DICOM Correction Proposal

STATUS	Final Text
Date of Last Update	2023-01-21
Person Assigned	Clunie
Submitter Name	Jörg Riesmeier <dicom@jriesmeier.com>
Submission Date	2022-03-24

Correction Number	CP-2241
Log Summary: Fix wrong reference to XML Schema for syntax of the Native DICOM Model	
Name of Standard PS3.19 2022e	
Rationale for Correction: The syntax of the Native DICOM Model in PS3.19 Section A.1.6 is not specified by an XML Schema but by a Relax NG Compact scheme (see e.g. Section A.2.6 for comparison). This wrong reference should be fixed.	
Correction Wording:	

Amend PS3.19 Chapter 2

2 Normative References

The following standards contain provisions that, through reference in this text, constitute provisions of this Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this Standard are encouraged to investigate the possibilities of applying the most recent editions of the standards indicated below.

[ISO/IEC Directives, Part 2] ISO/IEC. 2016/05. 7.0. *Rules for the structure and drafting of International Standards*. http://www.iec.ch/members_experts/refdocs/iec/isoiecdir-2%7Bed7.0%7Den.pdf .

[ISO 8822] ISO. 1988. *Information processing systems - Open Systems Interconnection - Connection oriented presentation service definition*.

[WSDL 1.1] W3C. 26 June 2007. *W3C Recommendation Web Services Description Language (WSDL) 1.1*. http://www.w3.org/TR/wsdl .

[XSD1] XML Schema Part 1: Structures Second Edition. W3C Recommendation 28 October 2004 https://www.w3.org/TR/xmlschema-1/

[XPath 2.0] W3C. October 2016. *W3C Recommendation XML Path Language (XPath) 2.0*. http://www.w3.org/TR/xpath20/ .

[InfoSet] W3C. 4 February 2004. *W3C Recommendation XML Information Set*. http://www.w3.org/TR/xml-infoset/ .

IETF RFC2045,2046,2048 MIME Multipurpose Internet Mail Extension

IETF RFC3240 application/dicom MIME Sub-type Registration

IETF RFC3986 Uniform Resource Identifiers (URI) : Generic Syntax

ISO/IEC 19757 DSDL Document Schema Definition Languages (DSDL)

ITU-T Recommendation X.667 UUID (also IETF RFC4122)

For reference see PS3.19 Chapter 4

4 Symbols and Abbreviations

The following symbols and abbreviations are used in this Part of the Standard.

…

XSD XML Schema Definition

…

Amend PS3.19 Chapter 10

10 Data Exchange Model Conventions

Models that can be used by the model-based DataExchange interface methods are defined in Annex A. These models are defined in the form of XML Schemas written in Relax NG Compact form of DSDL as specified by ISO/IEC 19757.

> Note
>
> An implementer may translate the Relax NG Compact form to other forms for use within their implementation as long as the exchanged XML Infosets will validate against the schema specified by the Standard. For example, a particular implementation may internally utilize a schema with stronger validation rules (e.g., using Schematron rules as specified in ISO/IEC 19757, or using **W3C XML Scheme Definition (**XSD~~L~~**) Language [XSD1]** with assertion rules) as long as the XML produced for exchange over the interface can be parsed with the schema specified in the Standard, and that XML from well-formed DICOM objects produced by the schema specified in the Standard can still be utilized by the implementation's internal schema.

Actual instances of the models are XML Infosets. Annex A follows the following conventions in describing models.

> Note
>
> 1. The models are defined via XML schemas to allow the use of commonly available tools to manipulate and navigate the model. For example, an XPath statement can be used to identify portions of interest within the model such as a particular DICOM Attribute and extract it.
> 2. Some implementations may parse directly from the incoming object, which may not be in XML form, into an internal representation, such as the domain object model (DOM) without ever having converted the object to XML.

Within each model description is a table of XML Elements and XML Attributes used to describe an XML Infoset of that model. These tables utilize the following conventions:

1. XML Element names (listed in the first column) are in CamelCase, with the first letter capitalized.
2. XML Attribute names (listed in the first column) are in camelCase with the first letter in lower case.
3. XML Element and XML Attribute names with a set of ">" characters in front of them are nested within the first preceding XML Element with one fewer ">" characters in front of its name. A nested XML Attribute is associated with the immediately enclosing XML Element.
4. The entries in the "Optionality" column have the following interpretations:
    - "R" indicates that the XML Element or XML Attribute is required in all models.
    - "C" indicates that the XML Element or XML Attribute is required in all models if the condition stated in the Description is met.
    - "O" indicates that the XML Element or XML Attribute is optional.
    - If the XML Element or XML Attribute is nested inside another XML Element, and that enclosing XML Element is not present (i.e., it is defined with an Optionality of "O" and is not present in the XML Infoset, or it is defined with

an Optionality of "RC" and is not included in the XML Infoset because the condition was not met), then the nested XML Element or XML Attribute shall not be present in the XML Infoset irrespective of its optionality.

