XML Schema Documentation

Table of Contents

top

Schema Document Properties

Target Namespace	**http://physics.nist.gov/Divisions/Div846/Gp4/ANSIN4242/2005/ANSIN4242**
Element and Attribute Namespaces	• Global element and attribute declarations belong to this schema's target namespace. • By default, local element declarations belong to this schema's target namespace. • By default, local attribute declarations have no namespace.

Declared Namespaces

Prefix	Namespace
Default namespace	**http://physics.nist.gov/Divisions/Div846/Gp4/ANSIN4242/2005/ANSIN4242**
xml	http://www.w3.org/XML/1998/namespace
n42ns	**http://physics.nist.gov/Divisions/Div846/Gp4/ANSIN4242/2005/ANSIN4242**
xs	http://www.w3.org/2001/XMLSchema

Schema Component Representation

```
<xs:schema
targetNamespace="http://physics.nist.gov/Divisions/Div846/Gp4/ANSIN4242/2005/ANSI
elementFormDefault="qualified" id="n42">
   ...
</xs:schema>
```

top

Global Declarations

Element: N42InstrumentData

Name	N42InstrumentData
Type	N42InstrumentData
Nillable	no
Abstract	no
Documentation	root element
Diagram	

N42InstrumentData
Type N42InstrumentData

XML Instance Representation

```
<N42InstrumentData>
   <Remark> ... </Remark> [0..1]
   <Measurement> Measurement </Measurement> [1..*] ?
   <Calibration> Calibration </Calibration> [0..*] ?
   Allow any elements from a namespace other than this schema's
   namespace (lax validation). [0..*]
</N42InstrumentData>
```

Schema Component Representation

```
<xs:element name="N42InstrumentData" type="N42InstrumentData"/>
```

top

Element: Remark

Name	Remark
Type	xs:string
Nillable	no
Abstract	no
Documentation	Remark (i.e., comment)
Diagram	

Remark
Type xs:string

XML Instance Representation

```
<Remark> xs:string </Remark>
```

Schema Component Representation

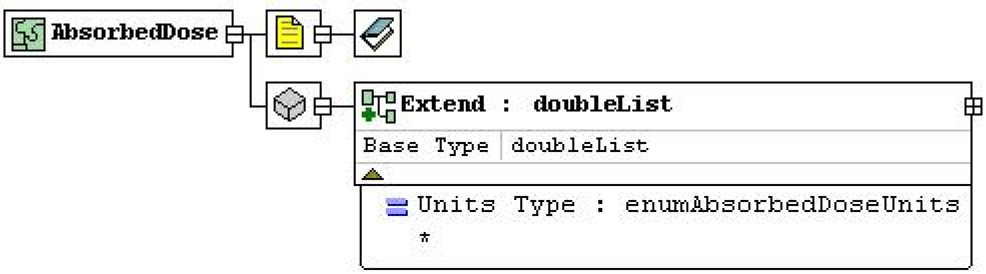
```
<xs:element name="Remark" type="xs:string"/>
```

top

Global Definitions

Complex Type: AbsorbedDose

Super-types:	doubleList (by restriction) < **AbsorbedDose** (by extension)
Sub-types:	None

Name	AbsorbedDose
Abstract	no
Documentation	Absorbed dose rate
Diagram	

XML Instance Representation

```
<...
 Units="enumAbsorbedDoseUnits [1] ?"
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
   doubleList
</...>
```

Schema Component Representation

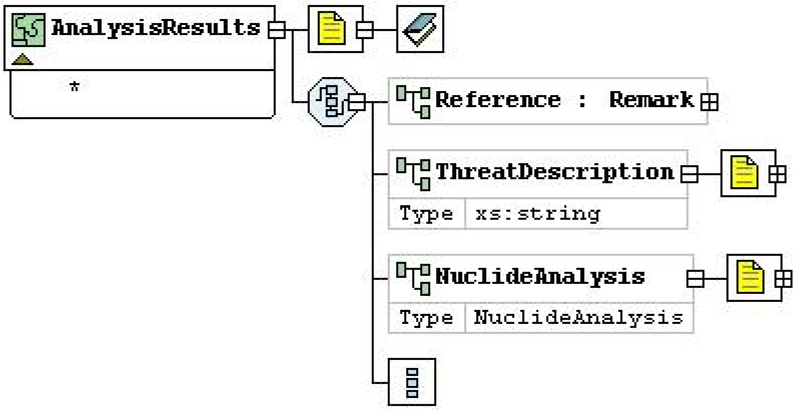
```
<xs:complexType name="AbsorbedDose">
   <xs:simpleContent>
      <xs:extension base="doubleList">
         <xs:attribute name="Units" type="enumAbsorbedDoseUnits"
         use="required"/>
         <xs:anyAttribute namespace="##other" processContents="lax"/>
      </xs:extension>
   </xs:simpleContent>
</xs:complexType>
```

top

Complex Type: AnalysisResults

Super-types:	None
Sub-types:	None

Name	AnalysisResults
Abstract	no
Documentation	Results of data analysis of a measurement
Diagram	



XML Instance Representation

```
<...
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
   <Remark> ... </Remark> [0..1]
   <ThreatDescription> xs:string </ThreatDescription> [0..1] ?
   <NuclideAnalysis> NuclideAnalysis </NuclideAnalysis> [0..1] ?
   Allow any elements from a namespace other than this schema's
   namespace (lax validation). [0..*]
</...>
```

Schema Component Representation

```
<xs:complexType name="AnalysisResults">
   <xs:sequence>
      <xs:element ref="Remark" minOccurs="0"/>
      <xs:element name="ThreatDescription" type="xs:string"
      minOccurs="0"/>
      <xs:element name="NuclideAnalysis" type="NuclideAnalysis"
      minOccurs="0"/>
      <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
      processContents="lax"/>
   </xs:sequence>
   <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

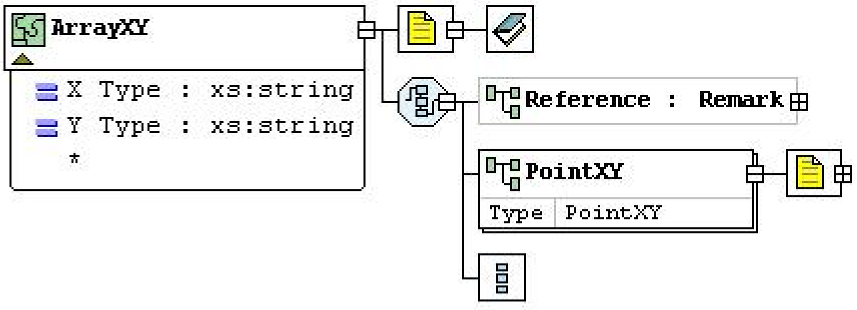
top

Complex Type: ArrayXY

Super-types:	None
Sub-types:	None

Name	ArrayXY
Abstract	no
Documentation	Array of ordinal pairs

Diagram



XML Instance Representation

```
<...
 X="xs:string [0..1] ?"
 Y="xs:string [0..1] ?"
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
   <Remark> ... </Remark> [0..1]
   <PointXY> PointXY </PointXY> [1..*] ?
   Allow any elements from a namespace other than this schema's
   namespace (lax validation). [0..*]
</...>
```

Schema Component Representation

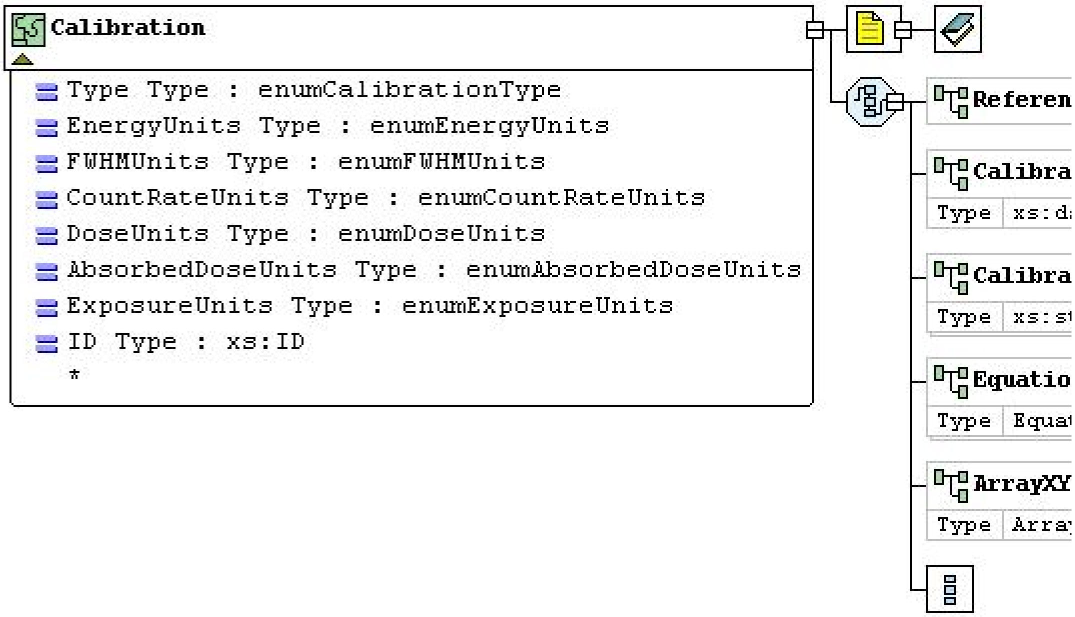
```
<xs:complexType name="ArrayXY">
   <xs:sequence>
      <xs:element ref="Remark" minOccurs="0"/>
      <xs:element name="PointXY" type="PointXY" minOccurs="1"
      maxOccurs="unbounded"/>
      <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
      processContents="lax"/>
   </xs:sequence>
   <xs:attribute name="X" type="xs:string"/>
   <xs:attribute name="Y" type="xs:string"/>
   <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

top

Complex Type: Calibration

Super-types:	None
Sub-types:	None

Name	Calibration
Abstract	no
Documentation	Calibration data
Diagram	


Calibration
Type Type : enumCalibrationType
EnergyUnits Type : enumEnergyUnits
FWHMUnits Type : enumFWHMUnits
CountRateUnits Type : enumCountRateUnits
DoseUnits Type : enumDoseUnits
AbsorbedDoseUnits Type : enumAbsorbedDoseUnits
ExposureUnits Type : enumExposureUnits
ID Type : xs:ID
*
Referen
Calibra
Type xs:d
Calibra
Type xs:s
Equatio
Type Equat
ArrayXY
Type Array

