Amount.....	22
Non Us Mailing Address.....	23
Nullable Boolean.....	23
Null Flavor.....	23
Organization Name.....	25
Person Name.....	25
Physical Quantity.....	26
Physical Quantity Interval.....	26
Physical Quantity Ratio.....	26
Point In Time.....	27
Rate Quantity.....	27
Real.....	28
Telecommunications.....	28
Time Interval.....	28
Time Quantity.....	29
Us Mailing Address.....	29
VA Code.....	29
REFERENCES.....	31

Acknowledgments

NEED FHIM BOILERPLATE

Revision History

Rev	Date	By Whom	Changes

Chapter 1

INTRODUCTION

Topics:

- [Overview](#)
- [Approach](#)
- [Implementation Guide Scope](#)
- [Audience](#)
- [Organization of This Guide](#)
- [Conventions Used in This Guide](#)

NEED FHIM BOILERPLATE

Overview

NEED FHIM BOILERPLATE

Approach

NEED FHIM BOILERPLATE

Implementation Guide Scope

NEED FHIM BOILERPLATE

Audience

NEED FHIM BOILERPLATE

Organization of This Guide

NEED FHIM BOILERPLATE

Conventions Used in This Guide

NEED FHIM BOILERPLATE

Chapter

2

DATATYPES

Topics:

- [*Address*](#)
- [*Address Type*](#)
- [*Any*](#)
- [*Birth Address*](#)
- [*CD*](#)
- [*Code*](#)
- [*Coded Ordinal*](#)
- [*HL72 Coded Element*](#)
- [*HL72 Entity Identifier*](#)
- [*HL72 Extended Composite Id*](#)
- [*HL72 Hierarchic Designator*](#)
- [*HL72 Organization Identifier Extended*](#)
- [*HL72 Person Location*](#)
- [*HL73 Coded Simple Value*](#)
- [*HL73 Concept Descriptor*](#)
- [*HL73 Id*](#)
- [*Id*](#)
- [*Integer Interval*](#)
- [*Integer Ratio*](#)
- [*Monetary Amount*](#)
- [*Non Us Mailing Address*](#)
- [*Nullable Boolean*](#)
- [*Null Flavor*](#)
- [*Organization Name*](#)
- [*Person Name*](#)
- [*Physical Quantity*](#)
- [*Physical Quantity Interval*](#)
- [*Physical Quantity Ratio*](#)
- [*Point In Time*](#)
- [*Rate Quantity*](#)
- [*Real*](#)
- [*Telecommunications*](#)
- [*Time Interval*](#)
- [*Time Quantity*](#)
- [*Us Mailing Address*](#)

This section describes Datatypes used in the FHIM

- [VA Code](#) |

Address

A physical address at which the person resides or may be contacted. 7/8/10: Renamed from MailingAddress to Address. Moved county from Person to here.

1. Contains exactly one [1..1] **addressType** with data type *Address Type*
 - Indicates the kind of address that is contained within this class. Examples include primaryHome, Work, etc. Note that in HL7 V3, this concept is part of the Address datatype (the 'use code'). This concept is made explicit in this Address class, because this is a platform-independent model - non V3 implementations will need other mechanisms to deal with the type.
- 2.
3. Contains exactly one [1..1] **line1** with data type *String*
 - The first line of a mailing address. Unlike HL7, we have chosen not to break up the parts of each line.
 - The first line of the street address. While this street address could be broken into several constituent parts, for the purpose of this logical model, the whole line is treated as a single concept.
4. Contains zero or one [0..1] **line2** with data type *String*
 - The second line of a mailing address. Unlike HL7, we have chosen not to break up the parts of each line.
 - The second line of the street address. While this street address could be broken into several constituent parts, for the purpose of this logical model, the whole line is treated as a single concept.
5. Contains zero or one [0..1] **line3** with data type *String*
 - The third line of a mailing address. Unlike HL7, we have chosen not to break up the parts of each line.
 - The first line of the street address. While this street address could be broken into several constituent parts, for the purpose of this logical model, the whole line is treated as a single concept. Note that the third line is rarely used.
6. Contains exactly one [1..1] **city** with data type *String*
 - An Address Part (ADXP) that contains the name of the city, town, village, or other community or delivery center.
 - "The name of the city, town, village, or other community or delivery center." - HL7 V3
7. Contains zero or one [0..1] **county** with data type *Code*
 - A region created by territorial division for the purpose of local government. In the United States, a county (or parish in Louisiana) is the largest administrative district within a state. This property is used primarily for statistical and pricing information (i.e., the same service may be more expensive in an affluent section of the country than in a less-affluent portion).
8. Contains exactly one [1..1] **postalCode** with data type *String*
 - An Address Part (ADXP) that contains a postal code designating a region defined by the postal service.
 - A code designating a region defined by the postal service
9. Contains zero or one [0..1] **country** with data type *Code*
 - An Address Part (ADXP) that contains the Country of the address.
 - A state or nation

Address Type

Value Set	AddressType
Description	This enumeration describes types of Addresses at which a person or organization exists or can be reached.

Concept Code	Concept Name	Code System	Description
mobileContact			A telecommunication device that moves and stays with its owner. May have characteristics of all other use codes, suitable for urgent matters, not the first choice for routine business.
pager			A paging device suitable to solicit a callback or to leave a very short message.
primaryHome			The primary home, to reach a person after business hours.
temporaryAddress			The temporary address where a person resides. An address that is different from the permanent address, but at which the person is residing for a limited, defined period of time. Note that for military personnel, this address may represent a location at which the person is temporarily assigned or deployed.
vacationHome			A vacation home, to reach a person while on vacation
workPlace			An office address. First choice for business related contacts during business hours.

Any

This **abstract** class is used to represent a datatype that is not known at the logical model level, but rather will be substituted with a (set of) real data type(s) when transformed to a given platform.

1.

Birth Address

A simplified version of a Postal Address which contains only a city, state, and country, all of which are optional. This is used to record a person's place of birth, used for identification and statistical purposes.

1. Contains zero or one [0..1] **city** with data type *String*
 - An Address Part (ADXP) that contains the name of the city, town, village, or other community or delivery center.
 - "The name of the city, town, village, or other community or delivery center." - HL7 V3
2. Contains zero or one [0..1] **state** with data type *String*
 - An Address Part (ADXP) that contains a sub-unit of a state or province. A sub-unit of a country with limited sovereignty in a federally organized country.
 - An Address Part (ADXP) that contains the state or province. A state or provinces is a sub-unit of a country with limited sovereignty in a federally organized country.
3. Contains zero or one [0..1] **country** with data type *Code*
 - An Address Part (ADXP) that contains the Country of the address.
 - A state or nation

CD

Temporary class used to trigger MDHT Terminology Constraints

- 1.

