DEUTSCHE NORM

Juni 2011

DIN EN ISO 21090

DIN

ICS 35.240.80

Medizinische Informatik – Harmonisierte Datentypen für den Datenaustausch im Gesundheitswesen (ISO 21090:2011); Englische Fassung EN ISO 21090:2011, nur auf CD-ROM

Health Informatics –
Harmonized data types for information interchange (ISO 21090:2011);
English version EN ISO 21090:2011, only on CD-ROM

Informatique de santé –
Types de données harmonisées pour une interchangeabilité d'informations (ISO 21090:2011);
Version anglaise EN ISO 21090:2011, seulement en CD-ROM

Gesamtumfang 208 Seiten

Normenausschuss Medizin (NAMed) im DIN

AVA Documents
Document Sharing Hub

Full document details: DIN EN ISO 21090:2011-06

Nationales Vorwort

Diese Norm enthält unter Berücksichtigung des Präsidialbeschlusses 13/1983 den englischen Originaltext der Internationalen Norm EN ISO 21090:2011. Diese Internationale Norm wurde von der WG 2 „Messaging and Communication“ des Technischen Komitees ISO/TC 215 „Health informatics“ übernommen und im Rahmen eines gemeinsamen Projektes mit dem Technischen Komitee CEN/TC 251 „Medizinische Informatik“ (Sekretariat: NEN, Niederlande) erarbeitet. Der Arbeitsausschuss NA 063-07-02 AA „Interoperabilität“ des Fachbereichs 7 „Medizinische Informatik“ des Normenausschusses Medizin (NAMed) im DIN hat an der Erarbeitung mitgewirkt.

Für die in diesem Dokument zitierten Internationalen Normen wird im Folgenden auf die entsprechenden Deutschen Normen hingewiesen:

ISO 8601 siehe DIN ISO 8601

Zusammenfassung

Diese Internationale Norm beschreibt einen Satz von Datentypen, der im Gesundheitswesen üblich ist, um den Informationsaustausch im Gesundheitswesen zu unterstützen. Die Semantik beruht auf der ISO 11404. Es wird auch eine UML und eine XML Darstellung dieser Datentypen beschrieben.

Die Arbeiten berücksichtigen insbesondere die Arbeiten von HL7, Version 3, ISO 11404 und EN 13606.

Begriffe

Begriff	Definition	Term	Definition
Attribut	Merkmal eines Objektes, dem ein Name und ein Typ zugewiesen wurde ANMERKUNG Der Wert eines Attributes kann sich innerhalb der Lebensdauer eines Objektes ändern	attribute	characteristic of an object that is assigned a name and a type NOTE The value of an attribute can change during the lifetime of the object.
Klasse	Beschreibung für einen Objektdatensatz mit ähnlicher Struktur, Verhalten und Beziehungen	class	descriptor for a set of objects with similar structure, behaviour and relationships
Kode	Begriffsdarstellung veröffentlicht vom Autor eines Kodesystems als Teil des Kodesystems, als Entität des Kodesystems	code	concept representation published by the author of a code system as part of the code system, being an entity of that code system

Documents
Document Sharing Hub

Full document details: DIN EN ISO 21090:2011-06

Begriff	Definition	Term	Definition
Kodesystem	kontrollierte Sammlung von Begriffsidentifikatoren, üblicherweise Kodes, aber manchmal auch komplexere Sätze von Regeln und Empfehlungen ANMERKUNG Sie werden oft als Sammlung von eindeutig, identifizierbaren Begriffen mit dazugehörigen Darstellungen, Benennungen, Assoziationen und Bedeutungen beschrieben. BEISPIELE ICD-9, LOINC und SNOMED	code system	managed collection of concept identifiers, usually codes, but sometimes more complex sets of rules and references NOTE They are often described as collections of uniquely identifiable concepts with associated representations, designations, associations and meanings. EXAMPLES ICD-9, LOINC and SNOMED
Begriff	eindeutige mentale Darstellung einer realen oder abstrakten Sache, eine atomare Gedankeneinheit ANMERKUNG 1 Er sollte eindeutig in einem Kodesystem sein. ANMERKUNG 2 Ein Begriff kann synonyme Bezeichnungen der Darstellung haben und kann eine primitive oder zusammengesetzte Bezeichnung sein.	concept	unitary mental representation of a real or abstract thing; an atomic unit of thought NOTE 1 It should be unique in a given code system. NOTE 2 A concept can have synonyms in terms of representation and it can be a primitive or compositional term.
Konformität	Erfüllung festgelegter Anforderungen; Einhaltung der Anforderungen einer informationsverarbeitenden Einheit von einer oder mehreren spezifischen Spezifikationen oder Normen	conformance	fulfillment of a specified requirement; adherence of an information processing entity to the requirements of one or more specific specifications or standards
Datentyp	Menge von eindeutigen Werten, charakterisiert durch die Eigenschaften dieser Werte und die Verwendung von diesen Werten	datatype	set of distinct values, characterized by properties of those values, and by operations on those values
Aufzählung	Datentypen, die aus einem Satz von nutzerspezifisch benannten aufgezählten Buchstabensymbolen bestehen ANMERKUNG Die Buchstabensymbole haben eine relative Ordnung, aber es ist keine Algebra für sie festgelegt	enumeration	datatype whose instances are a set of user-specified named enumeration literals NOTE The literals have a relative order, but no algebra is defined on them.

AVA Documents
Document Sharing Hub

Full document details: DIN EN ISO 21090:2011-06

Begriff	Definition	Term	Definition
Verallgemeinerung	taxonimische Beziehung zwischen einer mehr allgemeinen Klasse, Schnittstelle oder Begriffs zu einer mehr spezifischen Klasse, Schnittstelle oder Begriff ANMERKUNG 1 Jede Instanz eines spezifischen Elements ist auch eine Instanz eines allgemeinen Elements. Dadurch hat ein spezifisches Element auch alle Funktionen eines mehr allgemeinen Elements ANMERKUNG 2 Das mehr spezifische Element ist vollständig konsistent mit dem mehr allgemeinen Element und beinhaltet zusätzliche Informationen ANMERKUNG 3 Eine Instanz eines mehr spezifischen Elements kann verwendet werden, wo ein mehr allgemeines Element erlaubt ist	generalization	taxonomic relationship between a more general class, interface or concept and a more specific class, interface or concept NOTE 1 Each instance of the specific element is also an instance of the general element. Thus, the specific element has all the features of the more general element. NOTE 2 The more specific element is fully consistent with the more general element and contains additional information. NOTE 3 An instance of the more specific element can be used where the more general element is allowed.
Informationsverarbeitende Entität	alles was Informationen verarbeitet und den Begriff der Datentypen beinhaltet, inklusive anderer Norme, Spezifikationen, datenverarbeitenden Einrichtungen und Diensten	information processing entity	anything that processes information and contains the concept of datatype, including other standards, specifications, data handling facilities and services
Vererbung	Mechanismus durch den mehr spezifische Elemente Strukturen und Verhaltensweisen von mehr allgemeinen Elementen enthalten	inheritance	mechanism by which more specific elements incorporate structure and behaviour of more general elements
Schnittstelle	Festlegung für die extern sichtbare Verwendung der Klassen, ohne Festlegung der internen Struktur	interface	specifier for the externally-visible operations of class, without specification of internal structure
Invarianz	Rolle, um die Funktionen einer Klasse festzulegen, die immer richtig sein müssen	invariant	rule about the features of a class which must always be true

Documents
Document Sharing Hub

Full document details: DIN EN ISO 21090:2011-06

Begriff	Definition	Term	Definition
Verwendung	Dienst, für den die Anforderungen besteht, dass die Instanz einer Klasse diesen gegebenenfalls leisten muss ANMERKUNG Eine Verwendung hat einen Namen und eine Liste von Argumenten mit zugeteilten Namen und Typen und antwortet mit einem Wert des spezifischen Typs.	operation	service that an instance of the class may be requested to perform NOTE An operation has a name and a list of arguments with assigned names and types, and returns a value of the type specified.
Spezifizierung	taxonimische Beziehung zwischen einer mehr allgemeinen Klasse, Schnittstelle oder Begriffs zu einer mehr spezifischen Klasse, Schnittstelle oder Begriff, wobei neue Funktionen oder durch vorgegebene Verhaltensweisen neu festgelegte existierende Funktionen zugefügt werden	specialization	taxonomic relationship between a more general class, interface or concept and a more specific class, interface or concept where the more specific entity adds new features or redefines existing features by constraining their possible behaviours
String Zeichensatz	Zeichensatz, der für „string“ Inhalte durchgehend in dieser Internationalen Norm verwendet wird	string character set	character set used in all string content throughout this International Standard
Satz von Werten	das, was einen eindeutig identifizierbaren Satz von validen Begriffsdarstellungen abbildet, wo jede Begriffsdarstellung daraufhin untersucht werden kann, ob sie ein Teil des Satzes von Werten ist oder nicht ANMERKUNG Eine Begriffsdarstellung kann ein einzelner Begriffskode oder eine nachträglich abgestimmte Kombination von Kodes sein.	valueSet	that which represents a uniquely identifiable set of valid concept representations, where any concept representation can be tested to determine whether or not it is a member of the value set NOTE A concept representation can be a single concept code or a post-coordinated combination of codes.

AVA Documents
Document Sharing Hub

Full document details: DIN EN ISO 21090:2011-06

Nationaler Anhang NA
(informativ)

Literaturhinweise

DIN ISO 8601, *Datenelemente und Austauschformate — Informationsaustausch — Darstellung von Datum und Uhrzeit*

Documents
Document Sharing Hub

Full document details: DIN EN ISO 21090:2011-06

EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

EN ISO 21090

February 2011

ICS 35.240.80

English Version

Health Informatics — Harmonized data types for information interchange (ISO 21090:2011)

Informatique de santé —
Types de données harmonisées pour une interchangeabilité d'informations
(ISO 21090:2011)

Medizinische Informatik —
Harmonisierte Datentypen für den Datenaustausch im Gesundheitswesen
(ISO 21090:2011)

This European Standard was approved by CEN on 12 February 2011.

CEN members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration. Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CEN member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and United Kingdom.

EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

Management Centre: Avenue Marnix 17, B-1000 Brussels

© 2011 CEN All rights of exploitation in any form and by any means reserved worldwide for CEN national Members.

Ref. No. EN ISO 21090:2011: E

Documents
Document Sharing Hub

Full document details: DIN EN ISO 21090:2011-06

Contents

Page

Foreword ... 3
Introduction ... 4
1 Scope ... 5
2 Normative references ... 5
3 Terms and definitions ... 6
4 Abbreviated terms ... 8
5 Conformance ... 8
5.1 General ... 8
5.2 Direct conformance ... 9
5.3 Indirect conformance ... 11
6 Datatypes overview ... 12
6.1 What is a datatype? ... 12
6.2 Definitions of datatypes ... 13
6.3 Datatype names and re-use of common datatype names ... 13
6.4 Mapping to this datatypes International Standard ... 14
6.5 Conformance with ISO/IEC 11404 ... 14
6.6 Reference to UML 2 ... 15
6.7 Modelling of datatypes ... 15
7 Datatypes ... 19
7.1 General properties ... 19
7.2 Top level model ... 24
7.3 Basic datatypes ... 26
7.4 Text and binary datatypes ... 37
7.5 Coded datatypes (terminology) ... 53
7.6 Identification and location datatypes ... 66
7.7 Name and address datatypes ... 77
7.8 Quantity datatypes ... 99
7.9 Collections of datatypes ... 128
7.10 Continuous set datatypes ... 143
7.11 Uncertainty datatypes ... 162
7.12 Structured text ... 165
Annex A (normative) XML representation ... 187
Annex B (normative) UML support types ... 190
Annex C (informative) RM-ODP viewpoint mappings ... 193
Annex D (informative) HL7 V3 Abstract Data Types mapping ... 194
Annex E (informative) Schema for XML representation ... 201
Bibliography ... 202

Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

Foreword

This document (EN ISO 21090:2011) has been prepared by Technical Committee ISO/TC 215 “Health informatics” in collaboration with Technical Committee CEN/TC 251 “Health informatics” the secretariat of which is held by NEN.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by August 2011, and conflicting national standards shall be withdrawn at the latest by August 2011.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CEN [and/or CENELEC] shall not be held responsible for identifying any or all such patent rights.

According to the CEN/CENELEC Internal Regulations, the national standards organizations of the following countries are bound to implement this European Standard: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and the United Kingdom.

Endorsement notice

The text of ISO 21090:2011 has been approved by CEN as a EN ISO 21090:2011 without any modification.

Documents
Document Sharing Hub

Full document details: DIN EN ISO 21090:2011-06

Introduction

Assistance from the Infrastructure and Messaging Committee in HL7 and the support of Connecting for Health have been instrumental in the preparation of this International Standard, which is a shared document between Health Level Seven (HL7) and ISO, and has been produced according the terms of the agreement between HL7, CEN and ISO (JIC, see http://www.global-e-health-standards.org/), which ensures that the content is fully available through ISO, CEN and HL7 publication channels.

Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

1 Scope

This International Standard

— provides a set of datatype definitions for representing and exchanging basic concepts that are commonly encountered in healthcare environments in support of information exchange in the healthcare environment;

— specifies a collection of healthcare-related datatypes suitable for use in a number of health-related information environments;

— declares the semantics of these datatypes using the terminology, notations and datatypes defined in ISO/IEC 11404, thus extending the set of datatypes defined in that standard;

— provides UML definitions of the same datatypes using the terminology, notation and types defined in Unified Modelling Language (UML) version 2.0;

— specifies an XML (Extensible Mark-up Language) based representation of the datatypes.

The requirements which underpin the scope reflect a mix of requirements gathered primarily from HL7 Version 3 and ISO/IEC 11404, and also from CEN/TS 14796, ISO 13606 (all parts) and past ISO work on healthcare datatypes.

This International Standard can offer a practical and useful contribution to the internal design of health information systems, but is primarily intended to be used when defining external interfaces or messages to support communication between them.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 4217, *Codes for the representation of currencies and funds*

ISO/IEC 8601, *Data elements and interchange formats — Information interchange — Representation of dates and times*

ISO/IEC 8824 (all parts), *Information technology — Abstract Syntax Notation One (ASN.1)*

ISO/IEC 11404:2007, *Information technology — General-Purpose Datatypes (GPD)*

ISO/TS 22220, *Health Informatics — Identification of subjects of health care*

IETF RFC 1738, *Uniform Resource Locators (URL)*

IETF RFC 1950, *ZLIB Compressed Data Format Specification version 3.3*

IETF RFC 1951, *DEFLATE Compressed Data Format Specification version 1.3*

IETF RFC 1952, *GZIP file format specification version 4.3*

IETF RFC 2045, *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*

IETF RFC 2046, *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*

IETF RFC 2396, *Uniform Resource Identifiers (URI): Generic Syntax*

Documents
Document Sharing Hub

Full document details: DIN EN ISO 21090:2011-06

IETF RFC 3066, *Tags for the Identification of Languages*

IETF RFC 3966, *The tel URI for Telephone Numbers*[1)]

FIPS PUB 180-1, *Secure Hash Standard*

FIPS PUB 180-2, *Secure Hash Standard*[2)]

Open Group, CDE 1.1, *Remote Procedure Call specification, Appendix A*

HL7 V3 Standard, *Data Types — Abstract Specification (R2)*

Regenstrief Institute, Inc. and the UCUM Organization, *The Unified Code for Units of Measure*[3)]

W3C Recommendation, *XML Signature Syntax and Processing*[4)]

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1
attribute
characteristic of an object that is assigned a name and a type

NOTE The value of an attribute can change during the lifetime of the object.

3.2
class
descriptor for a set of objects with similar structure, behaviour and relationships

3.3
code
concept representation published by the author of a code system as part of the code system, being an entity of that code system

3.4
code system
managed collection of concept identifiers, usually codes, but sometimes more complex sets of rules and references

NOTE They are often described as collections of uniquely identifiable concepts with associated representations, designations, associations and meanings.

EXAMPLES ICD-9, LOINC and SNOMED

1) Revision of IETF RFC 2806.

2) Revision of FIPS PUB 180-1.

3) Regenstrief Institute, Inc. and the UCUM Organization, Indianapolis, Indiana, USA [viewed 2010-08-23]. Available from: http://aurora.regenstrief.org/ucum.

4) World Wide Web Consortium (W3C) [viewed 2010-08-23]. Available from: http://www.w3.org/TR/xmldsig-core/.

Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

3.5
concept
unitary mental representation of a real or abstract thing; an atomic unit of thought

NOTE 1 It should be unique in a given code system.

NOTE 2 A concept can have synonyms in terms of representation and it can be a primitive or compositional term.

3.6
conformance
fulfillment of a specified requirement; adherence of an information processing entity to the requirements of one or more specific specifications or standards

3.7
datatype
set of distinct values, characterized by properties of those values, and by operations on those values

3.8
enumeration
datatype whose instances are a set of user-specified named enumeration literals

NOTE The literals have a relative order, but no algebra is defined on them.

3.9
generalization
taxonomic relationship between a more general class, interface or concept and a more specific class, interface or concept

NOTE 1 Each instance of the specific element is also an instance of the general element. Thus, the specific element has all the features of the more general element.

NOTE 2 The more specific element is fully consistent with the more general element and contains additional information.

NOTE 3 An instance of the more specific element can be used where the more general element is allowed.

3.10
information processing entity
anything that processes information and contains the concept of datatype, including other standards, specifications, data handling facilities and services

3.11
inheritance
mechanism by which more specific elements incorporate structure and behaviour of more general elements

3.12
interface
specifier for the externally-visible operations of class, without specification of internal structure

3.13
invariant
rule about the features of a class which must always be true

3.14
operation
service that an instance of the class may be requested to perform

NOTE An operation has a name and a list of arguments with assigned names and types, and returns a value of the type specified.

AVA Documents
Document Sharing Hub

Full document details: DIN EN ISO 21090:2011-06

3.15
specialization
taxonomic relationship between a more general class, interface or concept and a more specific class, interface or concept where the more specific entity adds new features or redefines existing features by constraining their possible behaviours

3.16
string character set
character set used in all string content throughout this International Standard

3.17
valueSet
that which represents a uniquely identifiable set of valid concept representations, where any concept representation can be tested to determine whether or not it is a member of the value set

NOTE A concept representation can be a single concept code or a post-coordinated combination of codes.

4 Abbreviated terms

For the purposes of this document, the following abbreviated terms apply.

CEN Comité Européen de Normalisation (European Committee for Standardization)

CNE Coded no exceptions

CWE Coded with exceptions

GPD General-Purpose Datatypes

HL7 Health Level Seven, Inc.

IETF Internet Engineering Task Force

OID Object Identifier

OMG Object Management Group

UML Unified Modelling Language

W3C World Wide Web Consortium

XML Extensible Mark-up Language

5 Conformance

5.1 General

An information processing product, system, element or other entity may conform to this International Standard either directly, by utilizing datatypes specified in this International Standard in a conforming manner, or indirectly, by means of mappings between internal datatypes used by the entity and the datatypes specified in this International Standard.

NOTE The term "information processing entity" is used as defined in 3.10, which is consistent with how it is used in ISO/IEC 11404:2007, Clause 4. Specifically, this definition includes applications and also other standards and specifications.

Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

5.2 Direct conformance

5.2.1 Direct conformance definition

An information processing entity which conforms directly to this International Standard shall:

a) specify which of the datatypes specified in Clause 7 are provided by the entity and which are not;

b) define the value spaces of the healthcare datatypes used by the entity to be identical to the value spaces specified by this International Standard;

c) specify to what extent the value spaces of the datatypes are constrained for use within its own context;

d) to the extent that the entity provides operations other than movement or translation of values, define operations on the healthcare datatypes which can be derived from, or are otherwise consistent with, the characterizing operations specified by this International Standard;

e) represent these datatypes using the Extensible Mark-up Language (XML) representation described herein, when the datatypes are represented in XML;

f) optionally, publish a formal conformance profile making these statements clear, or reference one published by some other information processing entity.

The above-mentioned requirements prohibit the use of a type-specifier defined in this International Standard to designate any other datatype (but, see 6.3 concerning the scope of the datatype names). They make no other limitation on the definition of additional datatypes in a conforming entity. For instance, a directly conforming information processing entity could continue to use ISO/IEC 11404 general-purpose datatypes in addition to these healthcare datatypes.

Requirement d) does not require all characterizing operations to be supported and permits the provision of additional operations. The intention is to permit the addition of semantic interpretation to the datatypes, as long as it does not conflict with the interpretations given in this International Standard. A conflict arises only when a given characterizing operation could not be implemented or would not be meaningful, given the entity provided operations on the datatype.

Examples of entities that could conform directly are language definitions or healthcare specifications whose datatypes, and the notation for them, are those defined herein. In addition, the verbatim support by a software tool or application package of the datatype syntax and definition facilities herein should not be precluded.

Information processing entities claiming direct conformance with this International Standard do not always need to use the datatypes defined in this International Standard to represent the concepts, i.e. simply because an address datatype is defined does not mean that the address datatypes must always be used for representing addresses. However, the type defined within this International Standard shall be used where the context is interoperability using these datatypes.

Information processing entities claiming direct conformance with this International Standard may further constrain the value domain of any of the datatypes within their context of use. The conformance statement shall make clear how constraints are applied within the information processing entity and how values that do not conform to the imposed constraints are handled.

Consistency of characterizing operations specified by conforming entities may be assessed by these criteria. Where operations have the same name as the operation defined within this International Standard, they are consistent if the operation can be invoked with the same parameters to return the same result. Other parameters may be defined, but shall have default values or be defined using additional definitions of operations with the same name but other parameter lists.

Information processing entities claiming direct conformance are not required to call any or all of the types defined in this International Standard "types". Other terms such as "data structures" may be used.

Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

5.2.2 Conformance statements

When an information processing entity claims direct conformance with this International Standard, it should make a conformance statement.

It is anticipated that other standards bodies would make conformance statements with regard to this International Standard both in a general sense and in the sense of adopting these datatypes for a particular standard. In addition, it is anticipated that certain countries publish profiles of these datatypes on either an advisory or a normative basis. Finally, vendors and purchasers of healthcare applications may well find use in creating, sharing and publishing these conformance statements.

This International Standard makes no rules about either the form of the statement or how it is published, but it should be clearly and formally presented and made available to all interested parties associated with the scope of the information processing entity.

In addition to specifying that conformance statements shall contain formal statements pertaining to 5.2.1 a) to d), this International Standard makes additional rules about what they shall or should say or may choose to do.

5.2.2.1 Direct conformance statements shall:

a) define which character set and encoding applies; the default is Unicode (see 6.7.5);

b) *if* an alternative mechanism for providing history and audit data is provided, define how it maps to the history and audit information on datatypes (see 7.1.3);

c) make clear how attribute and collection cardinality are specified (see 7.1.5);

d) define how the attributes nullFlavor, updateMode and flavorId on ANY are managed (see 7.3.3);

e) *if* quantities are used, make clear exactly how and when the QTY attributes expression, originalText, uncertainty and uncertaintyType are used;

f) make clear what methods may be used to provide alternative definitions for discrete set uniqueness (see 7.9.3);

g) *if* the structured documents types are used, document the scope of the document context and clearly define how references within this document context are resolved (see 7.12);

h) specify to what degree the XML format is adopted and define the namespace that is used (see A.1).

5.2.2.2 Direct conformance statements should:

a) define defaulting rules for language (see 7.4.2.3.7);

b) declare what languages are supported in the QTY.expression property (see 7.8.2.3.1);

c) describe which codes may be used in QSC.code (see 7.10.8.3);

d) *if* the structured documents types are used, define how version tracking works in the contexts where it is used (see 7.12.12.2.1).

5.2.2.3 Direct conformance statements may also:

a) define additional datatype flavors or additional authorities for the definition of flavors (see 6.7.6);

b) make additional arrangements for the use of derived data and the DER NullFlavor (see 7.1.4);

Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

c) define how the controlInformationRoot and controlInformationExtension properties on HXIT are used (see 7.3.2.3.4);

d) clarify how telecommunication and postal addresses are selected for particular purposes (see 7.6.2.3.2);

e) define the code systems to which different name and address part types are bound (see 7.7.3.6 and 7.7.5.6).

5.3 Indirect conformance

5.3.1 Indirect conformance definition

An information processing entity which conforms indirectly to this International Standard shall:

a) provide mappings between its internal datatypes and the healthcare datatypes conforming to the specifications of Clause 7;

b) specify for which of the datatypes in Clause 7 an inward mapping is provided, for which an outward mapping is provided and for which no mapping is provided;

c) specify whether the XML representation described in this International Standard is used when the datatypes are represented in XML, or whether it is used optionally to provide an alternative namespace for the XML representation;

d) optionally, publish a formal conformance profile making these statements clear or reference one published by some other information processing entity.

Examples of entities which could conform indirectly are healthcare specifications, applications, software engineering tools and other interface specifications, and many other entities that have a concept of datatype and an existing notation for it.

Standards for existing healthcare specifications yet to be proposed as International Standards are expected to provide for indirect conformance rather than direct conformance.

Information processing entities claiming indirect conformance with this International Standard do not always need to use the datatypes defined in this International Standard to represent the concepts, i.e. simply because an address datatype is defined does not mean that the address datatypes must always be used for representing addresses. However the type defined within this International Standard shall be used where the context is interoperability using these datatypes.

Information processing entities claiming indirect conformance with this International Standard may further constrain the value domain of any of the datatypes within their context of use. The conformance statement must make clear how constraints are applied within the information processing entity and how values that do not conform to the imposed constraints are handled.