XML Instance Representation

```
<...
 Type="enumCalibrationType [1]  ? "
 EnergyUnits="enumEnergyUnits [0..1]  ? "
 FWHMUnits="enumFWHMUnits [0..1]  ? "
 CountRateUnits="enumCountRateUnits [0..1]  ? "
 DoseUnits="enumDoseUnits [0..1]  ? "
 AbsorbedDoseUnits="enumAbsorbedDoseUnits [0..1]  ? "
 ExposureUnits="enumExposureUnits [0..1]  ? "
 ID="xs:ID [0..1]"
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
   <Remark> ... </Remark> [0..1]
   <CalibrationCreationDate> xs:dateTime </CalibrationCreationDate>
   [0..1]  ?
   <CalibrationMeasurementUUID> xs:string </CalibrationMeasurementUUID>
   [0..*]  ?
   <Equation> Equation </Equation> [0..*]  ?
   <ArrayXY> ArrayXY </ArrayXY> [0..1]  ?
   Allow any elements from a namespace other than this schema's
   namespace (lax validation). [0..*]
</...>
```

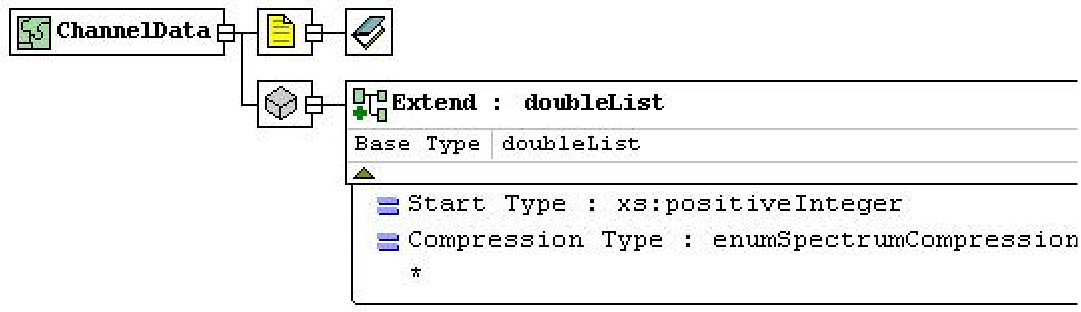
Schema Component Representation

```
<xs:complexType name="Calibration">
   <xs:sequence>
      <xs:element ref="Remark" minOccurs="0"/>
      <xs:element name="CalibrationCreationDate" type="xs:dateTime"
      minOccurs="0"/>
      <xs:element name="CalibrationMeasurementUUID" type="xs:string"
      minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="Equation" type="Equation" minOccurs="0"
      maxOccurs="unbounded"/>
      <xs:element name="ArrayXY" type="ArrayXY" minOccurs="0"/>
      <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
      processContents="lax"/>
   </xs:sequence>
   <xs:attribute name="Type" type="enumCalibrationType"
   use="required"/>
   <xs:attribute name="EnergyUnits" type="enumEnergyUnits"/>
   <xs:attribute name="FWHMUnits" type="enumFWHMUnits"/>
   <xs:attribute name="CountRateUnits" type="enumCountRateUnits"/>
   <xs:attribute name="DoseUnits" type="enumDoseUnits"/>
   <xs:attribute name="AbsorbedDoseUnits"
   type="enumAbsorbedDoseUnits"/>
   <xs:attribute name="ExposureUnits" type="enumExposureUnits"/>
   <xs:attribute name="ID" type="xs:ID"/>
   <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

top

Complex Type: ChannelData

Super-types:	doubleList (by restriction) < **ChannelData** (by extension)
Sub-types:	None

Name	ChannelData
Abstract	no
Documentation	The spectrum: counts vs. channel for the entire spectrum or a region of interest
Diagram	 ChannelData Extend : doubleList Base Type doubleList Start Type : xs:positiveInteger Compression Type : enumSpectrumCompression *

XML Instance Representation

```
<...
 Start="xs:positiveInteger [0..1] ?"
 Compression="enumSpectrumCompressionType [0..1] ?"
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
   doubleList
</...>
```

Schema Component Representation

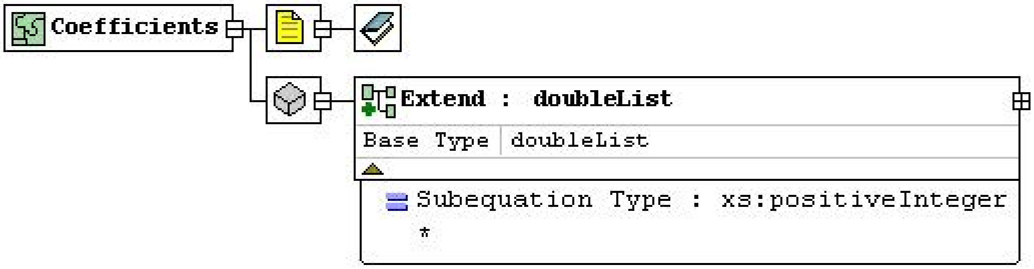
```
<xs:complexType name="ChannelData">
   <xs:simpleContent>
```

```
      <xs:extension base="doubleList">
        <xs:attribute name="Start" type="xs:positiveInteger"/>
        <xs:attribute name="Compression"
        type="enumSpectrumCompressionType"/>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
      </xs:extension>
    </xs:simpleContent>
</xs:complexType>
```

top

Complex Type: Coefficients

Super-types:	doubleList (by restriction) < **Coefficients** (by extension)
Sub-types:	None

Name	Coefficients
Abstract	no
Documentation	Equation coefficients
Diagram	 Coefficients Extend : doubleList Base Type doubleList Subequation Type : xs:positiveInteger *

XML Instance Representation

```
<...
 Subequation="xs:positiveInteger [0..1] ?"
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
   doubleList
</...>
```

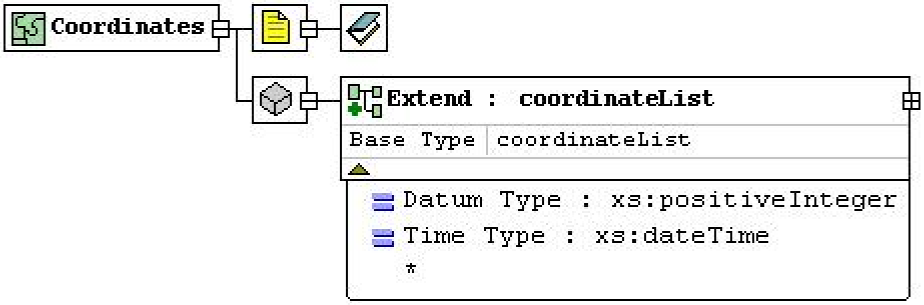
Schema Component Representation

```
<xs:complexType name="Coefficients">
   <xs:simpleContent>
      <xs:extension base="doubleList">
         <xs:attribute name="Subequation" type="xs:positiveInteger"/>
         <xs:anyAttribute namespace="##other" processContents="lax"/>
      </xs:extension>
   </xs:simpleContent>
</xs:complexType>
```

top

Complex Type: Coordinates

Super-types:	decimalList (by restriction) < coordinateList (by restriction) < **Coordinates** (by extension)
Sub-types:	None

Name	Coordinates
Abstract	no
Documentation	Geographical coordinates (latitude, longitude, optional elevation)
Diagram	 Coordinates Extend : coordinateList Base Type coordinateList Datum Type : xs:positiveInteger Time Type : xs:dateTime *

XML Instance Representation

```
<...
 Datum="xs:positiveInteger [0..1] ?"
 Time="xs:dateTime [0..1] ?"
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
   coordinateList
</...>
```

Schema Component Representation

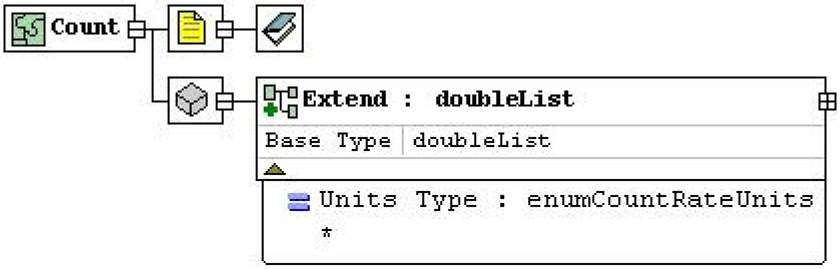
```
<xs:complexType name="Coordinates">
   <xs:simpleContent>
      <xs:extension base="coordinateList">
         <xs:attribute name="Datum" type="xs:positiveInteger"/>
         <xs:attribute name="Time" type="xs:dateTime"/>
         <xs:anyAttribute namespace="##other" processContents="lax"/>
      </xs:extension>
   </xs:simpleContent>
</xs:complexType>
```

top

Complex Type: Count

Super-types:	doubleList (by restriction) < **Count** (by extension)
Sub-types:	None

Name	Count

Abstract	no
Documentation	Count rate
Diagram	

XML Instance Representation