Code

"A word, letter, number, or other symbol used in a code system to mark, represent, or identify something: The code on the label shows the date of manufacture." - Dictionary.com This **abstract** datatype represents a "coded element" - some series of letters or numbers which can be "looked up" in a code system or value set. In practice, this datatype will be substituted with specific "flavors" of the code datatype for use in particular standards-based payloads.

1. Extends *CD*
2. Contains zero or one [0..1] **code** with data type *String*
 - This is a placeholder for the "code", which is a unique identifier of an entry in a coding system. Note that this property will be replaced by some other concept or group of concepts when this abstract class is replaced by a specific class for a given target platform.
3. Contains zero or one [0..1] **text** with data type *String*
 - This is a placeholder for the "display text" or "designation", which is the human-readable string representing the concept. Note that this property will be replaced by some other concept or group of concepts when this abstract class is replaced by a specific class for a given target platform.
4. Contains zero or one [0..1] **codeSystem** with data type *String*
 - This is a placeholder for the "coding system", which is a identifies the coding scheme or coding system of which the code is a member. Note that this property will be replaced by some other concept or group of concepts when this abstract class is replaced by a specific class for a given target platform.
5. Contains zero or one [0..1] **codeSystemVersion** with data type *String*
 - This is a placeholder for the "coding system version", which is identifies the version coding scheme or coding system of which the code is a member. Note that this property will be replaced by some other concept or group of concepts when this abstract class is replaced by a specific class for a given target platform. This version number identifies the possible members of a coding system as of a particular time. If a coding system isn't formally versioned, such as NDC, the date on which the coding system was published could be used here.

Coded Ordinal

1. Contains exactly one [1..1] **value** with data type *Real*

HL72 Coded Element

Represents the HL7 V2 Coded Element (CE) datatype. The CE datatype is made up of 6 pieces: Identifier, Text, Name of Coding System, Alternate Identifier, Alternate Text, and Alternate Name of Coding System.

1. Extends *Code*
2. Contains zero or one [0..1] **identifier** with data type *String*
 - "Sequence of characters (the code) that uniquely identifies the item being referenced. Different coding schemes will have different elements here." - HL7 V2
3. Contains zero or one [0..1] **text**

- "The descriptive or textual name of the identifier, e.g., myocardial infarction or X-ray impression." - HL7 V2
4. Contains zero or one [0..1] **codingSystem** with data type *String*
 - "Identifies the coding scheme being used in the identifier component. The combination of the identifier and name of coding system components will be a unique code for a data item. Each system has a unique identifier." - HL7 V2
 5. Contains zero or one [0..1] **alternateIdentifier** with data type *String*
 - "An alternate sequence of characters (the code) that uniquely identifies the item being referenced. See usage note in section introduction." - HL7 V2
 6. Contains zero or one [0..1] **alternateText** with data type *String*
 - "The descriptive or textual name of the alternate identifier. See usage note in section introduction" - HL7 V2
 7. Contains zero or one [0..1] **alternateCodingSystem** with data type *String*
 - "Identifies the coding scheme being used in the alternate identifier component." - HL7 V2

HL72 Entity Identifier

This class corresponds to the HL7 V2 Entity Identifier (EI) data type. "The entity identifier defines a given entity within a specified series of identifiers. The EI is appropriate for, but not limited to, machine or software generated identifiers. The generated identifier goes in the first component. The remaining components, 2 through 4, are known as the assigning authority; they identify the machine/system responsible for generating the identifier in component 1." - HL7 V2

1. Extends *Id*
2. Contains zero or one [0..1] **identifier**
 - "The first component, <entity identifier>, is usually defined to be unique within the series of identifiers created by the <assigning authority>, defined by a hierarchic designator, represented by components 2 through 4." - HL7 V2
3. Contains zero or one [0..1] **namespaceId** with data type *String*
 - "The assigning authority is a unique identifier of the system (or organization or agency or department) that creates the data." - HL7 V2
4. Contains zero or one [0..1] **universalId** with data type *String*
 - "The <universal ID> (UID), is a string formatted according to the scheme defined by the <universal ID type> (UID type) component. The UID is intended to be unique over time within the UID type. It is rigorously defined. Each UID must belong to one of the specifically enumerated schemes for constructing UIDs (defined by the UID type). The UID must follow the syntactic rules of the particular universal identifier scheme (defined by the third component). Note that these syntactic rules are not defined within HL7 but are defined by the rules of the particular universal identifier scheme." - HL7 V2
5. Contains zero or one [0..1] **universalIdType** with data type *String*
 - "The <universal ID type> (UID type) component governs the interpretation of the <universal ID> (UID) component. If the UID type is a known UID refer to HL7 Table 0301 - Universal ID type for valid values, then the UID is a universal ID of that type." - HL7 v2

HL72 Extended Composite Id

This class corresponds to the HL7 V2 Extended Composite ID With Check Digit (CX) data type, which is "used for specifying an identifier with its associated administrative detail". This datatype has been simplified in that some optional components, such as the check digits, which are not used in current HL7 v2 messaging within VA, have been omitted. In addition, the subcomponents which are themselves complex types have been exploded in order to have all subcomponents in the type. For example, the Assigning Authority component in the HL7 v2

datatype is an HD datatype, which has three components. This has been replaced by three properties in this class: AssigningAuthNamespaceId, AssigningAuthUniversalId, and AssigningAuthUniversalIdType.