Information processing entities claiming indirect conformance are not required to call any or all of the types defined in this International Standard "types". Other terms, such as "data structures" may be used.

5.3.2 Conformance statements

When an information processing entity claims indirect conformance with this International Standard, it should make a conformance statement.

This International Standard makes no rules about either the form of the statement or how it is published, but it should be made available to all interested parties associated with the scope of the information processing entity.

Documents
Document Sharing Hub

Full document details: DIN EN ISO 21090:2011-06

In addition to specifying that conformance statements shall contain formal statements pertaining to 5.3.1 a) to d), this International Standard makes additional rules about what they shall or should say or may choose to do.

5.3.2.1 Indirect conformance statements shall:

a) define which character set and encoding applies; the default is Unicode (see 6.7.5);

b) make clear what equality definitions apply and how (see 7.1.2);

c) make clear how attribute and collection cardinality are specified, if relevant (see 7.1.5);

d) *if* the structured documents types are used, document the scope of the document context and clearly define how references within this document context are resolved (see 7.12).

5.3.2.2 Indirect conformance statements should:

a) define defaulting rules for language (see 7.4.2.3.7);

b) if any exist, declare the mapping between W3C digital signature and alternate implementations (see 7.4.5.1).

5.3.2.3 Indirect conformance statements may also:

a) define additional datatype flavors or additional authorities for the definition of flavors (see 6.7.6);

b) make additional arrangements for the use of derived data and the DER NullFlavor (see 7.1.4);

c) define how the controlInformationRoot and controlInformationExtension properties on HXIT are used (see 7.3.2.3.4);

d) clarify how telecommunication and postal addresses are selected for particular purposes (see 7.6.2.3.2);

e) define the code systems to which different name and address part types are bound (see 7.7.3.6 and 7.7.5.6);

f) declare what languages are supported in the QTY.expression property (see 7.8.2.3.1);

g) describe which codes may be used in QSC.code (see 7.10.8.3);

h) *if* the structured documents types are used, define how version tracking works in the contexts where it is used (see 7.12.12.2.1).

6 Datatypes overview

6.1 What is a datatype?

In ISO/IEC 11404, a "datatype" is defined as a set of distinct values, characterized by properties of those values, and by operations on those values (ISO/IEC 11404:2007, 3.12).

A datatype consists of three main features:

— a value space;

— a set of properties;

— a set of characterizing operations.

Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

Generally, the definitions of the scope of datatypes revolve around one or other of the following notions.

— Immutability (the properties of the datatype cannot change, instead a new instance is created: datatypes have no lifecycle).

— The relationship between equality and identity (if two datatypes are equal they are the same instance).

— Coherency of a single concept (each datatype should represent a single concept space).

Since the application of these concepts to the healthcare information domain and the implications of these for the scope of datatypes are inherently a matter of perspective, the selection criterion for the datatypes defined in this International Standard is based on the set that has emerged from the debates held within the various stakeholder standardization bodies that define healthcare information standards. Since healthcare information standards and specifications are expected to provide mappings to this International Standard, the process has been deliberately inclusive. These other standards may choose to represent these datatypes with other more complex structures, but should explain how to interconvert these structures with the datatypes defined herein.

6.2 Definitions of datatypes

This International Standard defines a set of named datatypes. Each datatype defined in this International Standard is allocated both a short name and a long name. The formal name of the datatype is the short name. Each datatype is defined in two different ways:

— in terms of the datatype specification language and types defined in ISO/IEC 11404;

— in UML, using primitive types taken from the UML kernel package.

The ISO/IEC 11404 definition is provided to ensure continuity between this International Standard and the ISO/IEC 11404 GPDs, while the UML definition is provided to foster software-driven implementation of these datatypes. The ISO/IEC 11404 definitions are semantic and abstract in nature, while the UML definitions are concrete structural definitions. This International Standard is focused on providing structural concrete definitions, such that the UML definitions take precedence over the ISO/IEC 11404-based definitions, which are provided in the interests of continuity with ISO/IEC 11404.

The datatypes defined in this International Standard are an implementation of the HL7 V3 Abstract Data Types (R2). What this means is that it is possible to implement the exchange of information based on the HL7 V3 Abstract Data Type definitions using the datatypes defined in this International Standard. Annex B demonstrates how these datatypes are implementations of the HL7 V3 Abstract Data Types (R2).

The datatypes defined in this International Standard are not restricted to the features described by the HL7 V3 Abstract Data Types, nor is the HL7 V3 Data Types Abstract specification required in order to make use of these datatypes. The semantic definitions in the HL7 V3 Abstract Data Types may be consulted for further useful information to help implementors understand the use of these datatypes.

6.3 Datatype names and re-use of common datatype names

Some of the names of these datatypes bear superficial similarity to similar datatypes defined in other specifications. For instance, this International Standard defines a type REAL and there is a type Real in ISO/IEC 11404, and a type called Real in the UML kernel (see Note 1 in B.2.7 concerning the use of floating value types).

This International Standard is not attempting to redefine or replace the definition of real in ISO/IEC 11404 or UML. Instead, a new type that wraps the underlying "primitive" type is defined, which builds on the functionality of the underlying type and fits it into the overall architectural framework.

There are many specifications, languages and implementation technologies that declare similar types to either the underlying real types or the REAL defined in this International Standard, with a profusion of names around the common theme of Real, Float, Decimal or Double.

Documents
Document Sharing Hub

Full document details: DIN EN ISO 21090:2011-06

This International Standard does not seek to find new previously unrecognised names for the types it declares that represent these base concepts; instead it just assigns commonly used names to them. Again, this International Standard is not attempting to redefine or replace any other definitions by using these common names.

For these name clashes between the datatypes defined in this International Standard and any other specification, implementers should use some form of namespacing to ensure that the names of the datatypes do not cause confusion, perhaps by prefixing the names with some string constant in implementation environments that do not support proper namespacing of types.

The types BL, ST, INT, REAL, SET, LIST and BAG defined in this International Standard use this "primitive type wrapper" pattern.

6.4 Mapping to this datatypes International Standard

Like ISO/IEC 11404, this International Standard anticipates the use of these datatypes within other specifications. These specifications must specify how the datatypes and features described in this International Standard are implemented within the specification. Datatypes may be adopted and used directly, or they may be mapped to other datatypes or structures in different places, or they may not be supported at all.

Each specification that uses these datatypes should publish a document or section describing the mapping, and providing assistance for implementors to interconvert data between specifications.

6.5 Conformance with ISO/IEC 11404

This International Standard asserts direct conformance with ISO/IEC 11404. Although this International Standard can be considered to provide support for all the general purpose datatypes, only the following types are actually used in this International Standard:

— boolean;

— enumerated;

— characterstring;

— integer;

— Real;

— class;

— set;

— bag;

— sequence;

— octet.

The healthcare datatypes are class constructs built using these base primitive types. The healthcare datatypes are partially defined using the datatype definition language defined in ISO/IEC 11404. Since this language does not provide for generalization/specialization relationships, the datatypes cannot be fully defined in the ISO/IEC 11404 language, since the generalization/specialization relationships are an important part of the definition of the datatypes.

Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

6.6 Reference to UML 2

This International Standard defines the datatypes using the UML. The datatypes are all specializations of the UML Classifier, and the types are defined in this International Standard or are built on the following UML Kernel types as defined in the OCL 2 specification:

— enumeration;

— boolean;

— integer;

— string;

— collection;

— sequence;

— set;

— bag.

6.7 Modelling of datatypes

6.7.1 General

This International Standard presents the relationship between individual datatypes using the facilities of the Unified Modelling Language (UML) version 2. This modelling technique provides a means whereby:

— properties that are common across groups of datatypes can be expressed once;

— a mechanism is provided whereby one datatype within a specification may be substituted by another.

6.7.2 Attribute definitions

Unless otherwise specified, the default value for all attributes and associations is nil.

6.7.3 Generalization/specialization

This International Standard defines a series of class representations of healthcare datatypes. Within these classes, a number of generalization/specialization relationships are defined. This has the normal meaning associated with it as defined in the UML standard, and any instance of a class may be replaced by an instance of a specialization of that class. However, some of the specifications that rely upon this International Standard may make additional constraints concerning which specializations are permissible in a given context.

6.7.4 Enumeration definitions

This International Standard defines a number of attributes that have an enumerated set of possible values. Each value in the enumeration represents a concept in a terminology. Within the terminology, there may be generalization/specialization relationships. In this International Standard, the enumerations are defined in three ways:

— a list of codes as an ISO/IEC 11404 enumeration;

— a list of codes as a UML enumeration definitions;

— a table defining the enumeration in the narrative of the standard.

AVA Documents
Document Sharing Hub

Full document details: DIN EN ISO 21090:2011-06

The table has four rows: Level, Code, Title and Definition.

Level	The level of the concept in the hierarchy of the terminology. All concepts marked with the same level and not separated by a concept with a level of less numerical magnitude are siblings, and all concepts that follow after another concept with a higher level value are children of that concept.
Code	The code that represents the concept. This is used in the enumerations and to identify the concept in any representation or exchange of data in this International Standard. The Code is indented to represent the hierarchy of the terminology.
Title	A short human-readable description of the concept.
Definition	A short definition of the intention of the concept.

The hierarchy in the enumeration is an important part of the specification. Although the enumerations are defined as linear lists within ISO/IEC 11404 and the UML definitions, any information processing entity that asserts direct or indirect conformance with this International Standard shall conform to the relationships when evaluating meaning (implication) within the enumeration. In addition, this International Standard occasionally refers to the relationships within the narrative when defining the outcome of some operations.

Except in the case of the AddressPartType enumeration, the hierarchies represent generalization/specialization (also known as subsumption). In these hierarchies, the children codes represent a more specialized meaning of the parent code. The AddressPartType enumeration is compositional in nature; here codes represented as child codes of another code represent parts of the concept represented by the parent code.

As an example, the following is a subset of the table that defines the NullFlavor Enumeration

NullFlavor Enumeration. OID: 2.16.840.1.113883.5.1008			
Level	Code	Description	Definition
1	NI	No information	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.
2	UNK	Unknown	A proper value is applicable, but not known.
3	ASKU	Asked but unknown	Information was sought but not found (e.g. patient was asked but did not know).
4	NAV	Temporarily unavailable	Information is not available at this time but it is expected that it would be available later.
3	NASK	Not asked	This information has not been sought (e.g. patient was not asked).
2	MSK	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. Warning: Using this Nullflavor does provide information that may be a breach of confidentiality, even though no detailed data are provided. Its primary purpose is for those circumstances where it is necessary to inform the receiver that the information does exist without providing any detail.
2	NA	Not applicable	No proper value is applicable in this context (e.g. last menstrual period for a male).

Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

In this table, all the concepts listed below NI are indented and marked with a higher level, therefore they are all specializations of NI. Codes ASKU, NAV and MASK are all specializations of the concept UNK, and codes UNK, MSK and NA are siblings, specializations of NI, but not of anything else. Consequently, the eumeration value ASKU implies that the enumeration value UNK is also applicable.

All the enumerations in this International Standard are maintained by HL7, unless otherwise specified. Revised tables are published on a regular basis. The values defined in this International Standard will not have their meaning changed, though they may be deprecated. When these revised tables are published by HL7 or ISO, new enumeration values may be pre-adopted by trading partner agreement prior to the issuance of a new edition of this International Standard.

The OID for the HL7 codeSystem is published for reference, and to assist when passing these enumerated codes to terminology subsystems which are likely to use the OID to unambiguously refer to the code.

6.7.5 Strings and character encoding

This International Standard makes reference to both the ISO/IEC 11404 characterstring, and the UML Kernel String. For the purposes of this International Standard, these types define the same functionality: an immutable sequence of known length containing zero or more logical characters. This type is hereinafter referred to in the narrrative as simply "String".

NOTE 1 Both ISO/IEC 11404 and the UML kernel define additional characterizing operations which might be useful for implementors, and which can be considered to apply, but are not directly of interest to this International Standard.

NOTE 2 This International Standard also declares a wrapper type for String called ST, which adds additional functionality related to how the notion of an immutable sequence of characters fits into the overall framework of healthcare datatypes with their associated notions to do with uncertainty, unreliability and conformance.

The String datatype contains a sequence of logical characters, which is different from carrying a sequence of bytes that encodes a sequence of logical characters.

NOTE 3 Implementers should consider the difference between the two concepts carefully when implementing these datatypes.

By default, the String type contains Unicode characters. Information processing entities claiming direct or indirect conformance should mandate that the unicode character set be used in all String types throughout this International Standard.

However, there a few character sets throughout the world which are not perfectly mapped into Unicode. For this reason, there are a few countries or regulatory domains which may mandate some other character set than Unicode. In these contexts, standards and specifications that claim direct or indirect conformance would support the use of character sets other than Unicode, and should be explicit about which character sets are supported and how they are represented.

NOTE 4 There is an obvious implementation cost to choosing to use something other than Unicode. Regulatary domains that choose to use something other than Unicode generally have considered this issue at length.