```
<...
 Units="enumCountRateUnits [1] ? "
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
   doubleList
</...>
```

Schema Component Representation

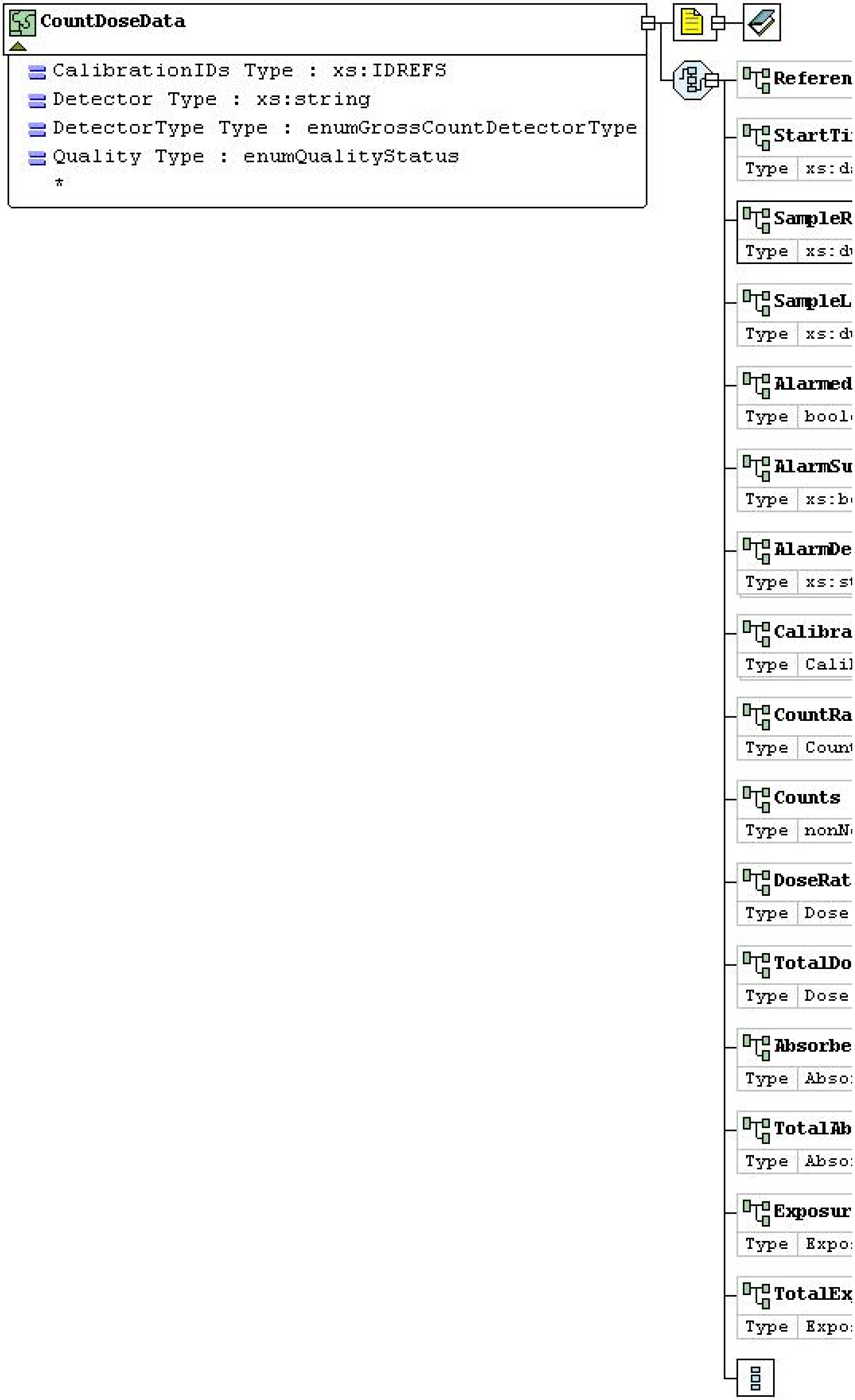
```
<xs:complexType name="Count">
   <xs:simpleContent>
      <xs:extension base="doubleList">
         <xs:attribute name="Units" type="enumCountRateUnits"
         use="required"/>
         <xs:anyAttribute namespace="##other" processContents="lax"/>
      </xs:extension>
   </xs:simpleContent>
</xs:complexType>
```

top

Complex Type: CountDoseData

Super-types:	None
Sub-types:	None

Name	CountDoseData
Abstract	no
Documentation	Data measured by a survey meter or personal radiation detector
Diagram	


CountDoseData
CalibrationIDs Type : xs:IDREFS
Detector Type : xs:string
DetectorType Type : enumGrossCountDetectorType
Quality Type : enumQualityStatus
*
Referen
StartTi
Type
xs:d
SampleR
Type
xs:d
SampleL
Type
xs:d
Alarmed
Type
bool
AlarmSu
Type
xs:b
AlarmDe
Type
xs:s
Calibra
Type
Cali
CountRa
Type
Coun
Counts
Type
nonN
DoseRat
Type
Dose
TotalDo
Type
Dose
Absorbe
Type
Abso
TotalAb
Type
Abso
Exposur
Type
Expo
TotalEx
Type
Expo

XML Instance Representation

```
<...
 CalibrationIDs="xs:IDREFS [0..1] ?"
 Detector="xs:string [0..1] ?"
 DetectorType="enumGrossCountDetectorType [0..1] ?"
 Quality="enumQualityStatus [0..1] ?"
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
   <Remark> ... </Remark> [0..1]
   <StartTime> xs:dateTime </StartTime> [0..1] ?
   <SampleRealTime> xs:duration </SampleRealTime> [1] ?
   <SampleLiveTime> xs:duration </SampleLiveTime> [0..1] ?
   <Alarmed> booleanList </Alarmed> [0..1] ?
   <AlarmSummary> xs:boolean </AlarmSummary> [0..1] ?
   <AlarmDescription> xs:string </AlarmDescription> [0..*] ?
   <Calibration> Calibration </Calibration> [0..*] ?
   <CountRate> Count </CountRate> [0..1] ?
   <Counts> nonNegativeIntegerList </Counts> [0..1] ?
   <DoseRate> Dose </DoseRate> [0..1] ?
   <TotalDose> Dose </TotalDose> [0..1] ?
   <AbsorbedDoseRate> AbsorbedDose </AbsorbedDoseRate> [0..1] ?
   <TotalAbsorbedDose> AbsorbedDose </TotalAbsorbedDose> [0..1] ?
   <ExposureRate> Exposure </ExposureRate> [0..1] ?
   <TotalExposure> Exposure </TotalExposure> [0..1] ?
   Allow any elements from a namespace other than this schema's
   namespace (lax validation). [0..*]
</...>
```

Schema Component Representation

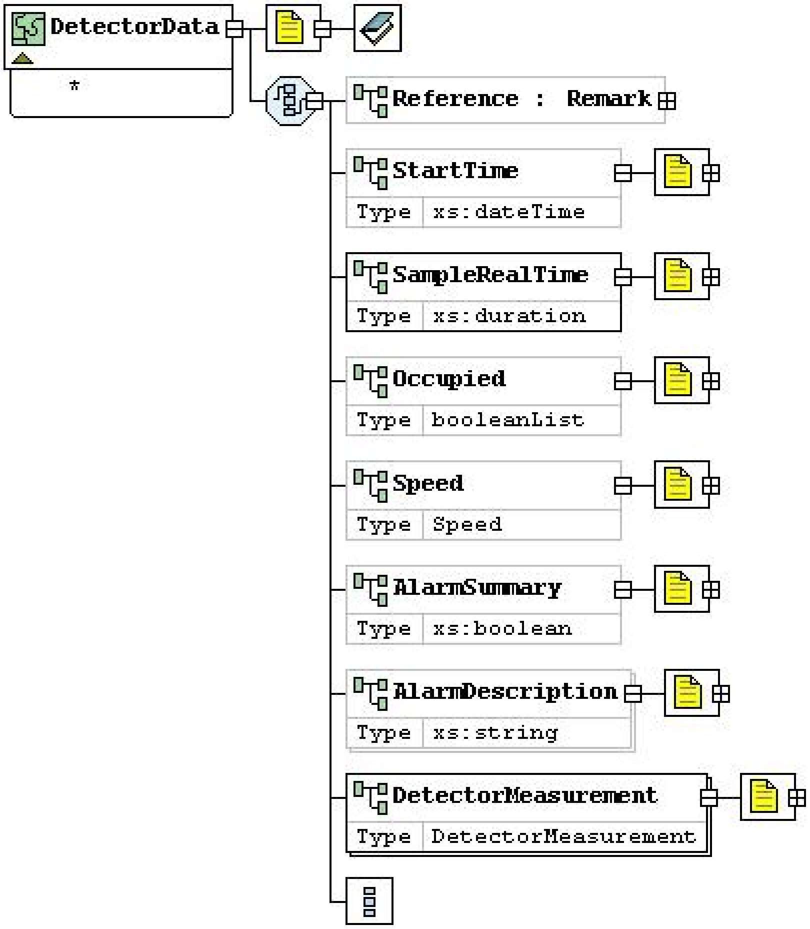
```
<xs:complexType name="CountDoseData">
   <xs:sequence>
      <xs:element ref="Remark" minOccurs="0"/>
      <xs:element name="StartTime" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="SampleRealTime" type="xs:duration"/>
      <xs:element name="SampleLiveTime" type="xs:duration"
      minOccurs="0"/>
      <xs:element name="Alarmed" type="booleanList" minOccurs="0"/>
      <xs:element name="AlarmSummary" type="xs:boolean" minOccurs="0"/>
      <xs:element name="AlarmDescription" type="xs:string"
      minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="Calibration" type="Calibration" minOccurs="0"
      maxOccurs="unbounded"/>
      <xs:element name="CountRate" type="Count" minOccurs="0"/>
      <xs:element name="Counts" type="nonNegativeIntegerList"
      minOccurs="0"/>
      <xs:element name="DoseRate" type="Dose" minOccurs="0"/>
      <xs:element name="TotalDose" type="Dose" minOccurs="0"/>
      <xs:element name="AbsorbedDoseRate" type="AbsorbedDose"
      minOccurs="0"/>
      <xs:element name="TotalAbsorbedDose" type="AbsorbedDose"
      minOccurs="0"/>
      <xs:element name="ExposureRate" type="Exposure" minOccurs="0"/>
      <xs:element name="TotalExposure" type="Exposure" minOccurs="0"/>
      <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
```