1. Extends *Id*
2. Contains exactly one [1..1] **entityIdentifier** with data type *String*
 - "The value of the identifier itself" - HL7 v2
3. Contains exactly one [1..1] **assigningAuthNamespaceId** with data type *String*
 - The Assigning Authority is "a unique name of the system (or organization or agency or department) that creates the data" - HL7 v2. Note that Assigning Authority in HL7 v2 is a complex type (an HD datatype). This property contains the Namespace Id sub-component, which is "used as the HL7 identifier for the user-defined table of values for this component."
4. Contains zero or one [0..1] **assigningAuthNameUniversalId** with data type *String*
 - The Assigning Authority is "a unique name of the system (or organization or agency or department) that creates the data" - HL7 v2. Note that Assigning Authority in HL7 v2 is a complex type (an HD datatype). This property contains the Universal Id sub-component, which is defined thusly: "The <universal ID> (UID), is a string formatted according to the scheme defined by the <universal ID type> (UID type) component. The UID is intended to be unique over time within the UID type. It is rigorously defined. Each UID must belong to one of the specifically enumerated schemes for constructing UIDs (defined by the UID type). The UID must follow the syntactic rules of the particular universal identifier scheme (defined by the third component). Note that these syntactic rules are not defined within HL7 but are defined by the rules of the particular universal identifier scheme." - HL7 v2
5. Contains exactly one [1..1] **assigningAuthNameUniversalIdType** with data type *String*
 - The Assigning Authority is "a unique name of the system (or organization or agency or department) that creates the data" - HL7 v2. Note that Assigning Authority in HL7 v2 is a complex type (an HD datatype). This property contains the Universal Id Type sub-component, which is defined thusly: "The <universal ID type> (UID type) component governs the interpretation of the <universal ID> (UID) component. If the UID type is a known UID refer to HL7 Table 0301 - Universal ID type for valid values, then the UID is a universal ID of that type." - HL7 v2
6. Contains exactly one [1..1] **identifierTypeCode** with data type *String*
 - "A code corresponding to the type of identifier. In some cases, this code may be used as a qualifier to the Assigning authority component. Refer to HL7 Table 0203 - Identifier type for suggested values" - HL7 v2
7. Contains exactly one [1..1] **assigningFacNamespaceId** with data type *String*
 - The Assigning Facility is "the place or location identifier where the identifier was first assigned to the patient. This component is not an inherent part of the identifier but rather part of the history of the identifier: as part of this data type, its existence is a convenience for certain intercommunicating systems." - HL7 v2. Note that the Assigning Facility in HL7 v2 is a complex type (an HD datatype). This property contains the Namespace Id sub-component, which is "used as the HL7 identifier for the user-defined table of values for this component."
8. Contains zero or one [0..1] **assigningFacUniversalId** with data type *String*
 - The Assigning Facility is "the place or location identifier where the identifier was first assigned to the patient. This component is not an inherent part of the identifier but rather part of the history of the identifier: as part of this data type, its existence is a convenience for certain intercommunicating systems." - HL7 v2. Note that Assigning Authority in HL7 v2 is a complex type (an HD datatype). This property contains the Universal Id sub-component, which is defined thusly: "The <universal ID> (UID), is a string formatted according to the scheme defined by the <universal ID type> (UID type) component. The UID is intended to be unique over time within the UID type. It is rigorously defined. Each UID must belong to one of the specifically enumerated schemes for constructing UIDs (defined by the UID type). The UID must follow the syntactic rules of the particular universal identifier scheme (defined by the third component). Note that these syntactic rules are not defined within HL7 but are defined by the rules of the particular universal identifier scheme." - HL7 v2
9. Contains zero or one [0..1] **assigningFacUniversalIdType** with data type *String*
 - The Assigning Facility is "the place or location identifier where the identifier was first assigned to the patient. This component is not an inherent part of the identifier but rather part of the history of the identifier: as part

of this data type, its existence is a convenience for certain intercommunicating systems." - HL7 v2. Note that Assigning Authority in HL7 v2 is a complex type (an HD datatype). This property contains the Universal Id Type sub-component, which is defined thusly: "The <universal ID type> (UID type) component governs the interpretation of the <universal ID> (UID) component. If the UID type is a known UID refer to HL7 Table 0301 - Universal ID type for valid values, then the UID is a universal ID of that type." - HL7 v2

10. Contains zero or one [0..1] **expirationDate** with data type *Point In Time*

- "The last date, if known, on which the identifier is valid and active" - HL7 v2

HL72 Hierarchic Designator

This class corresponds to the HL7 V2 Hierarchic Designator (HD) data type. An HD "identifies an (administrative or system or application or other) entity that has responsibility for managing or assigning a defined set of instance identifiers (such as placer or filler number, patient identifiers, provider identifiers, etc.). This entity could be a particular health care application such as a registration system that assigns patient identifiers, a governmental entity such as a licensing authority that assigns professional identifiers or drivers license numbers, or a facility where such identifiers are assigned." - HL7 v2

1. Extends *Id*
2. Contains zero or one [0..1] **namespaceId** with data type *String*
 - This property is "used as the HL7 identifier for the user-defined table of values for this component." - HL7 v2.
3. Contains zero or one [0..1] **universalId** with data type *String*
 - "The <universal ID> (UID), is a string formatted according to the scheme defined by the <universal ID type> (UID type) component. The UID is intended to be unique over time within the UID type. It is rigorously defined. Each UID must belong to one of the specifically enumerated schemes for constructing UIDs (defined by the UID type). The UID must follow the syntactic rules of the particular universal identifier scheme (defined by the third component). Note that these syntactic rules are not defined within HL7 but are defined by the rules of the particular universal identifier scheme." - HL7 v2
4. Contains zero or one [0..1] **universalIdType** with data type *String*
 - "The <universal ID type> (UID type) component governs the interpretation of the <universal ID> (UID) component. If the UID type is a known UID refer to HL7 Table 0301 - Universal ID type for valid values, then the UID is a universal ID of that type." - HL7 v2

HL72 Organization Identifier Extended

This class corresponds to the HL7 V2 Extended Composite Name and Identification Number for Organizations (XON) data type. An XON is used "to specify the name and ID number of an organization." - HL7 v2

1. Extends *Id*
2. Contains zero or one [0..1] **identifier**
 - "Contains the sequence of characters (the code) that uniquely identifies the item being referenced by XON.1 Organization Name." - HL7 v2
3. Contains zero or one [0..1] **name** with data type *String*
 - "The name of the specified organization" - HL7 v2
4. Contains zero or one [0..1] **nameTypeCode** with data type *String*
 - "A code that represents the type of name i.e., legal name, display name. Refer to User-defined Table 0204 - Organizational Name Type for suggested values." - HL7 v2
5. Contains zero or one [0..1] **assigningAuthority**
 - "The assigning authority is a unique identifier of the system (or organization or agency or department) that creates the data" - HL7 v2

6. Contains zero or one [0..1] **identityTypeCode** with data type *String*
 - "A code corresponding to the type of identifier. In some cases, this code may be used as a qualifier to the "Assigning authority" component." - HL7 v2
7. Contains zero or one [0..1] **assigningFacility** with data type *String*
 - "The place or location identifier where the identifier was first assigned to the person. This component is not an inherent part of the identifier but rather part of the history of the identifier: as part of this data type, its existence is a convenience for certain intercommunicating systems." - HL7 v2
8. Contains zero or one [0..1] **nameRepresentation** with data type *String*
 - "Different &name/address types& and representations of the same &name/address& should be described by repeating of this field" - HL7 v2

HL72 Person Location

This class corresponds to the HL7 V2 Person Location (PL) data type. A PL "is used to specify a patient location within a healthcare institution. Which components are valued depends on the needs of the site. For example for a patient treated at home, only the person location type is valued. It is most commonly used for specifying patient locations, but may refer to other types of persons within a healthcare setting." - HL7 v2