5. The entries in the "Cardinality" column have the following interpretations:

   - "A" indicates that this is represented as an XML Attribute instead of an XML Element, hence has a cardinality of 1 by definition.
   - "1" indicates that only a single instance of this XML Element is included inside the immediately enclosing XML Element, or at the top level if this XML Element is not nested inside another XML Element.
   - "0-n" indicates that zero to n instances of this XML Element are included inside the immediately enclosing XML Element, or at the top level if this XML Element is not nested inside another XML Element.
   - "1-n" indicates that one to n instances of this XML Element are included inside the immediately enclosing XML Element, or at the top level if this XML Element is not nested inside another XML Element.

6. Sets of XML Elements and XML Attributes that are often repeated within models may be defined as macros. Macros that may have general applicability are defined in this section. Macros that are unique to a particular model may be defined in the Annex specific that model. When a macro is included within a table, it is as if the contents of the Macro's table were inserted within the table referencing the macro. Any set of ">" characters in front of the directive to include a Macro in the table are prepended to the XML Element and XML Attribute names listed in the Macro's table.

Change PS3.19 Section A.1.6

A.1.6 Schema

The ~~N~~**n**ormative version of the ~~XML~~ **Relax NG Compact** **s**~~S~~chema for the Native DICOM Model follows:

```
default namespace="http://dicom.nema.org/PS3.19/models/NativeDICOM"

# This schema was created as an intermediary, a means of describing
# native binary encoded DICOM objects as XML Infosets, thus allowing
# one to manipulate binary DICOM objects using familiar XML tools.
# As such, the schema is designed to facilitate a simple, mechanical,
# bi-directional translation between binary encoded DICOM and XML-like
# constructs without constraints, and to simplify identifying portions
# of a DICOM object using XPath statements.
#
# Since this schema has minimal type checking, it is neither intended
# to be used for any operation that involves hand coding, nor to
# describe a definitive, fully validating encoding of DICOM concepts
# into XML, as what one might use, for example, in a robust XML
# database system or in XML-based forms, though it may be used
# as a means for translating binary DICOM Objects into such a form
# (e.g., through an XSLT script).

start = element NativeDicomModel { DicomDataSet }

# A DICOM Data Set is as defined in PS3.5.  It does not appear
# as an XML Element, since it does not appear in the binary encoded
# DICOM objects.  It exists here merely as a documentation aid.
DicomDataSet = DicomAttribute*

DicomAttribute = element DicomAttribute {
  Tag, VR, Keyword?, PrivateCreator?,
  (BulkData | Value+ | Item+ | PersonName+ | InlineBinary)?
}
BulkData = element BulkData{ UUID | URI }
Value = element Value { Number, xsd:string }
InlineBinary = element InlineBinary { xsd:base64Binary }
Item = element Item { Number, DicomDataSet }
PersonName = element PersonName {
  Number,
  element Alphabetic  { NameComponents }?,
  element Ideographic { NameComponents }?,
  element Phonetic    { NameComponents }?
}

NameComponents =
  element FamilyName {xsd:string}?,
  element GivenName  {xsd:string}?,
  element MiddleName {xsd:string}?,
  element NamePrefix {xsd:string}?,
  element NameSuffix {xsd:string}?

# keyword is the attribute tag from PS3.6
# (derived from the DICOM Attribute's name)
Keyword = attribute keyword { xsd:token }
# canonical XML definition of Hex, with lowercase letters disallowed
Tag = attribute tag { xsd:string{ minLength="8" maxLength="8" pattern="[0-9A-F]{8}" }
}
VR = attribute vr { "AE" | "AS" | "AT"| "CS" | "DA" | "DS" | "DT" | "FL" | "FD"
                    | "IS" | "LO" | "LT" | "OB" | "OD" | "OF" | "OL" | "OV" | | "OW" |
"PN" | "SH" | "SL"
                    | "SQ" | "SS" | "ST" | "SV" | "TM" | "UC" | "UI" | "UL" | "UN" |
"UR" | "US" | "UT" | "UV" }
PrivateCreator = attribute privateCreator{ xsd:string }
UUID = attribute uuid { xsd:string }
URI = attribute uri { xsd:anyURI }
Number = attribute number { xsd:positiveInteger }
```