The character set for any given implementation environment, whether Unicode or something else, is referred to throughout this International Standard as "the String Character Set“.

Whether the String Character Set is Unicode or something else, any given encoding that serves the implementation of the primitive string type in a particular character set may include control bytes that alter the interpretation of the text. It is assumed that any operations performed take this into consideration. However, as the operations described in this International Standard are enacted upon the logical sequence of characters, this issue is not discussed any further.

Documents
Document Sharing Hub

Full document details: DIN EN ISO 21090:2011-06

6.7.6 Flavors

In addition to the basic datatypes, this International Standard also defines a number of datatype flavors not independent datatypes in their own right – they are not defined as UML classes, nor as XML schema types. Instead, Datatype flavors describe common constraint patterns on the datatypes. As such, datatype flavors cannot introduce new attributes, new codes, default values, or any new defintional material. Datatype flavors may only make rules about what or how existing features of a class may be used.

Since datatype flavors may not introduce new features or meaning, and since they do not exist as independent classes in their own right, information processing entities do not need to understand a flavor in order to process the information correctly. For this reason, any information processing entity declaring direct or indirect conformance with this International Standard is able to define datatype flavors or reference flavors defined by some other authority, as long as the standard naming rules are followed. These are:

— names shall consist of a sequence of valid characters, namely letters, digits, underscores and periods; non-whitespace unicode characters may be used at the descretion of the information processing entity;

— names shall begin with the name of the type from which they are derived, followed by a period, a namespace, another period and then some additional valid characters;

— namespaces are used to prevent flavors described by different sources from having clashing names;. the namespace should either be an ISO 3166-1 country code, the applicable HL7 Realm identifier, or a DNS name.

Examples:

TS.CA.BIRTH	Rules for dates of birth published by the relevant Canadian Authority
TS.NPFIT.NHS_NUMBER	NHS Number flavor (fixed root) published by NPFIT in UK
ED.AU.KESTRAL.DOCUMENT	Rules for acceptable document format for the Australian company "Kestral"

Datatype flavors defined in this International Standard do not need or have a namespace.

Applications should not reject an instance because the flavorId attribute references a flavor that is not known to the application. Applications may reject an instance that references a flavor to which the instance does not conform, but are not required to do so.

The flavors defined in this International Standard do not need to be implemented by or used in association with information processing entities declaring direct or indirect conformance with this International Standard.

6.7.7 Examples

Examples are provided for most datatypes. The examples serve to illustrate various points related to how the datatypes are used.

The examples are all given as XML examples following the form documented in Annex A. The examples are presented assuming that the XML document/element that contains them has a charset of UTF-8. Most examples include an xsi:type making their type explicitly clear, but this is generally not necessary where the type is fully specified by the context of use and/or schema.

The discussion in the examples may provide references to content published by standard development organizations other than ISO or HL7. These reference materials are not a normative part of this International Standard.

Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

7 Datatypes

7.1 General properties

7.1.1 Immutability

These datatypes are conceptually immutable. The datatypes are defined with no lifecycle, and no operations that allow the existing datatype to change are defined. However, the datatypes are defined in this International Standard as classes with attributes, which allows for the value of the datatype to be modified after it is created.

Although this may be useful for implementation, it is not the intent of this International Standard to define types that are semantically mutable. Specifications that use these datatypes must consider the types themselves immutable. In particular, implementers shall be diligent to prevent the aliasing effects that may arise from allowing the datatype properties to change once they are in use.

7.1.2 Equality

There are two aspects to equality *viz* "are these two data values the same instance?" or "do these two data values represent the same semantic concept?".

In UML/OCL there are two properties to reflect these two aspects of equality. The first is "=", which returns whether these two values are the same instance, and the second is "equals", which returns whether these types represent the same concept. For OCL primitive types, such as integer, the two properties return the same value. For UML Classifiers, the results may be different.

In this International Standard, the equality criteria for each datatype are specified. This specifies the second form of equality: do these two data values represent the same concept? These definitions of equality are carefully constructed to meet the criterion that equal operations shall be reflexive, symmetric, and transitive, thus:

reflexive	x equals x must be true
symmetric	If x equals y is true, then y equals x must be true
transitive	If x equals y, and y equals z, then x must equal z

Because this form of equality is inherently a semantic notion, the equality definitions may depend on the semantics of the type, and it may not be simple to determine. Information processing entities claiming direct conformance with this International Standard shall conform to these equality definitions. Information processing entities claiming indirect conformance shall make clear what equality definitions apply.

Datatype flavors do not change the definition of equality.

7.1.3 History and audit trail

The base type HXIT defines properties for specifying a validTime during which the value is, was or will be valid, and for specifying the identity of an event that was associated with the changes (known as control information).

The validTime is not an audit trail of the information, used to track when any particular systems associated particular versions of data with a concept; it is used to make statements about the time period during which the data item was a correct description of the concept. For instance, in many countries a person may change their name by marriage or other legal means, and so a particular name is only associated with a person for a limited time. Similarly, a person's addresses and contact details may change as they move.

The control information is the identity that links to some event in an information system which is associated with “control” information concerning the change to this data value on a system. This information specifically relates to the association of data with its concept in systems. The control information reference can be used to build an audit trail of values across multiple exchanges between systems. See 7.3.2 for more details.

Documents
Document Sharing Hub

Full document details: DIN EN ISO 21090:2011-06

The various specifications that make use of this International Standard may provide alternative mechanisms for specifying this history and audit data, particularly the second part relating to system audit trails. In such cases, the specification will declare, usually in its conformance statement, how such information is handled and potentially mapped into the properties defined in this International Standard.

Datatypes are immutable – their value cannot change. The concept of valid time and control act does not apply within a datatype. For this reason, whenever a datatype is re-used as the type of an attribute of another datatype, the invariants will specify that the valid time and control act attributes shall be null.

7.1.4 Null and NullFlavor

The base type ANY introduces a concept called nullFlavor. Though the nullFlavor concept has some relationship with the UML/OCL null, it is not the same thing, and the relationship and differences between the two shall be understood to properly implement this International Standard.

Any instance of a class defined in this International Standard may be null. This is the null defined in UML and OCL. A null instance is an instance of the type OclVoid, and conforms to all types. It carries no other information other than the fact it is null. This International Standard uses the ocl operations oclIsDefined and oclIsUndefined to make constraints on the use of this form of null in attributes of the types defined in this International Standard.

Alternatively, an instance of the class may be created, and its nullFlavor can be set to one of the NullFlavors.

In this case, the value is referred to as nullFlavored, or exceptional. The intent of this is that the value represents an exception to the normal value domain of the type. This does not mean that an instance representing exceptional value is not bound by the rules defined in this International Standard; it must still meet the invariants defined herein. However many of the rules are different for exceptional values as they represent semantic exceptions to the normal data. All exceptional values shall have a NullFlavor, and the nullFlavor provides more information as to why the value is an exception to the rules. The opposite of an exceptional value is a proper value: this is a value with no nullFlavor.

Since an instance of the type must be created in order to carry a nullFlavor, it can also have values assigned to its other attributes. This is different from the UML/OCL null which does not exist and cannot have any nonNull attributes. If a nullFlavor is present, other attributes may be populated, but there is no requirement that any information processing entity make any use of these values, except that the following cases should be properly understood where applicable:

— NullFlavor OTH on CD;

— NullFlavors NINF and PINF on Interval low and high boundaries, respectively;

— NullFlavor UNK on an II with an extension and no root.

Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

NullFlavor Enumeration. OID: 2.16.840.1.113883.5.1008			
1	NI	No information	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.
2	INV	Invalid	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.
3	OTH	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).
4	PINF	Positive infinity	Positive infinity of numbers.
4	NINF	Negative infinity	Negative infinity of numbers.
3	UNC	Unencoded	No attempt has been made to encode the information correctly but the raw source information is represented (usually in originalText).
3	DER	Derived	An actual value may exist, but it must be derived from the information provided (usually an expression is provided directly).
2	UNK	Unknown	A proper value is applicable, but not known.
3	ASKU	Asked but unknown	Information was sought but not found (e.g. patient was asked but didn't know).
4	NAV	Temporarily unavailable	Information is not available at this time, but it is expected that it will be available later.
3	NASK	Not asked	This information has not been sought (e.g. patient was not asked).
3	QS	Sufficient quantity	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. 'Add 10 mg of ingredient X, 50 mg of ingredient Y, and sufficient quantity of water to 100 ml.' The null flavor would be used to express the quantity of water
3	TRC	Trace	The content is greater than zero, but too small to be quantified.
2	MSK	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. **WARNING — Use of this null flavor does provide information that may be a breach of confidentiality, even though no detailed data are provided. Its primary purpose is for those circumstances where it is necessary to inform the receiver that the information does exist without providing any detail.**
2	NA	Not applicable	No proper value is applicable in this context (e.g. last menstrual period for a male).

Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

ISO/IEC 11404 defines the concept of a sentinel value, which is a value in the value space of the type that does not share in all the characterizing operations of the type. Though there are some conceptual similarities between nullFlavors and sentinel values, instances of type ANY with nullFlavors are not sentinel values. The characterizing operations still apply, though the result of these characterizing operations will be some flavor of null. This is like the OCL null behaviour, behavioural similarity is why the property is named "nullFlavor".

Unless specifically documented otherwise, all operations defined in this International Standard behave in the same fashion.

— If the operation is performed on a UML/OCL null value, the result is a UML/OCL null.

— If the operation is performed on a value with a NullFlavor.

  — If the operation takes no parameters, it will return a NullFlavor. Usually the NullFlavor will be NA, but other NullFlavors may be appropriate, depending on the semantics of the NullFlavors. When performing operations upon nullFlavored values, the semantic meaning of the NullFlavor shall be considered.

  — If the operation takes parameters, and any of the parameters are a UML/OCL null, the result will be a UML/OCL null.

  — Where operations involve values that are null or have NullFlavors, the resulting value will be null or have a NullFlavor unless the semantics of the datatypes and NullFlavors dictate otherwise. There are a few rules about specific operations in this International Standard.

— If the operation is performed on a proper value.

  — If the operation takes parameters, and any of the parameters are a UML/OCL null, the result will be a UML/OCL null.

  — Where operations involve values that are null or have nullFlavors, the resulting value will be null or have a nullFlavor unless the semantics of the datatypes and nullFlavors dictate otherwise. There are a few rules about specific operations in this International Standard.

  — Otherwise, the operation will perform as described.

Some specific operations deviate from these rules. This is documented for each operation.

One special case arises with equality comparisons of various nullFlavored values. Two values that have NullFlavor NotApplicable are considered equal. While NINF cannot be equal to NINF and PINF cannot be equal to PINF, since the actual value is not known, NINF and PINF are clearly not equal. In other cases, it is generally not safe for the comparison to return anything but a NullFlavor – usually NI.

A value of any type with a NullFlavor of NI and where all the other attributes are null or meet this rule recursively is semantically equivalent with a UML/OCL null, and these forms may be interconverted if desired. Most simple attributes are declared using a UML or OCL type, such as String, and these may be converted to the complex equivalent, such as ST, with a NullFlavor of NI, if desired.

Because an attribute may be either null or a nullFlavor, many of the invariants take the form (x.oclIsDefined and x.isNotNull) implies {condition}. For some invariants, it is not necessary to make this rule since the result of null (the value of x.isNotNull if x is actually null) implies anything is null, and the invariant will fail gracefully, but for others the null must be protected against in order for the correct outcome in all cases.

The nullFlavor concept provides a general framework for handling incomplete data which is often encountered in healthcare information collection, use and analysis. The nullFlavor property may also play a special role in conformance frameworks in specifications that make use of these types.

Not all the NullFlavors can be used with all the different types. The NullFlavors PINF and NINF may only be used with specific types (INT, REAL, PQ and TS). The NullFlavor UNC may only be used with types that have an originalText (CD, QTY, QSET, and specializations). The NullFlavors QS and TRC may only be used with PQ.

Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

The two NullFlavors OTH and INV with their other specializations draw a distinction between the actual value and the value as represented in the instance. Some of the datatypes may be used to provide a representation of the value which requires subsequent transformation to generate the real value. For instance, an expression may be provided which generates an actual value that is in the required value domain of the instance, or an uncoded CD value – with just an originalText. INV, DER and UNC offer the possibility that some transform – either based on additional information or knowlege – may generate a valid value, whereas OTH with its specializations is an assertion that it is believed that no better value exists. This invites questions of confidence – how confident is the source that no better information exists, how sure is the processor that it believes the source is correct? However these are not resolvable. For this reason, the difference should not be taken as absolute, but as a statement of intent from the source.

Although INV and its specializations represent exceptional values in their context of use, they are only exceptional within the parameters defined by this International Standard. Actual values shall always conform to the rules defined by this International Standard.

The NullFlavor DER shall only be used where the context of use makes clear how the actual value can be derived from the information provided. This International Standard provides QTY.expression (see 7.8.2.3.1). Other information processing entities may make other arrangements for the use of DER in their conformance statements.