1. Extends *Id*
2. Contains zero or one [0..1] **pointOfCare** with data type *String*
 - "Specifies the code for the point where patient care is administered. It is conditional on Person Location Type (e.g., nursing unit or department or clinic). After floor, it is the most general patient location designation." - HL7 v2
3. Contains zero or one [0..1] **room** with data type *String*
 - "Specifies the code for the patient's room. After point of care, it is the most general person location designation." - HL7 v2
4. Contains zero or one [0..1] **bed** with data type *String*
 - "Specifies the code for the patient's bed. After room, it is the most general person location designation." - HL7 v2
5. Contains zero or one [0..1] **facilityNamespaceId** with data type *String*
 - The Facility is "is subject to site interpretation but generally describes the highest level physical designation of an institution, medical center or enterprise. It is the most general person location designation." - HL7 v2. Note that the Facility in HL7 v2 is a complex type (an HD datatype). This property contains the Namespace Id sub-component, which is "used as the HL7 identifier for the user-defined table of values for this component."
6. Contains zero or one [0..1] **facilityUniversalId** with data type *String*
 - The Facility is "is subject to site interpretation but generally describes the highest level physical designation of an institution, medical center or enterprise. It is the most general person location designation." - HL7 v2. Note that the Facility in HL7 v2 is a complex type (an HD datatype). This property contains the Universal Id sub-component, which is defined thusly: "The &universal ID& (UID), is a string formatted according to the scheme defined by the &universal ID type& (UID type) component. The UID is intended to be unique over time within the UID type. It is rigorously defined. Each UID must belong to one of the specifically enumerated schemes for constructing UIDs (defined by the UID type). The UID must follow the syntactic rules of the particular universal identifier scheme (defined by the third component). Note that these syntactic rules are not defined within HL7 but are defined by the rules of the particular universal identifier scheme." - HL7 v2
7. Contains zero or one [0..1] **facilityUniversalIdType** with data type *String*
 - The Facility is "is subject to site interpretation but generally describes the highest level physical designation of an institution, medical center or enterprise. It is the most general person location designation." - HL7 v2. Note that the Facility in HL7 v2 is a complex type (an HD datatype). This property contains the Universal Id Type sub-component, which is defined thusly: "The &universal ID type& (UID type) component governs

the interpretation of the <universal ID> (UID) component. If the UID type is a known UID refer to HL7 Table 0301 - Universal ID type for valid values, then the UID is a universal ID of that type." - HL7 v2

8. Contains zero or one [0..1] **personLocationType** with data type *String*
 - "Person location type is the categorization of the person s location defined by facility, building, floor, point of care, room or bed. Although not a required field, when used, it may be the only populated field. It usually includes values such as nursing unit, department, clinic, SNF, physician s office." - HL7 v2
9. Contains zero or one [0..1] **locationDescription** with data type *String*
 - "Describes the location in free text" - HL7 v2

HL73 Coded Simple Value

"Coded data in its simplest form, where only the code is not predetermined. The code system and code system version are fixed by the context in which CS value occurs. CS is used for coded attributes that have a single HL7-defined value set." - HL7 V3

1. Extends *Code*
2. Contains exactly one [1..1] **code**
 - "The plain code symbol defined by the code system, or an expression in a syntax defined by the code system which describes the concept." - HL7 V3
3. Contains exactly one [1..1] **codeSystem**
 - "The code system that defines the code." - HL7 V3
4. Contains exactly one [1..1] **codeSystemName** with data type *String*
 - "The common name of the coding system." - HL7 V3
5. Contains zero or one [0..1] **codeSystemVersion**
 - "If applicable, a version descriptor defined specifically for the given code system." - HL7 V3

HL73 Concept Descriptor

"A reference to a concept defined in a code system" - HL7 V3

1. Extends *Code*
2. Contains exactly one [1..1] **code**
 - "The plain code symbol defined by the code system, or an expression in a syntax defined by the code system which describes the concept." - HL7 V3
3. Contains exactly one [1..1] **codeSystem**
 - "The code system that defines the code." - HL7 V3
4. Contains exactly one [1..1] **codeSystemName** with data type *String*
 - "The common name of the coding system." - HL7 V3
5. Contains zero or one [0..1] **codeSystemVersion**
 - "If applicable, a version descriptor defined specifically for the given code system." - HL7 V3
6. Contains exactly one [1..1] **valueSet** with data type *String*
 - "The value set that applied when this CD was created." - HL7 V3
7. Contains zero or one [0..1] **valueSetVersion** with data type *String*
 - "The version of the value set that applied when this CD was created" - HL7 V3
8. Contains exactly one [1..1] **displayName** with data type *String*
 - "A name, title, or representation for the code or expression as it exists in the code system identified by the value of codeSystem." - HL7 V3
9. Contains exactly one [1..1] **originalText** with data type *String*

- "The text as seen and/or selected by the user who entered the data which represents the intended meaning of the user." - HL7 V3
10. Contains zero or more [0..*] **codingRationale** with data type *String*
 - "The reason a particular CD has been provided. " - HL7 V3
 11. Contains zero or more [0..*] **translation** with data type *HL73 Concept Descriptor*
 - "A set of other CDs that each represent a translation of this CD into equivalent codes within the same code system or into corresponding concepts from other code systems." - HL7 V3
 12. Contains zero or one [0..1] **source** with data type *HL73 Concept Descriptor*
 - "The CD from which this CD was translated, if it was translated from another CD." - HL7 V3

HL73 Id

This class corresponds to the HL7 V3 II datatype, which is defined as: "An identifier that uniquely identifies a thing or object. Examples are object identifier for HL7 RIM objects, medical record number, order id, service catalog item id, Vehicle Identification Number (VIN), etc. Note that a II does not identify the type of the object being identified, or the type of the association between the object and the identifier - it is only the identifier itself."

1. Extends *Id*
2. Contains exactly one [1..1] **root** with data type *String*
 - A unique identifier that guarantees the global uniqueness of the instance identifier. The root alone may be the entire instance identifier. This string should be one of the following RUID, OID or UUID
 - "A unique identifier that guarantees the global uniqueness of the instance identifier. The root alone may be the entire instance identifier" - HL7 V3
3. Contains zero or one [0..1] **extension** with data type *String*
 - A character string as a unique identifier within the scope of the identifier root.
 - "A character string as a unique identifier within the scope of the identifier root." - HL7 V3
4. Contains zero or one [0..1] **identifierName** with data type *String*
 - A human readable name or mnemonic for the assigning authority. The Assigning Authority Name has no computational value. The purpose of a Assigning Authority Name is to assist an unaided human interpreter of an II value to interpret the authority. Note: no automated processing must depend on the assigning authority name to be present in any form.
 - "This is a human-readable name for the namespace represented in the root. It is a descriptive name for the actual namespace. e.g. "California, U.S. Driver's License Number, 1970-". " - HL7 V3

Id

An identifier that uniquely identifies a thing or object. This *abstract* datatype represents a "identifier" - some series of letters or numbers which uniquely identifies something. In practice, this datatype will be substituted with specific "flavors" of the Id datatype for use in particular standards-based payloads.