```
    processContents="lax"/>
  </xs:sequence>
  <xs:attribute name="CalibrationIDs" type="xs:IDREFS"/>
  <xs:attribute name="Detector" type="xs:string"/>
  <xs:attribute name="DetectorType"
  type="enumGrossCountDetectorType"/>
  <xs:attribute name="Quality" type="enumQualityStatus"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

top

Complex Type: DetectorData

Super-types:	None
Sub-types:	None

Name	DetectorData
Abstract	no
Documentation	Data measured by gross count or spectroscopic portal monitor
Diagram	

XML Instance Representation

```
<...
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
   <Remark> ... </Remark> [0..1]
   <StartTime> xs:dateTime </StartTime> [0..1] ?
   <SampleRealTime> xs:duration </SampleRealTime> [1] ?
   <Occupied> booleanList </Occupied> [0..1] ?
   <Speed> Speed </Speed> [0..1] ?
   <AlarmSummary> xs:boolean </AlarmSummary> [0..1] ?
   <AlarmDescription> xs:string </AlarmDescription> [0..*] ?
   <DetectorMeasurement> DetectorMeasurement </DetectorMeasurement>
   [1..*] ?
   Allow any elements from a namespace other than this schema's
   namespace (lax validation). [0..*]
</...>
```

Schema Component Representation

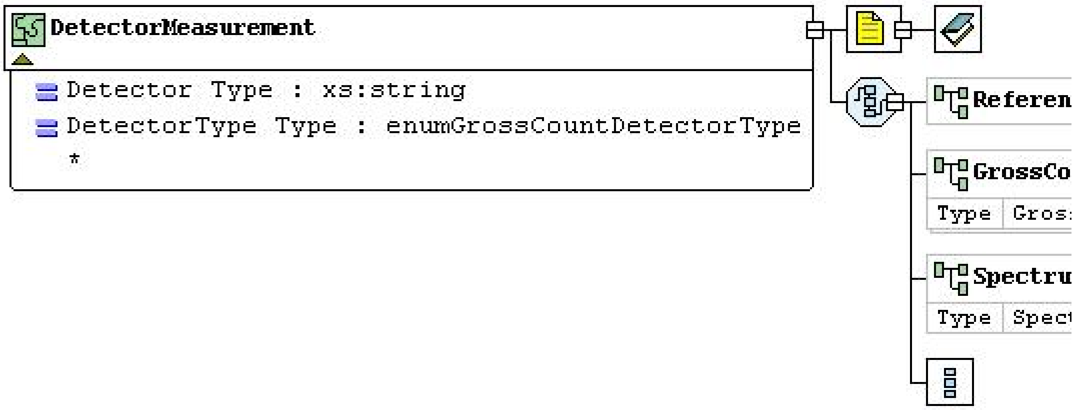
```
<xs:complexType name="DetectorData">
   <xs:sequence>
      <xs:element ref="Remark" minOccurs="0"/>
      <xs:element name="StartTime" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="SampleRealTime" type="xs:duration"/>
      <xs:element name="Occupied" type="booleanList" minOccurs="0"/>
      <xs:element name="Speed" type="Speed" minOccurs="0"/>
      <xs:element name="AlarmSummary" type="xs:boolean" minOccurs="0"/>
      <xs:element name="AlarmDescription" type="xs:string"
      minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="DetectorMeasurement" type="DetectorMeasurement"
      minOccurs="1" maxOccurs="unbounded"/>
      <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
      processContents="lax"/>
   </xs:sequence>
   <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

top

Complex Type: DetectorMeasurement

Super-types:	None
Sub-types:	None

Name	DetectorMeasurement
Abstract	no
Documentation	RPM measurements from one detector
Diagram	


DetectorMeasurement
Detector Type : xs:string
DetectorType Type : enumGrossCountDetectorType
*
Referen
GrossCo
Type
Gros:
Spectru
Type
Spect

XML Instance Representation

```
<...
 Detector="xs:string [1]  ? "
 DetectorType="enumGrossCountDetectorType [1]  ? "
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
   <Remark> ... </Remark> [0..1]
   <GrossCountMeasurement> GrossCountMeasurement
   </GrossCountMeasurement> [0..*]  ?
   <SpectrumMeasurement> SpectrumMeasurement </SpectrumMeasurement>
   [0..1]  ?
   Allow any elements from a namespace other than this schema's
   namespace (lax validation). [0..*]
</...>
```

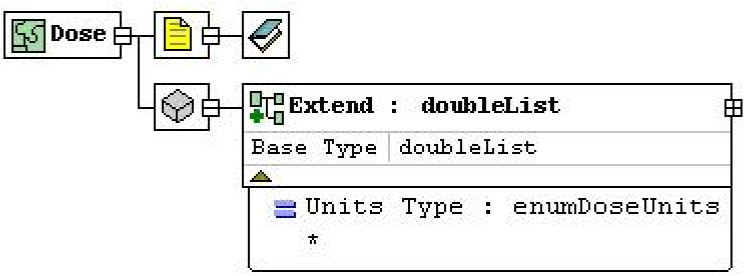
Schema Component Representation

```
<xs:complexType name="DetectorMeasurement">
   <xs:sequence>
      <xs:element ref="Remark" minOccurs="0"/>
      <xs:element name="GrossCountMeasurement"
      type="GrossCountMeasurement" minOccurs="0"
      maxOccurs="unbounded"/>
      <xs:element name="SpectrumMeasurement" type="SpectrumMeasurement"
      minOccurs="0"/>
      <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
      processContents="lax"/>
   </xs:sequence>
   <xs:attribute name="Detector" type="xs:string" use="required"/>
   <xs:attribute name="DetectorType" type="enumGrossCountDetectorType"
   use="required"/>
   <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

top

Complex Type: Dose

Super-types:	doubleList (by restriction) < **Dose** (by extension)
Sub-types:	None

Name	Dose
Abstract	no
Documentation	dose rate
Diagram	 Dose Extend : doubleList Base Type doubleList Units Type : enumDoseUnits *

XML Instance Representation

```
<...
 Units="enumDoseUnits [1] ?"
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
   doubleList
</...>
```

Schema Component Representation

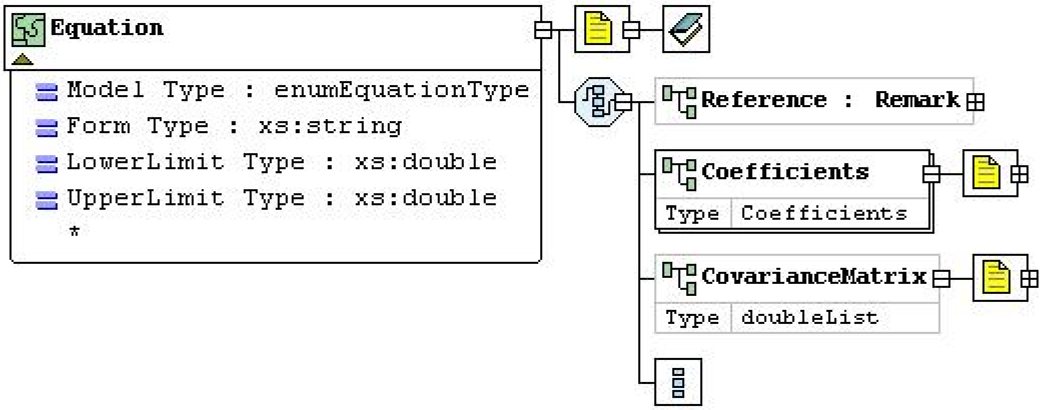
```
<xs:complexType name="Dose">
   <xs:simpleContent>
      <xs:extension base="doubleList">
         <xs:attribute name="Units" type="enumDoseUnits"
         use="required"/>
         <xs:anyAttribute namespace="##other" processContents="lax"/>
      </xs:extension>
   </xs:simpleContent>
</xs:complexType>
```

top

Complex Type: Equation

Super-types:	None
Sub-types:	None

Name	Equation
Abstract	no
Documentation	Encapsulates an equation and its coefficients
Diagram	



XML Instance Representation

```
<...
 Model="enumEquationType [1] ?"
 Form="xs:string [0..1] ?"
 LowerLimit="xs:double [0..1] ?"
 UpperLimit="xs:double [0..1] ?"
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
   <Remark> ... </Remark> [0..1]
   <Coefficients> Coefficients </Coefficients> [1..*] ?
   <CovarianceMatrix> doubleList </CovarianceMatrix> [0..1] ?
   Allow any elements from a namespace other than this schema's
   namespace (lax validation). [0..*]
</...>
```