NOTE Most often, the correct NullFlavor to use is NI, and there need not be any burden on implementors with regard to choosing or persisting the correct NullFlavor. Other than some technical requirements that are clearly documented in this International Standard regarding the use of the NullFlavors NA, INV, DER, OTH, PINF and NINF, whenever it is not clear which NullFlavor is applicable, implementers should be comfortable using the NullFlavor NI. In particular, if a user does not respond to an input field in a data collection procedure, or the data are missing for some unknown reason, NI would be the most appropriate NullFlavor to choose.

EXAMPLES

Use Case	NullFlavor of Choice
User does not respond to input on a screen form	NI
Source is not configured to encode plain text input to required codeSystem	UNC
Source is unable to encode this particular plain text input to the required codeSystem because it cannot match the text	OTH
Patient is unconscious and cannot provide name	NAV
The system does not support this element	NI
No proper dosage is provided, but an expression is provided so that the destination system can calculate the proper dosage from the patient's weight	INV
The patient does not have an address – No Fixed Place of Abode	NA
Reporting the duration of an adverse reaction that is ongoing using an IVL<TS>	IVL.high = NA because the reaction is ongoing – the concept of high does not apply
Reporting the duration of an adverse reaction using an IVL<TS> when it is not known whether the reaction has terminated	IVL.high = UNK – we do not know
The source system is responding to a query for patient details, and has decided not to include the address because of applicable security and/or privacy policy.	MSK
NOTE Normal security/privacy policy is not to inform the information recipient that information has been suppressed, for good reason. However there are a few cases where workflow reasons dictate that it is necessary to inform the user that information has been denied. The MSK NullFlavor is provided for these minor cases.	

Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

7.1.5 Conformance

Conformance, as discussed in Clause 5, is concerned whether information processing entities conform to this International Standard. There is an additional aspect, which is also often called conformance, which concerns which rules are applied to the datatypes by information processing entities such as other standards when the datatypes are used.

Any information processing entity that uses these datatypes may constrain their use by making some human language narrative statements concerning how they are used, or by using some formal language statement in a processible language such as OCL. In addition, this International Standard recognises two additional means by which the possible values of the datatypes can be constrained, called "mandatory" and "cardinality“.

Any external attribute which is assigned a type from this International Standard may also have a nominal flag "mandatory" set to true. If the context of use sets this flag to true, the instance shall contain a valid data value that is not null, has no nullFlavor, conforms to all the constraints stated in this International Standard and any additional constraints on the value domain stated in the model. If this flag is not set to true, and the instance does not meet the constraints specified in the constraining model, the instance shall either be labelled using some form of nullFlavor (though other information may still be provided), or shall be omitted completely, in which case the default value (usually NullFlavor NI) applies.

The context of use may also apply a cardinality to an attribute. A cardinality consists of a minimum value, specified as a whole number, and a maximum value, specified as a whole number or "*". The cardinality is usually presented as [minimum value]..[maximum value], e.g. 0..1 or [1..*]. The meaning of the cardinality differs between collection based attributes and other attributes.

For attributes with a collection type (<dtref ref="dt-COLL"/> and its specializations), the cardinality specifies how many items may be in the collection. A cardinality maximum value of * means that there is no limit to the number of items in the collection.

NOTE 1 This does not imply that information processing entities are required to handle infinitely large collections of data, but the specification itself places no limit on the size of the collection). The minimum value specifies how many items must be in the collection.

NOTE 2 In the case of a mandatory collection, the collection shall contain at least one non-null (not null, and no nonFlavor) item.

For other attributes, the only cardinalities that may be applied are 0..0, 0..1, and 1..1. Cardinality of 0 means that the attribute is not to be represented in the instance, and has an implicit NullFlavor of NI. Cardinality of 1 means that the attribute has a value, though the value may be a NullFlavor unless the attribute is also mandatory.

A mandatory attribute shall have a minimum cardinality of 1 or more.

NOTE 3 This use of cardinality is a little different than the standard use of cardinality on attributes in UML. In UML, if an attribute is assigned a type of DSET(CS) and a cardinality of 2..3, this means that there must be 2 or 3 sets of CS. These uses are not incompatible; both forms of cardinality may be applied in any information processing entity that claims direct or indirect conformance. Which form is intended should be made clear in the documentation.

7.2 Top level model

For convenience and reference, Figure 1 provides an overview of the datatypes defined in this International Standard as a UML diagram (see Figure 1).

Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

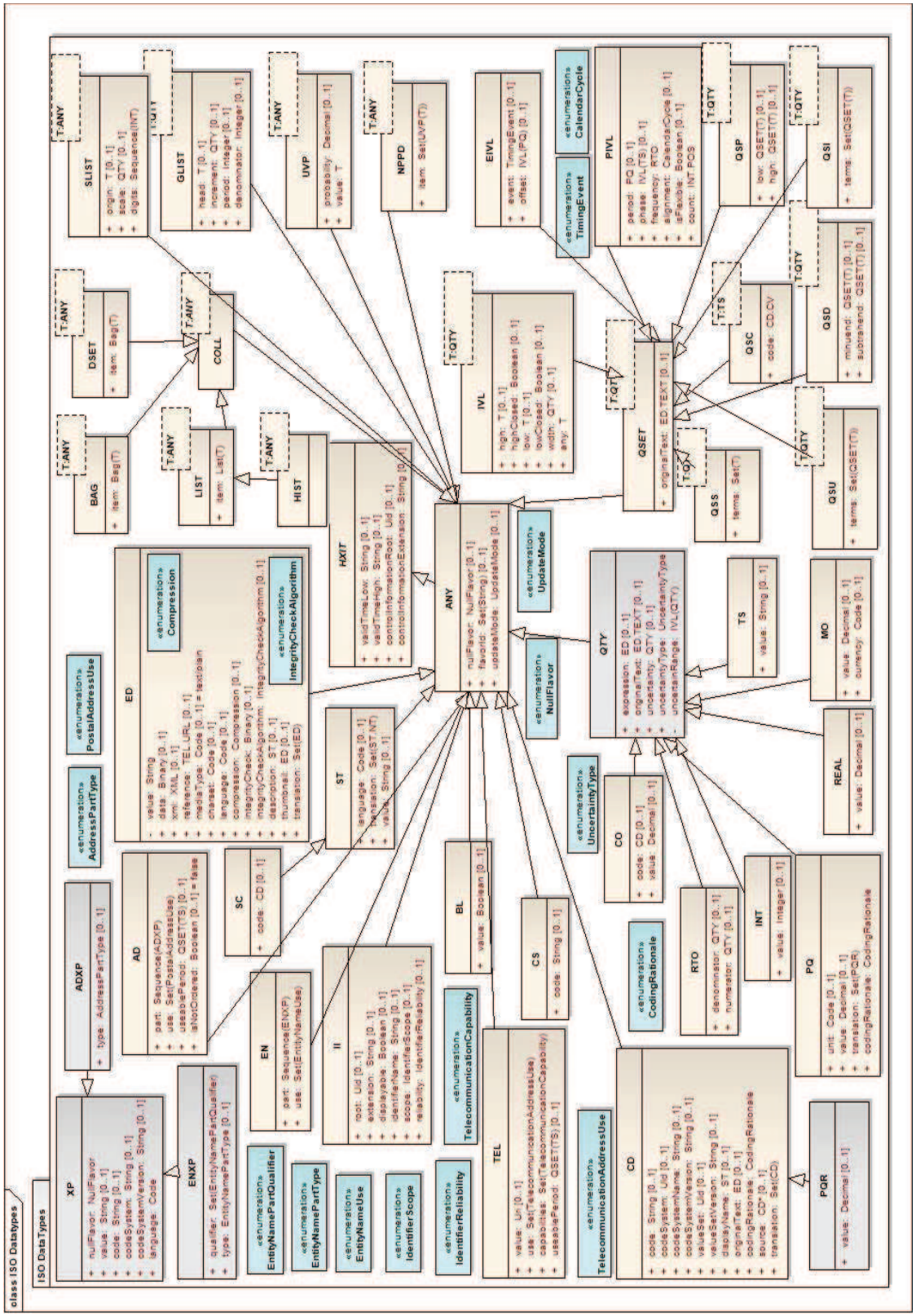

Figure 1 — Top level modelBasic datatypes

Documents
Document Sharing Hub

Full document details: DIN EN ISO 21090:2011-06

7.3 Basic datatypes

7.3.1 Overview

Basic datatypes are datatypes that provide infrastructural support for specific datatypes, which are defined in subsequent sections (see Figure 2).

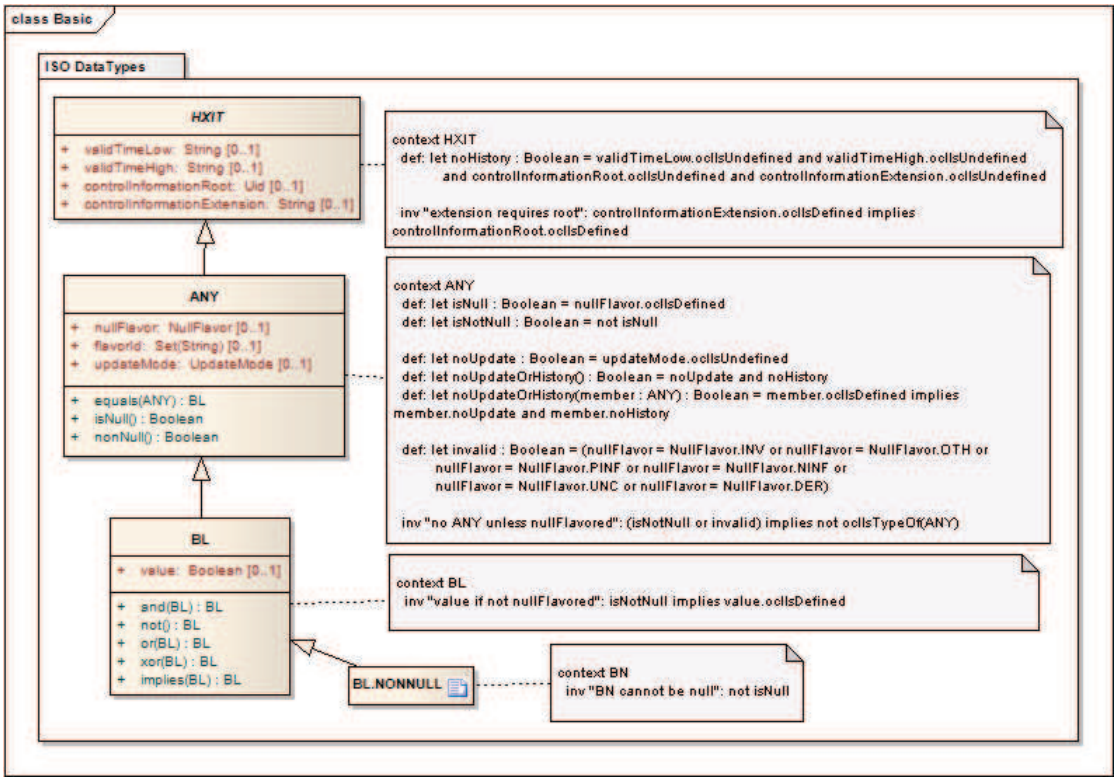

Figure 2 — Basic datatypes

7.3.2 HXIT

7.3.2.1 Description

Abstract and private – this datatype is not for use outside the datatypes in this International Standard.

Information about the history of this value: period of validity and a reference to an identified event that established this value as valid.

Because of the way that the types are defined, a number of attributes of the datatypes have values with a type derived from HXIT. In these cases the HXIT attributes are constrained to null. The only case where the HXIT attributes are allowed within a datatype is on items in a collection (DSET, LIST, BAG, HIST).

The use of these attributes is generally subject to further constraints in the specifications that make use of these types.

Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

7.3.2.2 ISO/IEC 11404 Syntax

```
type HXIT = class (
  validTimeLow : characterstring,
  validTimeHigh : characterstring,
  controlInformationRoot : characterstring,
  controlInformationExtension : characterstring
)
```

7.3.2.3 Attributes

7.3.2.3.1 validTimeLow : String: The time that the given information became or will become valid.

This is not the time that any system first observed the value, but the time that the actual value became correct (i.e. when a patient changes their name).

7.3.2.3.2 validTimeHigh : String: The time that the given information ceased or will cease to be correct.

Both validTimeLow and validTimeHigh shall be valid timestamps using the format described in 7.8.13.3.1 (TS.value).

7.3.2.3.3 controlInformationRoot : Uid: The root of the identifier of the event associated with setting the datatype to its specified value.

7.3.2.3.4 controlInformationExtension : String: The extension of the identifier of the event associated with setting the datatype to its specified value.