1. Contains zero or one [0..1] **identifier** with data type *String*
 - This is a placeholder for the "entity identifier," which is usually defined to be unique within the series of identifiers created by the "assigning authority". Note that this property will be replaced by some other concept or group of concepts when this abstract class is replaced by a specific class for a given target platform.
2. Contains zero or one [0..1] **idType** with data type *String*
 - Indicates what kind of identifier is being represented. For example, the idType might be represent a Driver's Licence, in which case the assigning Authority might be the Virginia Department of Motor Vehicles. Note that: a) this information can (and probably should) be also be made clear by the name and definition of the property whose datatype is an Id; and b) this property might be an enumeration in the future.

3. Contains zero or one [0..1] **assigningAuthority** with data type *String*

- This is a placeholder for the "assigning authority", which is a unique identifier of the system (or organization or agency or department) that creates the data. Note that this property will be replaced by some other concept or group of concepts when this abstract class is replaced by a specific class for a given target platform.

Integer Interval

An interval of integer numbers stating the minimal and maximal number of repetitions of the Act.

This data type represents an Interval, where the Low and High Limits are Integers. An Interval is a "set of consecutive values of an ordered base data type." - HL7 V3 "Integer numbers are precise numbers that are results of counting and enumerating. Integer numbers are discrete, the set of integers is infinite but countable. No arbitrary limit is imposed on the range of integer numbers." - HL7 V3

1. Contains exactly one [1..1] **low** with data type *Integer*

- The minimal number of repetitions of the Act.
- "The low limit of the interval. The low limit SHALL NOT be positive infinity." - HL7 V3

2. Contains zero or one [0..1] **high** with data type *Integer*

- The maximal number of repetitions of the Act.
- "The high limit of the interval. The high limit SHALL NOT be negative infinity, and SHALL be higher than the low limit if one exists." - HL7 V3

Integer Ratio

A ratio (numerator : denominator) specifying the relative quantities of the Entity playing the Role in the Entity scoping the Role, used for Roles that represent composition relationships between the scoping and playing Entities.

This data type represents an Ratio, where the Numerator and Denominator are Integers. A Ratio is "a quantity constructed as the quotient of a numerator quantity divided by a denominator quantity. Common factors in the numerator and denominator are not automatically cancelled out. The RTO data type supports titers (e.g., "1:128") and other quantities produced by laboratories that truly represent ratios." - HL7 V3 "Integer numbers are precise numbers that are results of counting and enumerating. Integer numbers are discrete, the set of integers is infinite but countable. No arbitrary limit is imposed on the range of integer numbers." - HL7 V3

1. Contains exactly one [1..1] **numerator** with data type *Integer*

- The quantity that is being divided in the ratio. The default is the integer number 1
- "The quantity that is being divided in the ratio. The default is the integer number 1 (one)." - HL7 V3

2. Contains exactly one [1..1] **denominator** with data type *Integer*

- The quantity that divides the numerator in the ratio. The default is the integer number 1 (one.) The denominator must not be zero.
- "The quantity that divides the numerator in the ratio. The default is the integer number 1 (one). The denominator SHALL not be zero." - HL7 V3

Monetary Amount

Indicates the monetary amount to be transferred from the debit to the credit account.

"A quantity expressing an amount of money in some currency. While the monetary amount is a single kind of quantity (money) the exchange rates between the different units are variable. This is the principle difference between PQ and MO, and the reason why currency units are not physical units." - HL7 V3

1. Contains exactly one [1..1] **currency** with data type *Code*

- Currencies are the units in which monetary amounts are denominated in different economic regions.
 - "The currency unit as defined in ISO 4217" - HL7 V3
2. Contains exactly one [1..1] **value** with data type *Real*
 - The amount of money in some currency.
 - "The magnitude of the Monetary Amount in terms of currency" - HL7 V3

Non Us Mailing Address

A specialization of Address for non-US postal addresses. Note that the Address class does not have a State or Province attribute - the US Mailing Address includes a code for the State, whereas the Non-US Mailing address merely has a string for the "jurisdiction", which would have the state or province as needed by that country's postal designation conventions.

1. Extends *Address*
2. Contains exactly one [1..1] **jurisdiction** with data type *String*
 - This property is a string for the "jurisdiction", which would have the state or province as needed by the country's postal designation conventions. In some cases, this property will be blank, and the jurisdiction would be part of one of the street address lines.

Nullable Boolean

This datatype represents a boolean value which may be unknown. In other words, this datatype can represent Yes, No, and Unknown; unlike a traditional boolean which can only represent True and False. For this logical model, this class has been modeled in the HL7 V3 style, which has the advantage of being able to account for different kinds of "unknowns", but has the disadvantage of having two fields, one of which must always be populated while the other must always be empty. Implementers may choose to implement this class using a single field instead.

1. Contains exactly one [1..1] **value** with data type *Boolean*
 - This property contains either a True or a False (or Yes or No). If "unknown" is needed, this property will be empty, and the nullFlavor property will be valued.
2. Contains exactly one [1..1] **nullFlavor** with data type *Null Flavor*
 - This property indicates why the value property is empty. It uses the NullFlavor enumeration. This property must be valued when the value property is empty, and must be empty when the value property is valued.

