DICOM Correction Proposal

Status	Assigned
Date of Last Update	2015/03/18
Person Assigned	Clunie
Submitter Name	Harry Solomon
Submission Date	2015/03/12

Correction Number CP-1481
Log Summary: Update PS3.19 for expanded Code Sequence
Name of Standard PS3.19 2015a
Rationale for Correction: Use of the Code Sequence in PS3.19 definitions needs to be expanded to support the new definition of CP1031.
Correction Wording:

10.1 Coded Terminology

Models may make use of coded terminology. The representation of coded terminology in DICOM is described in PS3.3. Specific terminology of interest, organized in Context Groups in PS3.16, can be referenced using the following macro.

Table 10.1-1. Coded Terminology Macro

Name	Optionality	Cardinality	Description
BASIC CODED ENTRY ATTRIBUTES			
CodeValue	~~R~~**C**	1	The particular code value identifying the referenced term or concept. **Required if LongCodeValue or URNCodeValue are not present.** See Section 8.1 in PS3.3.
CodingSchemeDesignator	R	1	Designates the coding scheme in which the codeValue is defined. See Section 8.2 in PS3.3.
CodingSchemeVersion	C	1	See Section 8.2 in PS3.3. Required if the codingSchemeDesignator is not sufficient to identify the codeValue.
CodeMeaning	O	0-1	A brief human readable description of what the coded value means. See Section 8.3 in PS3.3.
LongCodeValue	**C**	**1**	**Required if CodeValue or URNCodeValue are not present.**
URNCodeValue	**C**	**1**	**Required if CodeValue or LongCodeValue are not present.**
EquivalentCodeSequence			

Name	Optionality	Cardinality	Description
ENHANCED ENCODING MODE			
ContextIdentifier	O	0-1	Identifies a Context Group defined within a Mapping Resource from which the values of codeValue and codeMeaning were selected or have been added as a Private Context Group extension See Section 8.6 in PS3.3 and Section 8.7 in PS3.3.
MappingResource	C	1	See Section 8.4 in PS3.3. Required if the contextIdentifier XML Attribute is present.
ContextGroupVersion	C	1	See Section 8.5 in PS3.3. Required if the contextIdentifier XML Attribute is present.
ContextGroupExtensionFlag	O	0-1	Indicates whether the codeValue/codingScheme/codeMeaning is selected from a private extension of the Context Group identified in the contextIdentifier XML Attribute. See Section 8.7 in PS3.3. Enumerated Values: "Y" "N"
ContextGroupLocalVersion	C	1	See Section 8.7 in PS3.3.
ContextGroupExtensionCreatorUID	C	1	Identifies the person or organization who created an extension to the Context Group. See Section 8.7 in PS3.3. Required if the value of contextGroupExtensionFlag is "Y".

A.2.6 Schema

The Relax NG Compact schema for the Abstract Multi-Dimensional Image Model follows:

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

start = AbstractImageDataSet
AbstractImageDataSet =

 element AbstractImageDataSet {
    element Component{
      attribute idNumber { xsd:positiveInteger },
      attribute datatype { ComponentDatatype },
      attribute minValue { xsd:double }?,
      attribute maxValue { xsd:double }?,
      element Semantics { CodedTerm },
      element Unit { CodedTerm  },
      element RealWordMapping {
        attribute rescaleSlope { xsd:double },
        attribute rescaleIntercept { xsd:double },
        element Unit { CodedTerm },
        element Semantics { CodedTerm }
      }*
    }+,
```

```
    element Dimension {
      attribute idNumber { xsd:positiveInteger },
      attribute numberOfSamples { xsd:positiveInteger },
      element Semantics { CodedTerm },
      (element Regular {
         attribute width { xsd:double },
         attribute spacing { xsd:double },
         element Unit { CodedTerm  },
         element AxisDirection { CodedTerm  }?,
         element AxisOrientation { CodedTerm  }?
       }
       | element Irregular {
           element origin { xsd:double },
           element SampleLocation {
             attribute index { xsd:positiveInteger },
             attribute width { xsd:double },
             attribute distanceToOrigin { xsd:double }
           }+,
         element Unit { CodedTerm  },
         element AxisDirection { CodedTerm  }?,
         element AxisOrientation { CodedTerm  }?
         }
       | element Qualitative {
           element Sample {
             attribute index { xsd:positiveInteger },
             element Semantics { CodedTerm }
           }+
         }),
      element Origin {
        attribute index { xsd:positiveInteger }?,
        attribute xCoord { xsd:double },
        attribute yCoord { xsd:double },
        attribute zCoord { xsd:double }
      }*,
      element DirectionCosines {
        attribute concernedSpatialDimension { xsd:positiveInteger },
        attribute index { xsd:positiveInteger }?,
        attribute cosAlongX { xsd:double },
        attribute cosAlongY { xsd:double },
        attribute cosAlongZ { xsd:double }
      }*
    }+,
    element PixelData { DimensionalData },
    element PixelMapOfValidData {
      attribute datatype { PixelMapDatatype },
      (
        attribute inValue { xsd:positiveInteger }
        | attribute outValue { xsd:positiveInteger }
      ),
      DimensionalData
    }?
  }

ComponentDatatype =
    "SIGNED_INT8"
    | "SIGNED_INT16"
    | "SIGNED_INT32"
    | "UNSIGNED_CHAR8"
    | "UNSIGNED_INT16"
    | "UNSIGNED_INT32"
    | "FLOAT32"
```

```
    | "FLOAT64"

PixelMapDatatype =
    "BIT1"
    | "UNSIGNED_INT8"

DimensionalData =
  element DimensionalData {
    attribute dimensionID { xsd:positiveInteger },
    element DataAt
    {
      attribute sampleNumber { xsd:positiveInteger },
      attribute descriptorUUID { xsd:string }?,
      (DimensionalData | BulkDataPointer)
    }+
  }

BulkDataPointer =
    attribute UUID { xsd:string }

CodedTerm =
    element CodeValue { xsd:string },
    element CodingSchemeDesignator { xsd:string },
    element CodingSchemeVersion { xsd:string }?,
    element CodeMeaning { xsd:string }?,
    (
      element ContextIdentifier { xsd:string },
      element MappingResource { xsd:string },
      element ContextGroupVersion { xsd:string }
    )?,
    (
      element ContextGroupExtensionFlag { xsd:string },
      element ContextGroupLocalVersion { xsd:string }?,
      element ContextGroupExtensionCreatorUID { xsd:string }?
    )?
```