Together, the root and extension identify a particular record of a real world event that may supply additional information about the value such as who made the change, when it was made, why it was made, what system originated the change. These attributes exist because sometimes this information is required, but the value is being represented in an external context that does not contain a proper relationship to the control information for the value itself. The record need not be directly or easily resolvable. Conformance statements may make additional statements about these two properties, or about how such a reference should be resolved.

7.3.2.4 Equality

There is no equality definition for HXIT, since it is an abstract and private type. The attributes of HXIT (validTimeLow, validTimeHigh, controlInformationRoot, controlInformationExtension) never participate in the determination of equality of specializations of HXIT.

7.3.2.5 Invariants

— If a controlInformationExtension is provided, a ControlInformationRoot shall also be provided

OCL for invariants:

```
def: let noHistory : Boolean =
    validTimeLow.oclIsUndefined and
    validTimeHigh.oclIsUndefined and
    controlInformationRoot.oclIsUndefined and
    controlInformationExtension.oclIsUndefined

inv "extension requires root":
    controlInformationExtension.oclIsDefined implies
    controlInformationRoot.oclIsDefined
```


Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

7.3.2.6 Example

```
<example
  xsi:type="ST" value="This is some content"
  validTimeLow="200506011000" validTimeHigh="200507031500"
    controlInformationRoot="1.2.3.4.5.6">
</example>
```

In this example, the value of the example attribute "This is some content" was valid from June 1, 2005, 10 a.m. to July 3, 2005 at 3 p.m. The value was set to "This is some content" by an event which is uniquely identified by the OID of 1.2.3.4.5.6. Some information system somewhere (and how to determine that should be in an applicable conformance profile) will be able to resolve this OID to a reference that may be used to determine the user who entered these data into the system.

7.3.3 ANY

7.3.3.1 Description

Specializes HXIT

Defines the basic properties of every data value. This is conceptually an abstract type, meaning that no proper value can be just a data value without belonging to any concrete type. Every public concrete type is a specialization of this general abstract DataValue type.

However exceptional values (nullFlavored values) may be of type ANY, except for the exceptional values that imply the NullFlavor INV, since this requires a type to be meaningful. Not all NullFlavors may be used with the type ANY (see 7.1.4 for more details)

7.3.3.2 ISO/IEC 11404 Syntax

```
type ANY = class (
  validTimeLow : characterstring,
  validTimeHigh : characterstring,
  controlInformationRoot : characterstring,
  controlInformationExtension : characterstring,
  nullFlavor : NullFlavor,
  updateMode : UpdateMode,
  flavorId : Set(characterstring)
)
```

The appropriate use of all three of the attributes on ANY (in addition to the four properties inherited from HXIT) is intimately bound to the specification with which the datatypes are used, and generally that specification needs to establish special ways to control their use. Information Processing Entities claiming direct or indirect conformance shall make it clear how the use of these three attributes is controlled.

7.3.3.3 Attributes

7.3.3.3.1 nullFlavor : NullFlavor: If the value is not a proper value, indicates the reason.

Though the nullFlavor concept has some relationship with the UML/OCL null, it is not the same thing, and the relationship and differences between the two must be understood to properly implement this International Standard. For further discussion, see 7.1.4 (Null and NullFlavor).

NOTE nullFlavor includes the concept of a UML null value, and also includes potentially fully populated instances that do not conform to the requirements placed on the instance (also known as "exceptional instances").

Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

Both nonNull and nullFlavored values shall always be valid according to the rules expressed in this International Standard.

If populated, the value of this attribute shall be taken from the HL7 NullFlavor code system. The current values are:

NullFlavor Enumeration. OID: 2.16.840.1.113883.5.1008			
1	NI	No information	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.
2	INV	Invalid	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.
3	OTH	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).
4	PINF	Positive infinity	Positive infinity of numbers.
4	NINF	Negative infinity	Negative infinity of numbers.
3	UNC	Unencoded	No attempt has been made to encode the information correctly but the raw source information is represented (usually in originalText).
3	DER	Derived	An actual value may exist, but it must be derived from the information provided (usually an expression is provided directly).
2	UNK	Unknown	A proper value is applicable, but not known.
3	ASKU	Asked but unknown	Information was sought but not found (e.g. patient was asked but didn't know).
4	NAV	Temporarily unavailable	Information is not available at this time but it is expected that it will be available later.
3	NASK	Not asked	This information has not been sought (e.g. patient was not asked).
3	QS	Sufficient quantity	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."Add 10 mg of ingredient X, 50 mg of ingredient Y, and sufficient quantity of water to 100 ml." The null flavor would be used to express the quantity of water.
3	TRC	Trace	The content is greater than zero, but too small to be quantified.
2	MSK	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. Warning: Using this null flavor does provide information that may be a breach of confidentiality, even though no detailed data are provided. Its primary purpose is for those circumstances where it is necessary to inform the receiver that the information does exist without providing any detail.
2	NA	Not applicable	No proper value is applicable in this context (e.g. last menstrual period for a male).

Documents
Document Sharing Hub

Full document details: DIN EN ISO 21090:2011-06

ISO/IEC 11404 Syntax for nullFlavor attribute

```
type NullFlavor = enumerated (NI, INV, OTH, NINF, PINF, UNC, DER, UNK,
ASKU, NAV, QS, NASK, TRC, MSK, NA)
```

Some of the Null flavors are not generally applicable to all datatypes. The NullFlavors NINF, PINF, QS, and TRC shall only be used in association with QTY types. The NullFlavor UNC shall only be used with any type that has an originalText, and when UNC is used the originalText property shall be populated. When the NullFlavor DER is used, an expression shall be provided.

7.3.3.3.2 updateMode : UpdateMode: This property allows a sending system to identify the role that the attribute plays in processing the instance that is being represented.

If populated, the value of this attribute shall be taken from the HL7 UpdateMode code system. The current values are:

UpdateMode Enumeration. OID: 2.16.840.1.113883.5.57			
1	A	Add	The item was (or is to be) added, having not been present immediately before. (If it is already present, this may be treated as an error condition.)
1	D	Remove	The item was (or is to be) removed (sometimes referred to as deleted). If the item is part of a collection, delete any matching items.
1	R	Replace	The item existed previously and has been (or is to be) revised. (If an item does not already exist, this may be treated as an error condition.)
1	AR	Add or replace	The item was (or is to be) either added or replaced. No assertion is made as to whether the item previously existed.
1	N	No change	There was (or is to be) no change to the item. This is primarily used when this element has not changed, but other attributes in the instance have changed.
1	U	Unknown	It is not specified whether or what kind of change has occurred to the item, or whether the item is present as a reference or identifying property.
1	K	Key	This item is part of the identifying information for the object that contains it.

ISO/IEC 11404 Syntax for updateMode attribute

```
type UpdateMode = enumerated (A, D, R, AR, N, U, K)
```

If no updateMode is provided, there is no information as to how this information updates any existing information. The above-mentioned descriptions use the word "matching". For the purposes of the datatypes, this means the equality operations defined in this International Standard (in other contexts where this code system is used, "matching" may have other meanings).

NOTE UpdateMode does not affect the semantics or behaviour of the datatype itself, but can affect the behaviour of systems processing objects containing instances of the datatype.

Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

7.3.3.3.3 flavorId : Set(String): Signals the imposition of one or more sets of constraints on the datatype. The sole purpose of specifying that a constraint that has been used to further constrain the datatype is to support validation of the instance: a validation engine can look up the rules expressed for the specified flavors and confirm that the instance conforms to the rules for the flavor. No other processing should depend on the content of the flavor attribute.

No other semantic or computational use shall depend on the value of this property. If this value is populated, the datatype flavor(s) shall be a valid constraint on the type of the value.

There is further discussion about the use of datatype flavors in A.3.

7.3.3.4 Equality

By default, equals is determined as specified below. Selected specializations of ANY override equals to specify how semantic equality is evaluated for the type. Each type clearly documents how equality is determined.

The following table summarizes the relationship between null, nullFlavor and equals:

other **this**	null	nullFlavor	proper value
Null	null	null	Null
nullFlavor	null	nullFlavor[a]	nullFlavor[a]
proper value	null	nullFlavor[a]	proper value[b]

[a] First common generalization of both NullFlavors.
[b] Unless specifically defined for a specialization, use the general equality algorithm.

The general equality algorithm says that two values are equal if they have the same type, and if all the attributes not defined on HXIT or ANY are also equal. If any of the attribute's equality is null or a nullFlavor then the result is null or has the most common NullFlavor.

NOTE 1 See the comments on comparing nullValues in 7.1.4 (Null and NullFlavors).

The equality operation is *reflexive*, *symmetric*, *transitive*, and *consistent*, and implementations shall conform to these requirements. The equality rules defined in this International Standard conform to these requirements.

This operation conforms to the general rules for operations and nullFlavors defined in 7.1.4 (Null and NullFlavor), but the rules are described here in depth, for greater clarity. In particular, the rules are transitive within the equals operation: if any of the attributes or collection items have a nullFlavor (other than the NullFlavor NA), the result will become a value of that nullFlavor, unless specifically defined otherwise.

Equals does not override the = operation defined in OCL, nor should it override the normal equivalent for the OCL = operation in any implementation platform. It defines semantic equality.

NOTE 2 UpdateMode and flavorId are always ignored when testing for equality.

7.3.3.5 Invariants

An instance may only be of type ANY (not a specialization) if it has a NullFlavor, and not if the NullFlavor implies INV

Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

OCL for invariants:

```
def: let isNull : Boolean = nullFlavor.oclIsDefined
def: let isNotNull : Boolean = not isNull
def: let noUpdate : Boolean = updateMode.oclIsUndefined
def: let noUpdateOrHistory() : Boolean = noUpdate and
   noHistory
def: let noUpdateOrHistory(member : ANY) : Boolean =
   member.oclIsDefined implies member.noUpdate and member.noHistory
def: let invalid : Boolean = (nullFlavor = NullFlavor.INV or
     nullFlavor = NullFlavor.OTH or nullFlavor = NullFlavor.PINF or
     nullFlavor = NullFlavor.NINF or nullFlavor = NullFlavor.UNC or
     nullFlavor = NullFlavor.DER)

inv "no ANY unless nullFlavored": (isNotNull or invalid) implies
       not oclIsTypeOf(ANY)
```

7.3.3.6 Operations

7.3.3.6.1 equals(other : ANY) : BL: Defines whether the this data value is considered semantically equal to another data value – that they carry the same meaning. See 7.3.3.4 for a discussion on how equality is determined.

7.3.3.6.2 isNull() : Boolean: Defines whether this type has a nullFlavor or not.

This operation is an exception from the normal rules for operations and nullFlavors: it will return true if the value has a nullFlavor, and false if it does not.

7.3.3.6.3 notNull() : Boolean: Defines whether this type does not have a nullFlavor or it does.

This operation is an exception from the normal rules for operations and nullFlavors: it will return false if the value has a nullFlavor, and true if it does not.

7.3.3.7 Examples

7.3.3.7.1 Simple true value

```
<example xsi:type="ANY" nullFlavor="UNK"/>
```

The value is unknown, and it is not even known what type the value might be (not fixed by context and not known in the instance).

7.3.3.7.2 Null and NullFlavor

This is a simple CD representing a coded concept (see 7.5.2)

```
<example code="784.0" codeSystem="2.16.840.1.113883.6.42">
  <displayName value="Headache"/>
</example>
```

This provides a code, a codeSystem, and a displayName. All the other attributes are null. In terms of meaning, the following instance is identical:

```
<example code="784.0" codeSystem="2.16.840.1.113883.6.42">
  <displayName value="Headache"/>
  <originalText nullFlavor="NI"/>
</example>
```


Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

This has the same meaning because if an attribute is null, all we can say is that we have no information concerning it, nor do we know why we have no information about it, which is the same statement as NullFlavor.NI: we have no information about why we are not a proper type.

In the same sense, this also has the same meaning:

```
<example code="784.0" codeSystem="2.16.840.1.113883.6.42">
  <displayName value="Headache"/>
  <translation nullFlavor="NI"/>
</example>
```

In this case, there is a clear difference: a translation exists. However there is no information about the translation, nor why there is no information. So the outcome in terms of meaning is the same.

This example shows that it is valid to provide additional information along with a nullFlavor:

```
< example code="784.0" codeSystem="2.16.840.1.113883.6.42">
  <displayName value="Headache"/>
  <translation nullFlavor="NI" codeSystem="2.16.840.1.113883.6.96"/>
</example>
```

This is a slightly different statement: that there is no information about the translation of this code into SNOMED-CT. There are cases where this information is of significance, though they are not common.

7.3.3.7.3 UpdateMode

The principle use for updateMode is in tightly coupled messaging systems. A tightly coupled message system is one where a limited number of applications have tight trading partner agreements, and well understood and managed information flows. In these circumstances, it may be advantageous to agree that instead of sending all available information in each message, only information about what has changed in each transaction is sent in each message. Doing this has the advantage of greatly saving on implementation costs for both sender and receiver, while raising the prospect of information loss or scrambling if the message flows get out of sequence. UpdateMode is required to properly implement information about what has changed in a transaction.