Null Flavor

Value Set	NullFlavor		
Description	This enumeration defines the set of possible "null-flavors", which are used to communicate why a datatype value is empty when otherwise it should not be empty. Null Flavor: "If a value is an exceptional value (NULL-value), this specifies in what way and why proper information is missing." - HL7 V3		
Concept Code	Concept Name	Code System	Description
AskedButUnknown			"Information was sought but not found (e.g., patient was asked but didn't know)" - HL7 V3
Derived			"An actual value may exist, but it must be derived from the provided information (usually an EXPR generic data type

Concept Code	Concept Name	Code System	Description
Invalid			extension will be used to convey the derivation expression" - HL7 V3 "The value as represented in the instance is not a member of the set of permitted data values in the constrained value domain of a variable." - HL7 V3
Masked			"There is information on this item available but it has not been provided by the sender due to security, privacy or other reasons. There may be an alternate mechanism for gaining access to this information." - HL7 V3
NegativeInfinity			"Negative infinity of numbers." - HL7 V3
NoInformation			"The value is exceptional (missing, omitted, incomplete, improper). No information as to the reason for being an exceptional value is provided. This is the most general exceptional value. It is also the default exceptional value." - HL7 V3
NotApplicable			"Known to have no proper value (e.g., last menstrual period for a male)." - HL7 V3
NotAsked			"This information has not been sought (e.g., patient was not asked)" - HL7 V3
Other			"The actual value is not a member of the set of permitted data values in the constrained value domain of a variable. (e.g., concept not provided by required code system)." - HL7 V3
PositiveInfinity			"Positive infinity of numbers." - HL7 V3
SufficientQuantity			"The specific quantity is not known, but is known to be non-zero and is not specified because it makes up the bulk of the material. e.g. 'Add 10mg of ingredient X, 50mg of ingredient Y, and sufficient quantity of water to 100mL.' The null flavor would be used to express the quantity of water." - HL7 V3
TemporarilyUnavailable			"Information is not available at this time but it is expected that it will be available later." - HL7 V3
Trace			"The content is greater than zero, but too small to be quantified." - HL7 V3
Unencoded			"The actual value has not yet been encoded within the approved valueset for the domain." - HL7 V3
Unknown			"A proper value is applicable, but not known." - HL7 V3

Organization Name

"A word or term by which a person or thing is commonly and distinctively known." - Collins English Dictionary. This datatype represents a organization's name.

1. Contains exactly one [1..1] **name** with data type *String*
 - "A word or term by which a person or thing is commonly and distinctively known." - Collins English Dictionary. This property represents a organization's name. Note that the datatype is simply a string, unlike HL7 V3's complex collection of name parts.

Person Name

The name of the person. Uses the VHIM-constrained Person Name data type.

"A word or term by which a person or thing is commonly and distinctively known." - Collins English Dictionary. This datatype represents a person's name.

1. Contains zero or more [0..*] **prefix** with data type *String*
 - 'A prefix has a strong association to the immediately following name part. A prefix has no implicit trailing white space (it has implicit leading white space though). Note that prefixes can be inverted' (HL7) A Person Name Prefix is usually an academic or nobility title. An Academic title includes a prefix like 'Dr.' There are still people with nobility titles (aristocrats). German 'von' is generally a nobility title, not a mere voorvoegsel. Others are 'Earl of' or 'His Majesty King of...' etc. Rarely used nowadays, but some systems do keep track of this.
 - "Contains a set of honorific terms that typically appear before a person's name, for example Mr., Mrs., Dr., etc. Prefixes have a strong association to the immediately following name part." - HL7 V3
2. Contains zero or more [0..*] **given** with data type *String*
 - 'Given name (don't call it 'first name' since this given names do not always come first)' (HL7)
 - "A set of names given to a person at birth, but not including the family name. In western cultures, this property would contain the 'first' and 'middle' names. Note that in some cultures, the given name is placed after the family name. Note also that this property contains multiple elements, so it can handle those situations where a person has more than one 'middle' name." - HL7 V3
3. Contains zero or one [0..1] **nickname** with data type *String*
 - 'A callme name is (usually a given name) that is preferred when a person is directly addressed.' (HL7)
 - A name added to or substituted for the proper name of a person.
4. Contains exactly one [1..1] **family** with data type *String*
 - 'Family name, this is the name that links to the genealogy' (HL7)
 - "The portion of a person's name that reflects the genealogy of the person. In western cultures, this is the 'last' name. In eastern cultures, the family name appears before the person's given name(s). In some cultures (e.g. Eritrea) the family name of a son is the first name of his father." - HL7 V3
5. Contains zero or more [0..*] **suffix** with data type *String*
 - 'A suffix has a strong association to the immediately preceding name part. A prefix has no implicit leading white space (it has implicit trailing white space though). Suffices can not be inverted' (HL7)
 - "Contains a list of honorific terms that typically appear after a person's name, for example Jr., Sr., MD, RN, etc. Prefixes have a strong association to the immediately following name part." - HL7 V3
6. Contains zero or one [0..1] **initials** with data type *String*
 - The first letter of each part of a proper name. This property stores the initials of a person's name, which is used in research studies to identify the patient without divulging the person's entire name.

Physical Quantity

The amount that was or is to be supplied

"A dimensioned quantity expressing the result of measuring" - HL7 V3

1. Contains exactly one [1..1] **value** with data type *Real*
 - The magnitude of the quantity measured in terms of the unit.
 - "The magnitude of the quantity measured in terms of the unit" - HL7 V3
2. Contains exactly one [1..1] **unit** with data type *Code*
 - The unit of measure specified in the Unified Code for Units of Measure (UCUM) [].
 - "The unit of measure specified in the Unified Code for Units of Measure (UCUM). The default unit is 1." - HL7 V3

Physical Quantity Interval

The amount of the therapeutic agent or other substance given at one administration event.

This data type represents an Interval, where the Low and High Limits are Physical Quantities. An Interval is a "set of consecutive values of an ordered base data type." - HL7 V3 A Physical Quantity is "a dimensioned quantity expressing the result of measuring" - HL7 V3

1. Contains zero or one [0..1] **lowValue** with data type *Real*
 - The magnitude of the quantity measured in terms of the unit.
 - "The low limit of the interval. The low limit SHALL NOT be positive infinity." - HL7 V3 This property contains the numeric portion (i.e., not including any units of measure) of the low limit.
2. Contains zero or one [0..1] **lowUnit** with data type *Code*
 - The unit of measure specified in the Unified Code for Units of Measure (UCUM) [].
 - "The low limit of the interval. The low limit SHALL NOT be positive infinity." - HL7 V3. This property contains the Unit of Measure of the low limit.
3. Contains zero or one [0..1] **highValue** with data type *Real*
 - The magnitude of the quantity measured in terms of the unit.
 - "The high limit of the interval. The high limit SHALL NOT be negative infinity, and SHALL be higher than the low limit if one exists." - HL7 V3. This property contains the numeric portion (i.e., not including any units of measure) of the high limit.
4. Contains zero or one [0..1] **highUnit** with data type *Code*
 - The unit of measure specified in the Unified Code for Units of Measure (UCUM) [].
 - "The high limit of the interval. The high limit SHALL NOT be negative infinity, and SHALL be higher than the low limit if one exists." - HL7 V3. This property contains the Unit of Measure of the high limit.