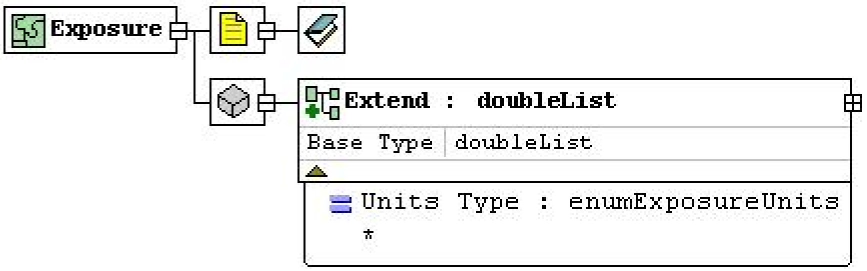
Schema Component Representation

```
<xs:complexType name="Equation">
   <xs:sequence>
      <xs:element ref="Remark" minOccurs="0"/>
      <xs:element name="Coefficients" type="Coefficients"
      maxOccurs="unbounded"/>
      <xs:element name="CovarianceMatrix" type="doubleList"
      minOccurs="0"/>
      <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
      processContents="lax"/>
   </xs:sequence>
   <xs:attribute name="Model" type="enumEquationType" use="required"/>
   <xs:attribute name="Form" type="xs:string"/>
   <xs:attribute name="LowerLimit" type="xs:double"/>
   <xs:attribute name="UpperLimit" type="xs:double"/>
   <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

top

Complex Type: Exposure

Super-types:	doubleList (by restriction) < **Exposure** (by extension)
Sub-types:	None

Name	Exposure
Abstract	no
Documentation	Exposure rate
Diagram	 Exposure Extend : doubleList Base Type doubleList Units Type : enumExposureUnits *

XML Instance Representation

```
<...
 Units="enumExposureUnits [1]  ? "
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
   doubleList
</...>
```

Schema Component Representation

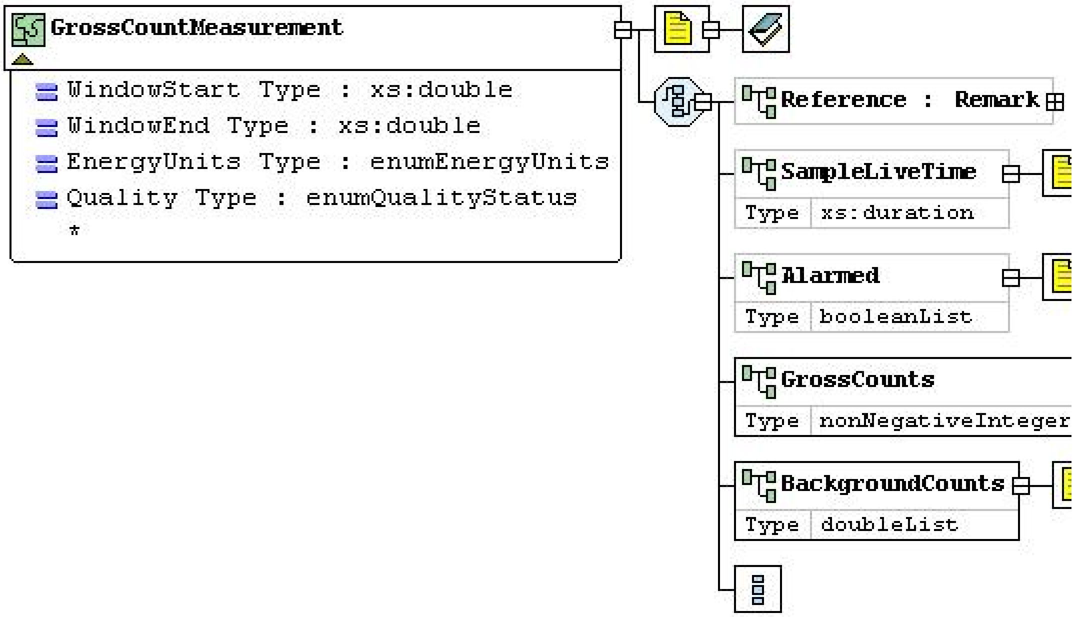
```
<xs:complexType name="Exposure">
   <xs:simpleContent>
      <xs:extension base="doubleList">
         <xs:attribute name="Units" type="enumExposureUnits"
         use="required"/>
         <xs:anyAttribute namespace="##other" processContents="lax"/>
      </xs:extension>
   </xs:simpleContent>
</xs:complexType>
```

top

Complex Type: GrossCountMeasurement

Super-types:	None
Sub-types:	None

Name	GrossCountMeasurement
Abstract	no
Documentation	Gross count portal monitor measurements
Diagram	



XML Instance Representation

```
<...
 WindowStart="xs:double [0..1] ?"
 WindowEnd="xs:double [0..1] ?"
 EnergyUnits="enumEnergyUnits [0..1] ?"
 Quality="enumQualityStatus [0..1] ?"
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
   <Remark> ... </Remark> [0..1]
   <SampleLiveTime> xs:duration </SampleLiveTime> [0..1] ?
   <Alarmed> booleanList </Alarmed> [0..1] ?
   <GrossCounts> nonNegativeIntegerList </GrossCounts> [1] ?
   <BackgroundCounts> doubleList </BackgroundCounts> [1] ?
   Allow any elements from a namespace other than this schema's
   namespace (lax validation). [0..*]
</...>
```

Schema Component Representation

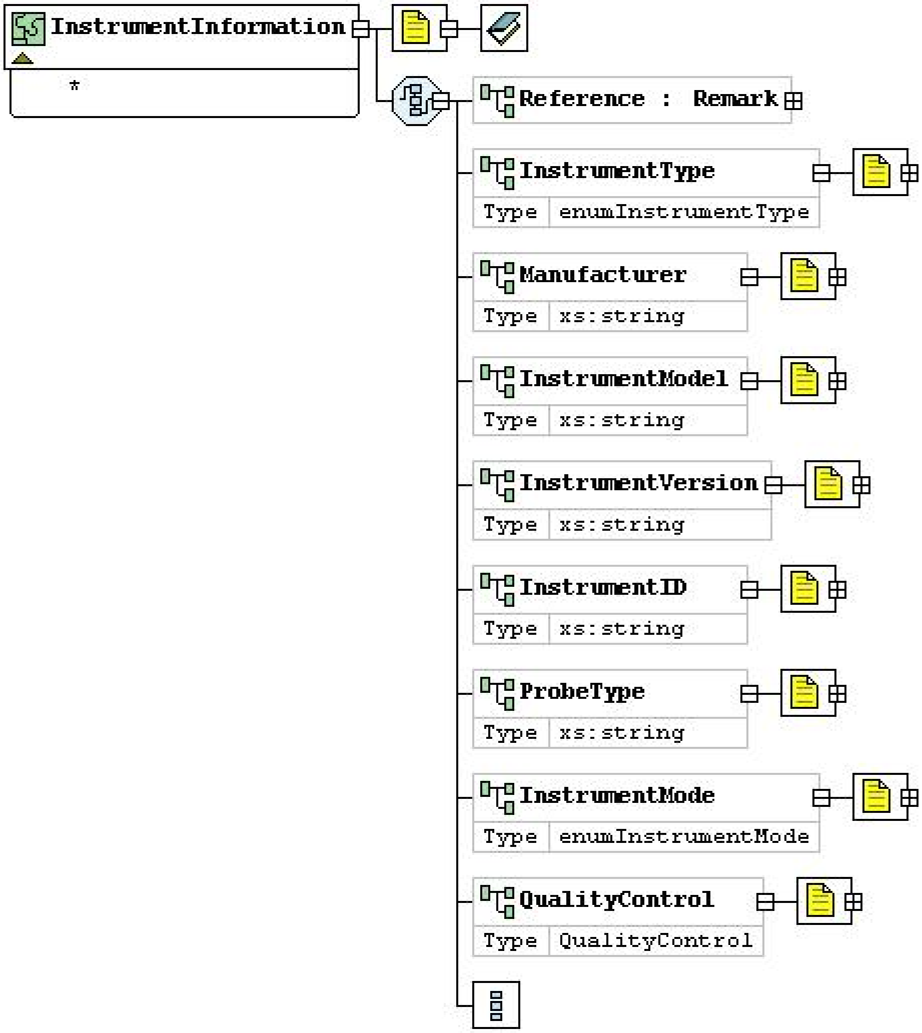
```
<xs:complexType name="GrossCountMeasurement">
  <xs:sequence>
    <xs:element ref="Remark" minOccurs="0"/>
    <xs:element name="SampleLiveTime" type="xs:duration"
    minOccurs="0"/>
    <xs:element name="Alarmed" type="booleanList" minOccurs="0"/>
    <xs:element name="GrossCounts" type="nonNegativeIntegerList"/>
    <xs:element name="BackgroundCounts" type="doubleList"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
    processContents="lax"/>
  </xs:sequence>
  <xs:attribute name="WindowStart" type="xs:double"/>
  <xs:attribute name="WindowEnd" type="xs:double"/>
  <xs:attribute name="EnergyUnits" type="enumEnergyUnits"/>
  <xs:attribute name="Quality" type="enumQualityStatus"/>
```

```
   <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

top

Complex Type: InstrumentInformation

Super-types:	None
Sub-types:	None

Name	InstrumentInformation
Abstract	no
Documentation	Description of the instrument
Diagram	

XML Instance Representation

```
<...
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
```

```
>
  <Remark> ... </Remark> [0..1]
  <InstrumentType> enumInstrumentType </InstrumentType> [0..1] ?
  <Manufacturer> xs:string </Manufacturer> [0..1] ?
  <InstrumentModel> xs:string </InstrumentModel> [0..1] ?
  <InstrumentVersion> xs:string </InstrumentVersion> [0..1] ?
  <InstrumentID> xs:string </InstrumentID> [0..1] ?
  <ProbeType> xs:string </ProbeType> [0..1] ?
  <InstrumentMode> enumInstrumentMode </InstrumentMode> [0..1] ?
  <QualityControl> QualityControl </QualityControl> [0..1] ?
  Allow any elements from a namespace other than this schema's
  namespace (lax validation). [0..*]
</...>
```