In more general use, such as clinical documents and EHRs, the use of such transaction-based processing is inappropriate, therefore the updateMode attribute should generally be fixed to null in these contexts.

The following examples provide some examples for how updateMode might be used in a tightly coupled messaging system. These examples should not taken as proscriptive guidance for how updateMode and the associated transaction-based processing should be implemented.

The first set of examples concern a simple case where an object model has a single attribute for the birth date of a person, birthDate: TS.

A user opens the patient management dialog box, and changes the patient’s birthdate to June 21, 1975. Since the system already has a birth date, the existing patient's birth date is being replaced. This leads to an instance being sent to the target tightly coupled application that includes the following fragment:

```
<birthDate value="19750621" updateMode="R"/>
```

The application checks and finds that it already has a value, so replaces it with the new value. If no value already existed, this would indicate an error condition, though how this should be handled depends on the details of the local agreements in force to make this kind of processing safe.

Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

Later, another user opens the patient management dialog box and removes the patient's birth date. This leads to an instance being sent to the target tightly coupled application that includes the following fragment:

```
<birthDate nullFlavor="NI" updateMode="D"/>
```

A nullFlavor is required to make the value valid, and means to delete the birth date and replace it with the NullFlavor NI. This is effectively the same as:

```
<birthDate nullFlavor="NI" updateMode="R"/>
```

which means to update the birthdate to the NullFlavor NI. As seen in this instance, updateMode is not very useful on single instances like this. Where it becomes useful is to differentiate between information sent because it is changing (UpdateMode.R) from information sent to establish identity. To illustrate this, it is assumed that the trading party agreement controlling this transactionally based processing has specified that when updating a patient record, the patient object always includes name, gender and date of birth as identifying data. In this case, the instance always contains a birth date. It can be marked that the birthDate attribute is present as an identifying rather than a changing attribute using the UpdateMode "Key":

```
<birthDate nullFlavor="NI" updateMode="K"/>
```

Where updateMode starts to become really useful is with maintaining lists. The following set of examples concern the patients contact list.

Typical applications have different entries on the user UI for the different kind of contact details. The particular kinds of contact details that need to be supported tend to be rather a moving target in the information age, so this International Standard implements the actual contact details as a typed list (increasingly many applications are following suit). In this hypothetical case, there is input fields for home telephone number, mobile telephone number, fax number and e-mail address, and the attribute is contacts: DSET(TEL). A typical patient record might generate a contact list like this in a non-transactional environment:

```
<contacts>
  <item value="tel:+11015551234" use="H" capabilities="voice"/>
  <item value="tel:+11995556787" use="MC" capabilities="voice sms"/>
  <item value="tel:+11015551235" capabilities="fax"/>
  <item value="mailto:example@example.com"/>
</contacts>
```

In a tightly coupled messaging system using transactional processing based on updateMode, this full list would rarely be sent. Instead, only bits of it are sent.

If a user entered the patient management dialog box and changed the user's home telephone number, the following instance would be sent:

```
<contacts>
  <item value="tel:+11015551234" use="H"
      capabilities="voice" updateMode="D"/>
  <item value="tel:+12315559876" use="H"
      capabilities="voice" updateMode="A"/>
</contacts>
```


Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

The receiving application compares its information with the instance. If it cannot find the existing number +11015551234, that is an error to be handled according to local agreements. If it can, it deletes it, and adds the new number. It might be tempting to send this instance instead:

```
<contacts>
  <item value="tel:+12315559876" use="H"
      capabilities="voice" updateMode="R"/>
</contacts>
```

with the instruction to replace the existing number. However, which number should be replaced? There is no answer, except that matching is based on equality; thus, this says to replace the home number +12315559876 with the home number +12315559876.

For this reason, it is best to use Add and Delete with datatype values. Replace and Add/Replace are not very useful since matching is done on equality – all that can be said is "replace value A with value A" – of limited use. These are generally more useful one complex objects, but have been included in the list of legal values for datatypes to ensure that all known transactional processing use cases are met.

There is one use case that UpdateModel.R can be used for. Since matching is done by equality, and equality is only based on the number itself, the following two instances have the same meaning:

```
<contacts>
  <item value="tel:+11015551234" use="H"
      capabilities="voice" upateMode="D"/>
  <item value="tel:+ 11015551234" use="H WP"
      capabilities="voice fax" updateMode="A"/>
</contacts>
```

```
<contacts>
  <item value="tel:+ 11015551234" use="H W"
      capabilities="voice fax" updateMode="R"/>
</contacts>
```

Both these instances mean to replace the existing details for +12315559876 with a new set of details for +12315559876, updating the use and capabilities.

7.3.4 BL

7.3.4.1 Description

Specializes ANY

BL stands for the values of two-valued logic. A BL value can be either true or false, or may have a nullFlavor.

7.3.4.2 ISO/IEC 11404 Syntax

```
type BL = class (
  validTimeLow : characterstring,
  validTimeHigh : characterstring,
  controlInformationRoot : characterstring,
  controlInformationExtension : characterstring,
  nullFlavor : NullFlavor,
  updateMode : UpdateMode,
  flavorId : Set(characterstring),
  value : boolean
)
```


Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

With any data value potentially having a nullFlavor, the two-valued logic is effectively extended to a three-valued logic as shown in the following truth tables:

NOT			AND	true	false	null		OR	true	false	null
True	false		true	true	false	null		true	true	true	True
false	true		false	false	false	false		false	true	false	null
null	null		null	null	false	null		null	true	null	null

In this table, null stands for either true null or a nullFlavor. If the null (or either of the nulls) is a true null, the result will be a true null. Where a boolean operation is performed upon two datatypes with different NullFlavors, the nullFlavor of the result shall be any common ancestor of the two different nullFlavors. The result should be the first common ancestor.

7.3.4.3 Attributes

7.3.4.3.1 value: Boolean: The value of the BL.

This is an example of the primitive type wrapping pattern. See 6.3 for additional details.

7.3.4.4 Equality

Two boolean values are equal if they are nonNull and have the same value.

7.3.4.5 Invariants

The BL shall have a value if it does not have a nullFlavor

OCL for invariants:

```
inv "value if not nullFlavored":
            isNotNull implies value.oclIsDefined
```

7.3.4.6 Operations

7.3.4.6.1 and(other : BL) : BL: True if both values are true. False if either value is false. Null or nullFlavored otherwise.

7.3.4.6.2 or(other : BL) : BL: True if either this or other are true. False if both values are false. Null or nullFlavored otherwise

7.3.4.6.3 xor(other : BL) : BL: True if either this and other are different and not null. False if both this and other have the same value. Null or nullFlavored otherwise.

7.3.4.6.4 implies(other : BL) : BL: True if either this is false, or if this and other are true. False if this is true and other is false. Null or nullFlavored otherwise.

7.3.4.6.5 not() : BL: False if this true, true if this is false. Null or nullFlavored otherwise.

NOTE These operations do not always confom to the general rules for nullFlavors and operations due to the special nature of these logical operations. For example, (null/nullFlavored) or True is true, because it does not matter whether the missing value might actually be true or false, and therefore the result will be the same. Also, from an implementer's point of view, these operations are strictly symmetric, and this can require some special case handling to prevent implementation errors when performing operations on null objects.

Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

7.3.4.7 Examples

7.3.4.7.1 Simple True Value

```
<example xsi:type="BL" value="true"/>
```

7.3.4.7.2 Unknown Value

```
<example xsi:type="BL" nullFlavor="UNK"/>
```

7.3.5 BL.NONNULL

7.3.5.1 Description:

A flavor that constrains BL

BL.NONNULL is a constrained instance of BL that cannot have a nullFlavor. By implication, a null can never be used in the place of a BL.NONNULL, though this is not a rule that can be enforced by this International Standard.

7.3.5.2 Invariants

BL.NONNULL cannot have a nullFlavor

OCL for invariants:

```
inv "cannot have a nullFlavor": not isNull
```

7.4 Text and binary datatypes

7.4.1 Overview

Text and binary datatypes are datatypes that provide support for text and multimedia data (see Figure 3).

Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

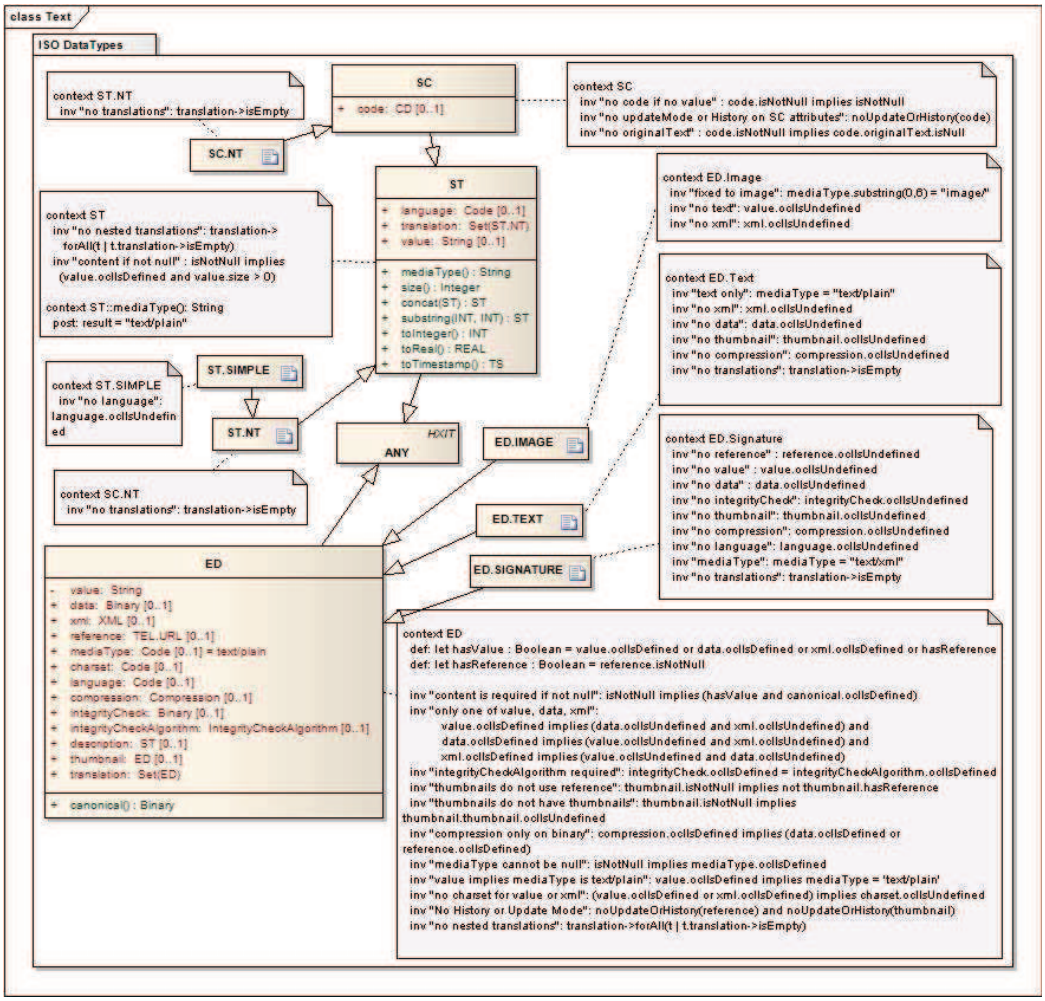

Figure 3 — Text and binary datatypes

7.4.2 ED (Encapsulated Data)

7.4.2.1 Description

Specializes ANY

Encapsulated data are data that are primarily intended for human interpretation or for further machine processing outside the scope of this International Standard. This includes unformatted or formatted written language, multimedia data, or structured information as defined by a different standard (e.g. XML-signatures).

NOTE Encapsulated data can be present in two forms, inline or by reference. The content is the same whether it is located inline or remote.Inline data are communicated or moved as part of the encapsulated data value, whereas by-reference data may reside at a different location: a URL/URI that provides reference to the information required to locate the data. Inline data may be provided in one of three different ways:

— as a plain sequence of characters (value):

— as a binary (a sequence of bytes) (data):

— as xml content (xml).

Content shall be provided if the ED has no nullFlavor. The content may be provided in-line (using only one of value, data or xml), or it may be provided as a reference.Content may be provided in-line and a reference also may be given; in these cases, it is expected that the content of the reference will be exactly the same as the in-line content. Information processing entities are not required to check this, but may regard it as an error condition if the content does not match.

Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

7.4.2.2 ISO/IEC 11404 Syntax

```
type ED = class (
  validTimeLow : characterstring,
  validTimeHigh : characterstring,
  controlInformationRoot : characterstring,
  controlInformationExtension : characterstring,
  nullFlavor : NullFlavor,
  updateMode : UpdateMode,
  flavorId : Set(characterstring),
  value : characterstring,
  data : Sequence(Octet),
  xml : XML,
  reference : TEL.URL,
  mediaType : characterstring,
  charset : characterstring,
  language : characterstring,
  compression : Compression,
  integrityCheck : Sequence(Octet),
  integrityCheckAlgorithm : IntegrityCheckAlgorithm,
  description : ST,
  thumbnail : ED,
  translation : Set(ED)
)
```

7.4.2.3 Attributes

7.4.2.3.1 value : String: A simple sequence of characters that contains the content.