Physical Quantity Ratio

A ratio (numerator : denominator) specifying the relative quantities of the Entity playing the Role in the Entity scoping the Role, used for Roles that represent composition relationships between the scoping and playing Entities.

This data type represents a Ratio, where the Low and High Limits are Physical Quantities. A Ratio is "A quantity constructed as the quotient of a numerator quantity divided by a denominator quantity." - HL7 V3 A Physical Quantity is "a dimensioned quantity expressing the result of measuring" - HL7 V3

1. Contains exactly one [1..1] **numerator** with data type *Physical Quantity*

- The quantity that is being divided in the ratio. The default is the integer number 1
 - "The quantity that is being divided in the ratio. The default is the integer number 1 (one)." - HL7 V3
2. Contains exactly one [1..1] **denominator** with data type *Physical Quantity*
- The quantity that divides the numerator in the ratio. The default is the integer number 1 (one.) The denominator must not be zero.
 - "The quantity that divides the numerator in the ratio. The default is the integer number 1 (one). The denominator SHALL not be zero." - HL7 V3

Point In Time

A quantity specifying a point on the axis of natural time. A point in time is most often represented as a calendar expression. Semantically, however, time is independent from calendars and best described by its relationship to elapsed time (measured as a physical quantity in the dimension of time.) A point in time plus an elapsed time yields another point in time. Inversely, a point in time minus another point in time yields an elapsed time. As nobody knows when time began, a point in time is conceptualized as the amount of time that has elapsed from some arbitrary zero-point, called an epoch. Because there is no absolute zero-point on the time axis natural time is a difference-scale quantity, where only differences are defined but no ratios. (For example, no point in time is - absolutely speaking - 'twice as late' as another point in time.) Given some arbitrary zero-point, one can express any point in time as an elapsed time measured from that offset. Such an arbitrary zero-point is called an epoch. This epoch-offset form is used as a semantic representation here, without implying that any system would have to implement the TS data type in that way. Systems that do not need to compute distances between points in time will not need any other representation than a calendar expression literal

A datatype containing date/time information. This datatype is a placeholder, as various platforms have differing built-in date/time datatypes. It is anticipated that this datatype will be replaced by a different datatype when transforming to a particular implementation platform.

1. Contains exactly one [1..1] **literal** with data type *String*
- For the default Gregorian calendar the calendar expression literals of this specification conform to the constrained ISO 8601 that is defined in ISO 8824 (ASN.1) under clause 32 (generalized time) and to the HL7 version 2 TS data format.
 - "TS literals are simple calendar expressions... [which] conform to the constrained ISO 8601... western calendar expressions begin with the 4-digit year; followed by the 2-digit month of the year; followed by the 2-digit day of the month; followed by the 2-digit hour of the day (beginning with zero); and so forth. For example, '200004010315' is a valid expression for April 1, 2000, 3:15 am. A calendar expression can be of variable precision, omitting parts from the right. For example, '20000401' is precise only to the day of the month. The least defined calendar period (i.e. the second) may be written as a REAL, with the number of integer digits specified, followed by the decimal point and any number of fractional digits. For example, '20000401031520.34' means April 1, 2000, 3:15 and 20.34 seconds. When other calendars are used in the future, a prefix 'GREG:' can be placed before the western (Gregorian) calendar expression to disambiguate from other calendars. Each calendar shall have its own prefix. However, the western calendar is the default if no prefix is present. In the modern Gregorian calendar (and all calendars where time of day is based on UTC), the calendar expression may contain a time zone suffix. The time zone suffix begins with a plus (+) or minus (-) followed by digits for the hour and, for non UTC times, minute cycles. UTC is designated as offset '+00' or '-00'; the ISO 8601 and ISO 8824 suffix 'Z' for UTC is not permitted." - HL7 V3

Rate Quantity

Identifies the speed with which the substance is introduced into the subject. Expressed as a physical (extensive) quantity over elapsed time (e.g., examples are 100 mL/h, 1 g/d, 40 mmol/h, etc.)

Rate: "A certain quantity or amount of one thing considered in relation to a unit of another thing and used as a standard or measure: at the rate of 60 miles an hour." - Dictionary.com This class represents a ratio of some Physical

Quantity over a period of time. This class is structurally similar to other Ratio classes (e.g., *IntegerRatio*), except the denominator is by definition a *TimeQuantity*.

1. Contains exactly one [1..1] **numerator** with data type *Physical Quantity*
 - The quantity that is being divided in the ratio. The default is the integer number 1
 - "The quantity that is being divided in the ratio. The default is the integer number 1 (one)." - HL7 V3
2. Contains exactly one [1..1] **denominator** with data type *Time Quantity*
 - The quantity that divides the numerator in the ratio. The default is the integer number 1 (one.) The denominator must not be zero.
 - "The quantity that divides the numerator in the ratio. The denominator SHALL not be zero." - HL7 V3 This property is by definition a period of time, with the unit almost always one. For example, one hour, one minute, etc.

Real

"A scalar magnitude. Typically used whenever quantities are measured, estimated, or computed from other real numbers. The typical representation is decimal, where the number of significant decimal digits is known as the precision." - HL7 V3 A datatype containing non-whole numbers. This datatype is a placeholder, as various platforms have differing built-in floating-point datatypes. It is anticipated that this datatype will be replaced by a different datatype when transforming to a particular implementation platform.

1. Contains exactly one [1..1] **literal** with data type *String*
 - This is a placeholder for an actual datatype that will be substituted via transformation to a platform-specific datatype.

Telecommunications

A collection of electronic addresses at which the person or organization may be reached. This includes telephones, email addresses, etc.

1. Contains exactly one [1..1] **addressType** with data type *Address Type*
 - Indicates the kind of communications address that is contained within this class. Examples include primaryHome, Work, etc. Note that in HL7 V3, this concept is part of the Telecom datatype (the 'use code'). This concept is made explicit in this Telecommunications class, because this is a platform-independent model. Non V3 implementations will need other mechanisms to deal with the type.
- 2.
3. Contains exactly one [1..1] **universalResourceId** with data type *String*
 - Represents a telecommunications address at which the person or organization may be reached. Note that this property is a simply a string, the formatting of which will depend on the type of communications address employed.

Time Interval

An interval of time specified as an interval of points in time - TS.