Schema Component Representation

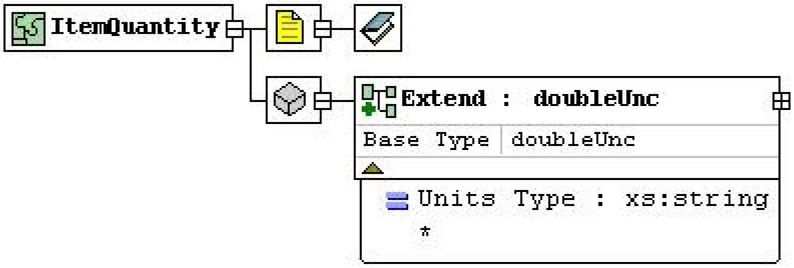
```
<xs:complexType name="InstrumentInformation">
  <xs:sequence>
    <xs:element ref="Remark" minOccurs="0"/>
    <xs:element name="InstrumentType" type="enumInstrumentType"
    minOccurs="0"/>
    <xs:element name="Manufacturer" type="xs:string" minOccurs="0"/>
    <xs:element name="InstrumentModel" type="xs:string"
    minOccurs="0"/>
    <xs:element name="InstrumentVersion" type="xs:string"
    minOccurs="0"/>
    <xs:element name="InstrumentID" type="xs:string" minOccurs="0"/>
    <xs:element name="ProbeType" type="xs:string" minOccurs="0"/>
    <xs:element name="InstrumentMode" type="enumInstrumentMode"
    minOccurs="0"/>
    <xs:element name="QualityControl" type="QualityControl"
    minOccurs="0"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
    processContents="lax"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

top

Complex Type: ItemQuantity

Super-types:	doubleList (by restriction) < doubleUnc (by restriction) < **ItemQuantity** (by extension)
Sub-types:	None

Name	ItemQuantity
Abstract	no
Documentation	Quantity or size of item and the uncertainty in that value
Diagram	



XML Instance Representation

```
<...
 Units="xs:string [1] ?"
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
   doubleUnc
</...>
```

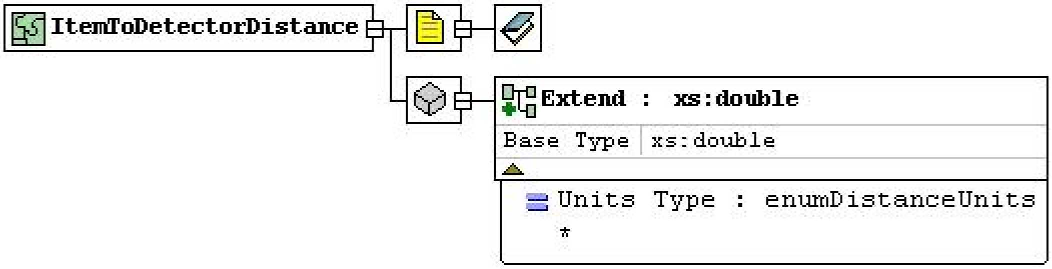
Schema Component Representation

```
<xs:complexType name="ItemQuantity">
  <xs:simpleContent>
    <xs:extension base="doubleUnc">
      <xs:attribute name="Units" type="xs:string" use="required"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

top

Complex Type: ItemToDetectorDistance

Super-types:	xs:double < **ItemToDetectorDistance** (by extension)
Sub-types:	None

Name	ItemToDetectorDistance
Abstract	no
Documentation	Effective item-to-detector distance
Diagram	

XML Instance Representation

```
<...
 Units="enumDistanceUnits [1] ?"
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
   xs:double
</...>
```

Schema Component Representation

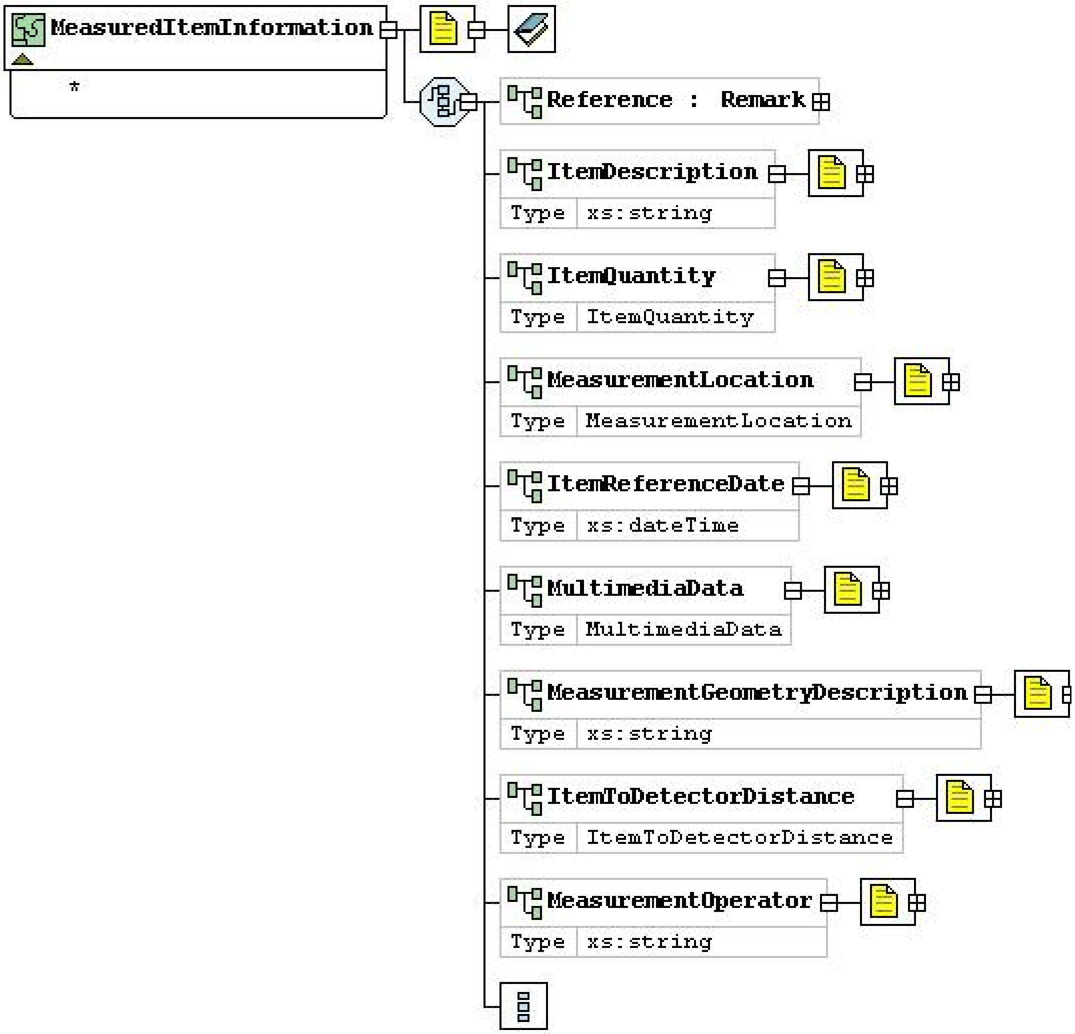
```
<xs:complexType name="ItemToDetectorDistance">
  <xs:simpleContent>
    <xs:extension base="xs:double">
      <xs:attribute name="Units" type="enumDistanceUnits"
      use="required"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

top

Complex Type: MeasuredItemInformation

Super-types:	None
Sub-types:	None

Name	MeasuredItemInformation
Abstract	no
Documentation	Description of measured item and where the measurement was done
Diagram	



XML Instance Representation