If value is used, the mediatype is fixed to text/plain and the charset shall be consistent with the String Character Set. See 6.7.5 for more details.

7.4.2.3.2 data : Binary: A simple sequence of byte values that contains the content.

7.4.2.3.3 xml : XML: The content represented in plain XML form.

A direct representation is provided for XML. This is because this International Standard includes an XML serialization of the data, and this xml attribute is handled specially in the serialization form. The xml data are not different in any semantic sense to the same data if represented in the value or data attributes.

NOTE These three representations of the ED data –as a sequence of characters, as a sequence of bytes, or as XML in a native XML format – are mutually incompatible and could also have been implemented as three specializations of an abstract ED supertype. However, doing so would complicate definition and implementation of the ED flavors and complicate the associated XML format (the additional of a mandatory xsi:type attribute) without significantly simplifying the overall implementation of ED.

Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

7.4.2.3.4 reference : TEL.URL : A URL the target of which provides the binary content.

The semantic value of an encapsulated data value is the same, regardless of whether the content is present as inline content or just by reference. However, an encapsulated data value without inline content behaves differently, since any attempt to examine the content requires the data to be downloaded from the reference. An encapsulated data value may have both inline content and a reference.

If data are provded in the value, data or xml attributes, the reference shall point to the same data. It is an error if the data resolved through the reference does not match either the integrity check, data as provided, or data that had earlier been retrieved through the reference and then cached. The mediatype of the ED shall match the type returned by accessing the reference.

The reference may contain a usablePeriod to indicate that the data may only be available for a limited period of time. Whether the reference is limited by a usablePeriod or not, the content of the reference shall be fixed for all time. Any application using the reference shall always receive the same data, or an error. The reference cannot be re-used to send a different version of the same data, or different data.

7.4.2.3.5 mediaType : Code: Identifies the type of the encapsulated data and can be used to determine a method to interpret or render the content.

The IANA defined domain of media types is established by the IETF RFCs 2045 and 2046. mediaType has a default value of text/plain and cannot be null. If the media type is different to text/plain, the *mediaType* attribute shall be populated.

If the content is compressed using a specified compression algorithm, the mediaType shall refer the mediaType of the uncompressed data, whether the data are accessed by reference or not.

7.4.2.3.6 charset : Code: An Internet Assigned Numbers Authority (IANA) Charset Registered character set and character encoding for character-based encoding types**.**

Whenever the content of the ED is character type data in any form, the charset property needs to be known. If the content is provided directly in the value attribute, then the charset shall be a known character set consistent with the String Character Set (see 6.7.5 for more details). If the content is provided as a reference, and the access method does not provide the charset for the content (such as by a mime header), then the charset shall be conveyed as part of the ED.

7.4.2.3.7 language: Code: The human language of the content. Valid codes are taken from IETF RFC 3066. If this attribute is null, the language may be inferred from elsewhere, either from the context or from unicode language tags, for example.

Conformance profiles should define defaulting rules for language for a given usage environment of this International Standard.

NOTE While language attribute usually alters the interpretation of the text, the language attribute does not alter the meaning of the characters in the text.

Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

7.4.2.3.8 compression : Compression: The compression algorithm, if any, used on the raw byte data.

If the attribute is null, the data are not compressed. Compression only applies to the binary form of the content.

If populated, the value of this attribute shall be taken from the HL7 CompressionAlgorithm code system. The current values are:

CompressionAlgorithm Enumeration. OID: 2.16.840.1.113883.5.1009			
1	DF	deflate	The deflate compressed data format as specified in IETF RFC 1951.
1	GZ	gzip	A compressed data format that is compatible with the widely used GZIP utility as specified in IETF RFC 1952 (uses the deflate algorithm).
1	ZL	zlib	A compressed data format that also uses the deflate algorithm. Specified as IETF RFC 1950.
1	Z	compress	Original UNIX compress algorithm and file format using the LZC algorithm (a variant of LZW). Patent encumbered and less efficient than deflate.
1	BZ	bzip	bzip-2 compression format. See [http://www.bzip.org/] for more information.
1	Z7	Z7	7z compression file format. See [`http://www.7-zip.org/7z.html`] for more information.

Some compression formats allow multiple archive files to be embedded within a single compressed volume. Applications shall ensure that the decompressed form of the data conforms to the stated media type.

ISO/IEC 11404 Syntax for compression attribute

```
type Compression = enumeration (DF, GZ, ZL, Z, BZ, Z7)
```

7.4.2.3.9 integrityCheck : Binary: A checksum calculated over the binary data.

The purpose of this property, when communicated with a reference is for anyone to validate later whether or not the reference still resolved to the same content that the reference resolved to when the encapsulated data value with reference was created. If the attribute is null, there is no integrityCheck.

It is an error if the data resolved through the reference does not match the integrity check.

The integrity check is calculated according to the integrityCheckAlgorithm. By default, the Secure Hash Algorithm-1 (SHA-1) shall be used. The integrity check is binary encoded according to the rules of the integrity check algorithm.

The integrity check is calculated over the raw binary data that is contained in the data component, or that is accessible through the reference. No transformations are made before the integrity check is calculated. If the data are compressed, the Integrity Check is calculated over the compressed data.

AVA Documents
Document Sharing Hub

Full document details: DIN EN ISO 21090:2011-06

7.4.2.3.10 integrityCheckAlgorithm : IntegrityCheckAlgorithm: The algorithm used to compute the integrityCheck value.

If populated, the value of this attribute shall be taken from the HL7 IntegrityCheckAlgorithm code system. The current values are:

IntegrityCheckAlgorithm Enumeration. OID: 2.16.840.1.113883.5.1010			
1	SHA1	secure hash algorithm - 1	This algorithm is defined in FIPS PUB 180-1: Secure Hash Standard. As of April 17, 1995.
1	SHA256	secure hash algorithm - 256	This algorithm is defined in FIPS PUB 180-2: Secure Hash Standard.

ISO/IEC 11404 Syntax for integrityCheckAlgorithm attribute

```
type IntegrityCheckAlgorithm = enumeration (SHA1, SHA256)
```

7.4.2.3.11 description : ST: An alternative description of the media where the context is not suitable for rendering the media.

For example, short text description of an image or sound clip, etc.; this attribute is not intended to be a complete substitute for the original. For complete substitutes, use the "translation" property.

The intent of this property is to allow compliance with disability requirements, such as those expressed in the Americans with Disability Act (also known as "Section 508"), where there is a requirement to provide a short text description of included media in some form that can be read by a screen reader. This is similar to a very short thumbnail with mediaType = text/plain.

7.4.2.3.12 thumbnail : ED: An abbreviated rendition of the full content.

A thumbnail requires significantly fewer resources than the full content, while still maintaining some distinctive similarity with the full content. A thumbnail is typically used with by-reference encapsulated data. It allows a user to select the appropriate content more efficiently before actually downloading through the reference.

Originally, the term thumbnail refers to an image in a lower resolution (or smaller size) than another image. However, the thumbnail concept can be metaphorically used for media types other than images. For example a movie may be represented by a shorter clip; an audio-clip may be represented by another audio-clip that is shorter, has a lower sampling rate, or a glossy compression; or an abstract provided for a long document.

A thumbnail itself shall not contain a thumbnail.

7.4.2.3.13 translation : Set(ED): Alternate renditions of the same content translated into a different language or a different mediaType.

The translations shall convey the same information, but in a different language or mediaType. Translations shall not contain translations. The translations do not take part in the test for equality, consequently shall not introduce any new semantics to the value.

7.4.2.4 Equality

Two nonNull values of type ED are equal if – and only if – their mediaType and data are equal. For those ED values with compressed data or referenced data, only the dereferenced and uncompressed data count for the equality test (the canonical content; see 7.4.2.6.1). The compression, thumbnail, translation and reference property themselves are excluded from the equality test. In addition the language property is excluded from the test (refer to 7.4.2.7.2). If the mediaType is character-based and the charset property is not equal, the charset property must be resolved through mapping of the data between the different character sets.

Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

An ED with a plain text content may also be equal to a ST with the same character content, following the rules described above.

7.4.2.5 Invariants

— Either reference, data, value or xml shall be provided if not nullFlavored, and at least one byte of data shall be referenced.

— Only one of data, value or xml may be specified.

— An integrityCheckAlgorithm shall be provided and may only be provided if an integrityCheck is provided.

— if a thumbnail is provided, it shall not use a reference.

— if a thumbnail is provided, it shall not have a thumbnail.

— Compression can only be specified if data are provided as a binary or as a reference.

— Mediatype cannot be null.

— If value is used, the mediaType is plain text.

— A character set shahll not be asserted for plain text or xml content (for plain text see 6.7.5 and implicitly derived for the XML content).

— Translations may not contain translations.

OCL for invariants:

```
def: let hasValue : Boolean = value.oclIsDefined or
       data.oclIsDefined or xml.oclIsDefined or hasReference
def: let hasReference : Boolean = reference.isNotNull

inv "content is required if not null": isNotNull implies
      (hasValue and canonical.oclIsDefined)
inv "only one of value, data, xml": value.oclIsDefined implies
        (data.oclIsUndefined and xml.oclIsUndefined) and
         data.oclIsDefined implies (value.oclIsUndefined and
               xml.oclIsUndefined) and
         xml.oclIsDefined implies (value.oclIsUndefined and
               data.oclIsUndefined)
inv "integrityCheckAlgorithm required": integrityCheck.oclIsDefined
           = integrityCheckAlgorithm.oclIsDefined
inv "thumbnails do not use reference": thumbnail.isNotNull implies
           not thumbnail.hasReference
inv "thumbnails do not have thumbnails": thumbnail.isNotNull
           implies thumbnail.thumbnail.oclIsUndefined
inv "compression only on binary": compression.oclIsDefined implies
           (data.oclIsDefined or reference.oclIsDefined)
inv "mediaType cannot be null": isNotNull implies
           mediaType.oclIsDefined
inv "value implies mediaType is text/plain": value.oclIsDefined
           implies mediaType = 'text/plain'
inv "no charset for value or xml": (value.oclIsDefined or
           xml.oclIsDefined) implies charset.oclIsUndefined
inv "No History or Update Mode": noUpdateOrHistory(reference) and
           noUpdateOrHistory(thumbnail)
   inv "no nested translations": translation->forAll(t |
          t.translation->isEmpty)
```


Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06

7.4.2.6 Operations

7.4.2.6.1 **canonical : Binary**: The sequence of bytes that is the actual data.

NOTE 1 This sequence of bytes is retrieved from the reference if one was provided, and decompressed if appropriate.

NOTE 2 This operation does not follow the normal rules for operations and nullFlavors because the return type cannot have a nullFlavor.

NOTE 3 The result is null if the ED has a nullFlavor.

7.4.2.7 Examples

7.4.2.7.1 Plain text

```
<example xsi:type="ED" value="this is plain text"
    mediaType="text/plain"/>
<example xsi:type="ED" value="this is plain text"/>
```

This ED is a simple representation of plain text. The mediaType is specified as text/plain. Since the default value for mediatype is text/plain, the mediaType does not need to be represented in the XML, and the second example is also valid. The character set is not specified and in this case, the character set shall not be present; the character set matches the encoding of the XML.

7.4.2.7.2 Language

```
<example xsi:type="ED" flavorId="ED.TEXT" value="this is plain text"
language="en" mediaType="text/plain"/>
<example xsi:type="ED" flavorId="ED.TEXT" value="dieses ist normaler Text"
language="de" mediaType="text/plain"/>
```

This explicitly notes that the language is English in the first case and German in the second case. If there is no language attribute, then the language is unknown, though it is usually safe to assume that the locally predominant language is appropriate. The type is assigned the flavor ED.TEXT – a text only flavor of ED.

```
<example xsi:type="ED" value="this is plain text" language="en">
  <translation xsi:type="ED" value="dieses ist normaler Text"
    language="de"/>
</example>
```

Here, the German version is contained as a translation of the English version.

```
<example xsi:type="ED" value="this is plain text" language="en-ca">
  <translation xsi:type="ED" value="ce ci est du texte non structuré"
    language="fr-ca"/>
</example>
```

Localized languages may also be used, as in this French Canadian example.

7.4.2.7.3 Binary content

```
<example xsi:type="ED">
  <data>dGhpcyBpcyBiaW5hcnkgY29udGVudA==</data>
</example>
```

This ED contains the plain text "this is binary content". Since the mediaType is text/plain, the mediaType attribute does not need to be populated as this is its default value.

Documents
Document Sharing Hub
Full document details: DIN EN ISO 21090:2011-06