This data type represents an Interval, where the Low and High Limits are Points In Time. An Interval is a "set of consecutive values of an ordered base data type." - HL7 V3 A PointInTime is "a quantity specifying a point on the axis of natural time. A point in time is most often represented as a calendar expression." - HL7 V3

1. Contains zero or one [0..1] **low** with data type *Point In Time*
 - This is the low limit of the interval.

- "The low limit of the interval. The low limit SHALL NOT be positive infinity." - HL7 V3
2. Contains zero or one [0..1] **high** with data type *Point In Time*
 - This is the high limit of the interval.
 - "The high limit of the interval. The high limit SHALL NOT be negative infinity, and SHALL be higher than the low limit if one exists." - HL7 V3
 3. Contains zero or one [0..1] **width** with data type *Time Quantity*
 - The difference between high and low boundary. The purpose of distinguishing a width property is to handle all cases of incomplete information symmetrically. In any interval representation only two of the three properties high, low, and width need to be stated and the third can be derived.
 - "The difference between HIGH and LOW boundary. The purpose of distinguishing width is to handle all cases of incomplete information symmetrically. In any IVL representation only two of the three properties HIGH, LOW, and width need to be stated: the third can be derived. When both boundaries are known, width can be derived as HIGH minus LOW. When one boundary and width is known, the other boundary is also known. When no boundary is known, width may still be known. For example, one knows that an activity takes about 30 minutes, but one may not yet know when that activity is started." - HL7 V3

Time Quantity

A length of time specified as a Physical Quantity, e.g., 5 minutes, 2.5 hours.

This class represents an amount of time, such as 10 minutes, 3.2 seconds, etc. This class is structurally identical to a Physical Quantity, although the units of measure are by definition time units (e.g., hours), rather than physical units (e.g., kilograms).

1. Contains exactly one [1..1] **value** with data type *Real*
 - Value of the number of time units
 - "The magnitude of the quantity measured in terms of the unit" - HL7 V3
2. Contains exactly one [1..1] **unit** with data type *Code*
 - The unit of measure specified in the Unified Code for Units of Measure (UCUM).
 - "The unit of measure specified in the Unified Code for Units of Measure (UCUM)." - HL7 V3 The units of measure used by this class are limited to those measuring time (e.g., minutes, hours, years, etc.)

Us Mailing Address

A specialization of MailingAddress that is used for U.S. addresses. Note that the state property may only contain a code for a U.S. State, territory, or APO.

1. Extends *Address*
2. Contains exactly one [1..1] **state** with data type *Code*
 - An Address Part (ADXP) that contains the state or province. A state or provinces is a sub-unit of a country with limited sovereignty in a federally organized country.

VA Code

This class represents a codified concept as used within the VA Enterprise Terminology Systems (VETS). This class is anticipated to only be used by VA clients. It contains VA Unique concept Identifiers (VIDs) for the concept, the surface form (or designation), and the Value Set of which the concept is a proper member. It also contains a display text, which is the surface form of the Designation VUID, which while technically not necessary, is useful to store in the class in order to avoid a call to a terminology service. In addition, this class contains four properties to contain alternate code information, primarily for use when an application requires a code from a particular coding system

(e.g., a billing system needs a CPT code). Finally, this class contains an alternate Reason property to indicate why the alternate fields are populated (e.g., perhaps because of a mediation failure).

1. Extends *Code*
2. Contains zero or one [0..1] **conceptVuid** with data type *String*
 - Identifies the concept being conveyed in this class, regardless of the surface form or designation used to describe the concept.
3. Contains zero or one [0..1] **designationVuid** with data type *String*
 - Identifies the surface form or designation used to describe the concept.
4. Contains zero or one [0..1] **valueSetVuid** with data type *String*
 - Identifies a set of concepts to which this concept belongs. The Value Set may be "States of the United States", of which the concepts of "Virginia" and "Utah" are members, but "Ottawa" is not. The value set void is used to define a priori a set of possible concepts that may be used "at run time".
5. Contains zero or one [0..1] **displayText** with data type *String*
 - Contains the text that corresponds to the designation void. Technically this is not necessary, but may be used in order to avoid a call to a terminology service.
6. Contains zero or one [0..1] **alternateCode** with data type *String*
 - At times, it may be useful or necessary to include an alternate representation of the concept being conveyed using another code system. These "alternate" properties contain the data necessary to identify the coding system, code, etc. that correspond to the concept. The alternate code contains the code in the alternate coding system that corresponds to the concept void.
7. Contains zero or one [0..1] **alternateCodeSystem** with data type *String*
 - At times, it may be useful or necessary to include an alternate representation of the concept being conveyed using another code system. These "alternate" properties contain the data necessary to identify the coding system, code, etc. that correspond to the concept. The alternate coding system identifies the alternate coding system, the code from which corresponds to the concept void.
8. Contains zero or one [0..1] **alternateCodeSystemVersion** with data type *String*
 - At times, it may be useful or necessary to include an alternate representation of the concept being conveyed using another code system. These "alternate" properties contain the data necessary to identify the coding system, code, etc. that correspond to the concept. The alternate coding system version identifies version of alternate coding system, if the alternate coding system is versioned. Otherwise, this property is empty.
9. Contains zero or one [0..1] **alternateDisplayText** with data type *String*
 - At times, it may be useful or necessary to include an alternate representation of the concept being conveyed using another code system. These "alternate" properties contain the data necessary to identify the coding system, code, etc. that correspond to the concept. The alternate designation text contains the textual value of the code in the alternate coding system that corresponds to the concept void. This text should ideally correspond with the designation void, and with the designation text.
10. Contains zero or one [0..1] **alternateReason** with data type *String*
 - At times, it may be useful or necessary to include an alternate representation of the concept being conveyed using another code system. These "alternate" properties contain the data necessary to identify the coding system, code, etc. that correspond to the concept. This property provides an explanation of why the alternate properties are populated. Possible values include: Backwards compatibility with systems requiring a particular coding scheme (e.g., billing systems require CPT and ICD codes), Mediation failure on incoming data from other systems, etc.

REFERENCES

- CDISC Submission Data Standards Team. Study Data Tabulation Model Implementation Guide: Human Clinical Trials Available at: www.CDISC.org
- Dolin RH, Alschuler L, Boyer S, Beebe C, Behlen FM, Biron PV, Shabo A, (Editors). HL7 Clinical Document Architecture, Release 2.0. ANSI-approved HL7 Standard; May 2005. Ann Arbor, Mich.: Health Level Seven, Inc. Available through [HL7](#) or if an HL7 member with the following link: [CDA Release 2 Normative Web Edition](#).
- [LOINC®](#) : Logical Observation Identifiers Names and Codes, Regenstrief Institute.
- Extensible Markup Language, www.w3.org/XML .
- Dolin RH, Alschuler L, Boyer S, Beebe C, Behlen FM, Biron PV, Shabo A., HL7 Clinical Document Architecture, Release 2. J Am Med Inform Assoc. 2006;13:30-39. Available at: <http://www.jamia.org/cgi/reprint/13/1/30> .