```
<...
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
   <Remark> ... </Remark> [0..1]
   <ItemDescription> xs:string </ItemDescription> [0..1] ?
   <ItemQuantity> ItemQuantity </ItemQuantity> [0..1] ?
   <MeasurementLocation> MeasurementLocation </MeasurementLocation>
   [0..1] ?
   <ItemReferenceDate> xs:dateTime </ItemReferenceDate> [0..1] ?
   <MultimediaData> MultimediaData </MultimediaData> [0..1] ?
   <MeasurementGeometryDescription> xs:string
   </MeasurementGeometryDescription> [0..1] ?
   <ItemToDetectorDistance> ItemToDetectorDistance
   </ItemToDetectorDistance> [0..1] ?
   <MeasurementOperator> xs:string </MeasurementOperator> [0..1] ?
   Allow any elements from a namespace other than this schema's
   namespace (lax validation). [0..*]
</...>
```

Schema Component Representation

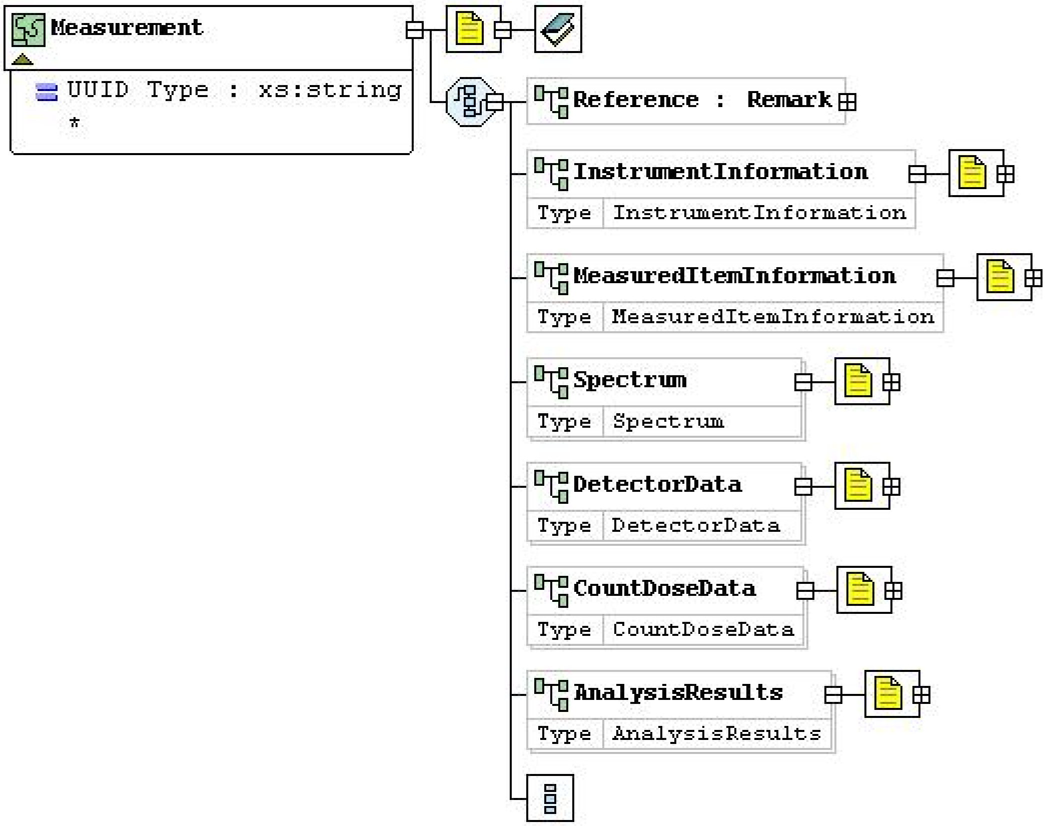
```
<xs:complexType name="MeasuredItemInformation">
  <xs:sequence>
    <xs:element ref="Remark" minOccurs="0"/>
    <xs:element name="ItemDescription" type="xs:string"
    minOccurs="0"/>
    <xs:element name="ItemQuantity" type="ItemQuantity"
    minOccurs="0"/>
    <xs:element name="MeasurementLocation" type="MeasurementLocation"
    minOccurs="0"/>
    <xs:element name="ItemReferenceDate" type="xs:dateTime"
    minOccurs="0"/>
    <xs:element name="MultimediaData" type="MultimediaData"
    minOccurs="0"/>
    <xs:element name="MeasurementGeometryDescription"
    type="xs:string" minOccurs="0"/>
    <xs:element name="ItemToDetectorDistance"
    type="ItemToDetectorDistance" minOccurs="0"/>
    <xs:element name="MeasurementOperator" type="xs:string"
    minOccurs="0"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
    processContents="lax"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

top

Complex Type: Measurement

Super-types:	None
Sub-types:	None

Name	Measurement
Abstract	no
Documentation	A "measurement" from an instrument
Diagram	



XML Instance Representation

```
<...
 UUID="xs:string [0..1] ? "
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
   <Remark> ... </Remark> [0..1]
   <InstrumentInformation> InstrumentInformation
   </InstrumentInformation> [0..1] ?
   <MeasuredItemInformation> MeasuredItemInformation
   </MeasuredItemInformation> [0..1] ?
   <Spectrum> Spectrum </Spectrum> [0..*] ?
   <DetectorData> DetectorData </DetectorData> [0..*] ?
   <CountDoseData> CountDoseData </CountDoseData> [0..*] ?
   <AnalysisResults> AnalysisResults </AnalysisResults> [0..*] ?
   Allow any elements from a namespace other than this schema's
   namespace (lax validation). [0..*]
</...>
```

Schema Component Representation

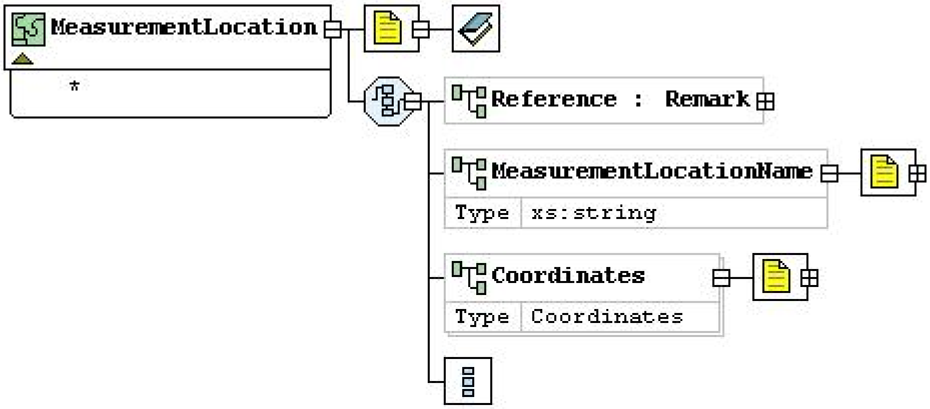
```
<xs:complexType name="Measurement">
   <xs:sequence>
      <xs:element ref="Remark" minOccurs="0"/>
      <xs:element name="InstrumentInformation"
      type="InstrumentInformation" minOccurs="0"/>
      <xs:element name="MeasuredItemInformation"
      type="MeasuredItemInformation" minOccurs="0"/>
      <xs:element name="Spectrum" type="Spectrum" minOccurs="0"
```

```
     maxOccurs="unbounded"/>
     <xs:element name="DetectorData" type="DetectorData" minOccurs="0"
     maxOccurs="unbounded"/>
     <xs:element name="CountDoseData" type="CountDoseData"
     minOccurs="0" maxOccurs="unbounded"/>
     <xs:element name="AnalysisResults" type="AnalysisResults"
     minOccurs="0" maxOccurs="unbounded"/>
     <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
     processContents="lax"/>
  </xs:sequence>
  <xs:attribute name="UUID" type="xs:string"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

top

Complex Type: MeasurementLocation

Super-types:	None
Sub-types:	None

Name	MeasurementLocation
Abstract	no
Documentation	The physical location where the measurement was performed
Diagram	 MeasurementLocation * Reference : Remark MeasurementLocationName Type xs:string Coordinates Type Coordinates

XML Instance Representation

```
<...
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
   <Remark> ... </Remark> [0..1]
   <MeasurementLocationName> xs:string </MeasurementLocationName>
   [0..1] ?
   <Coordinates> Coordinates </Coordinates> [0..*] ?
   Allow any elements from a namespace other than this schema's
   namespace (lax validation). [0..*]
</...>
```

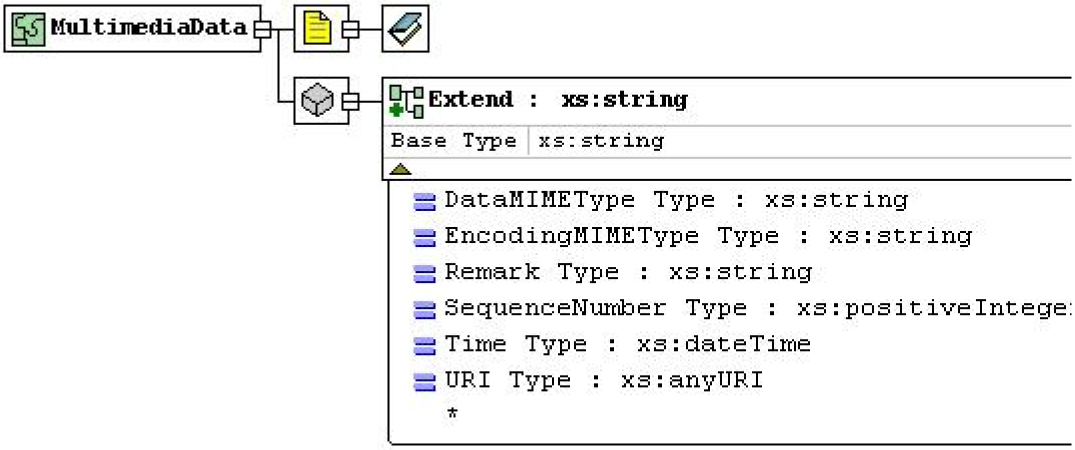
Schema Component Representation

```
<xs:complexType name="MeasurementLocation">
  <xs:sequence>
    <xs:element ref="Remark" minOccurs="0"/>
    <xs:element name="MeasurementLocationName" type="xs:string"
    minOccurs="0"/>
    <xs:element name="Coordinates" type="Coordinates" minOccurs="0"
    maxOccurs="unbounded"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
    processContents="lax"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

top

Complex Type: MultimediaData

Super-types:	xs:string < **MultimediaData** (by extension)
Sub-types:	None

Name	MultimediaData
Abstract	no
Documentation	Multimedia (image, sound, etc.) data
Diagram	

XML Instance Representation

```
<...
 DataMIMEType="xs:string [1] ? "
 EncodingMIMEType="xs:string [1] ? "
 Remark="xs:string [0..1] ? "
 SequenceNumber="xs:positiveInteger [0..1] ? "
 Time="xs:dateTime [0..1] ? "
 URI="xs:anyURI [0..1] ? "
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
```

```
  xs:string
</...>
```

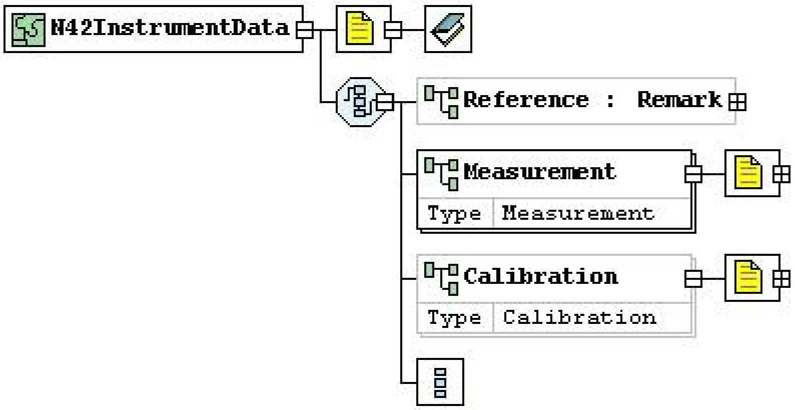
Schema Component Representation

```
<xs:complexType name="MultimediaData">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="DataMIMEType" type="xs:string"
      use="required"/>
      <xs:attribute name="EncodingMIMEType" type="xs:string"
      use="required"/>
      <xs:attribute name="Remark" type="xs:string"/>
      <xs:attribute name="SequenceNumber" type="xs:positiveInteger"/>
      <xs:attribute name="Time" type="xs:dateTime"/>
      <xs:attribute name="URI" type="xs:anyURI"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

top

Complex Type: N42InstrumentData

Super-types:	None
Sub-types:	None

Name	N42InstrumentData
Abstract	no
Documentation	Contains instrument data in the form of Measurement elements
Diagram	 N42InstrumentData Reference : Remark Measurement Type Measurement Calibration Type Calibration

XML Instance Representation

```
<...>
  <Remark> ... </Remark> [0..1]
  <Measurement> Measurement </Measurement> [1..*] ?
  <Calibration> Calibration </Calibration> [0..*] ?
  Allow any elements from a namespace other than this schema's
```

```
   namespace (lax validation). [0..*]
</...>
```

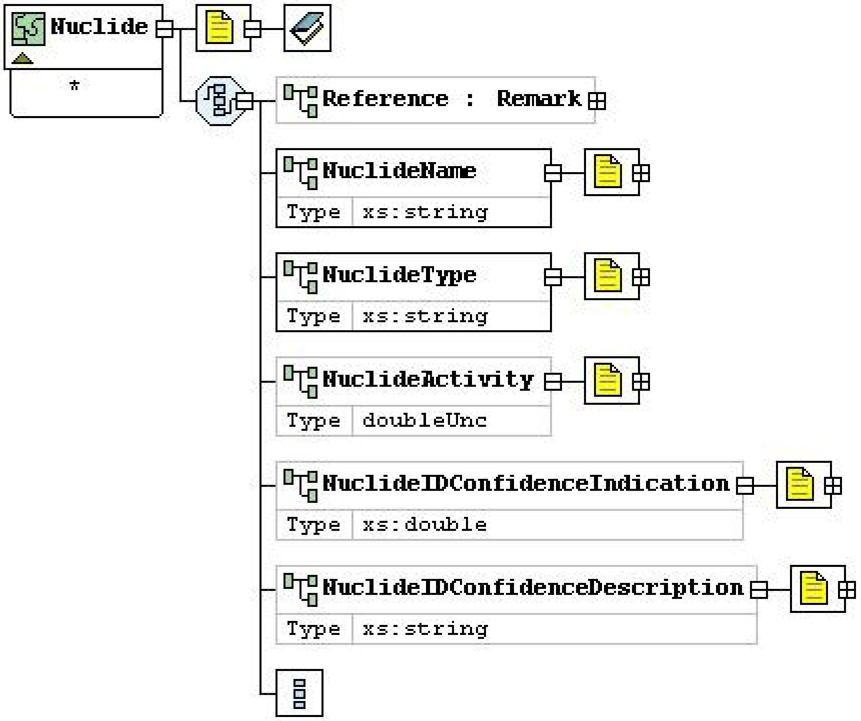
Schema Component Representation

```
<xs:complexType name="N42InstrumentData">
   <xs:sequence>
      <xs:element ref="Remark" minOccurs="0"/>
      <xs:element name="Measurement" type="Measurement"
      maxOccurs="unbounded"/>
      <xs:element name="Calibration" type="Calibration" minOccurs="0"
      maxOccurs="unbounded"/>
      <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
      processContents="lax"/>
   </xs:sequence>
</xs:complexType>
```

top

Complex Type: Nuclide

Super-types:	None
Sub-types:	None

Name	Nuclide
Abstract	no
Documentation	individual nuclide
Diagram	 Nuclide * Reference : Remark NuclideName Type xs:string NuclideType Type xs:string NuclideActivity Type doubleUnc NuclideIDConfidenceIndication Type xs:double NuclideIDConfidenceDescription Type xs:string

XML Instance Representation

```
<...
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
   <Remark> ... </Remark> [0..1]
   <NuclideName> xs:string </NuclideName> [1] ?
   <NuclideType> xs:string </NuclideType> [1] ?
   <NuclideActivity> doubleUnc </NuclideActivity> [0..1] ?
   <NuclideIDConfidenceIndication> xs:double
   </NuclideIDConfidenceIndication> [0..1] ?
   <NuclideIDConfidenceDescription> xs:string
   </NuclideIDConfidenceDescription> [0..1] ?
   Allow any elements from a namespace other than this schema's
   namespace (lax validation). [0..*]
</...>
```

Schema Component Representation

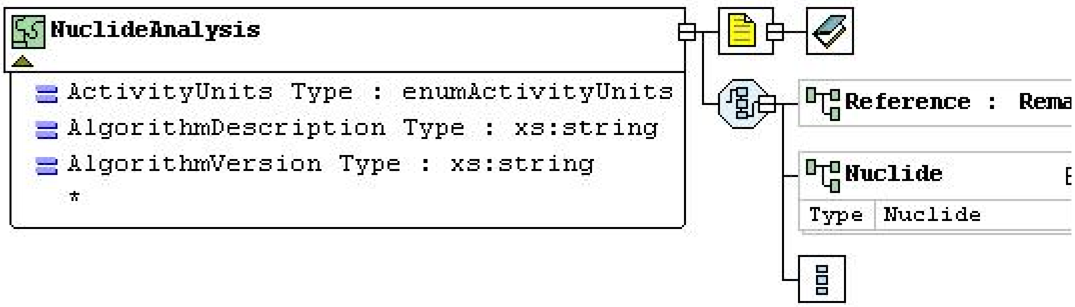
```
<xs:complexType name="Nuclide">
  <xs:sequence>
    <xs:element ref="Remark" minOccurs="0"/>
    <xs:element name="NuclideName" type="xs:string"/>
    <xs:element name="NuclideType" type="xs:string"/>
    <xs:element name="NuclideActivity" type="doubleUnc"
    minOccurs="0"/>
    <xs:element name="NuclideIDConfidenceIndication" type="xs:double"
    minOccurs="0"/>
    <xs:element name="NuclideIDConfidenceDescription"
    type="xs:string" minOccurs="0"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
    processContents="lax"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

top

Complex Type: NuclideAnalysis

Super-types:	None
Sub-types:	None

Name	NuclideAnalysis
Abstract	no
Documentation	Results of nuclide analysis
Diagram	


NuclideAnalysis
ActivityUnits Type : enumActivityUnits
AlgorithmDescription Type : xs:string
AlgorithmVersion Type : xs:string
*
Reference : Rema
Nuclide
Type Nuclide

XML Instance Representation

```
<...
 ActivityUnits="enumActivityUnits [1] ?"
 AlgorithmDescription="xs:string [0..1] ?"
 AlgorithmVersion="xs:string [0..1] ?"
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
   <Remark> ... </Remark> [0..1]
   <Nuclide> Nuclide </Nuclide> [0..*] ?
   Allow any elements from a namespace other than this schema's
   namespace (lax validation). [0..*]
</...>
```

Schema Component Representation

```
<xs:complexType name="NuclideAnalysis">
   <xs:sequence>
      <xs:element ref="Remark" minOccurs="0"/>
      <xs:element name="Nuclide" type="Nuclide" minOccurs="0"
      maxOccurs="unbounded"/>
      <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
      processContents="lax"/>
   </xs:sequence>
   <xs:attribute name="ActivityUnits" type="enumActivityUnits"
   use="required"/>
   <xs:attribute name="AlgorithmDescription" type="xs:string"/>
   <xs:attribute name="AlgorithmVersion" type="xs:string"/>
   <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

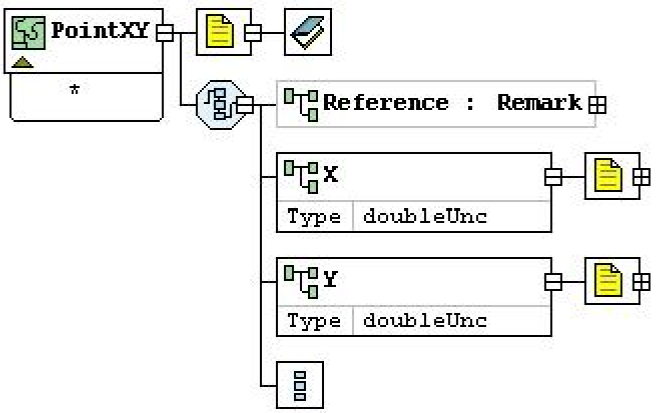
top

Complex Type: PointXY

Super-types:	None
Sub-types:	None

Name	PointXY
Abstract	no
Documentation	An (abscissa, ordinate) pair

Diagram



XML Instance Representation

```
<...
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
  <Remark> ... </Remark> [0..1]
  <X> doubleUnc </X> [1] ?
  <Y> doubleUnc </Y> [1] ?
  Allow any elements from a namespace other than this schema's
  namespace (lax validation). [0..*]
</...>
```

Schema Component Representation

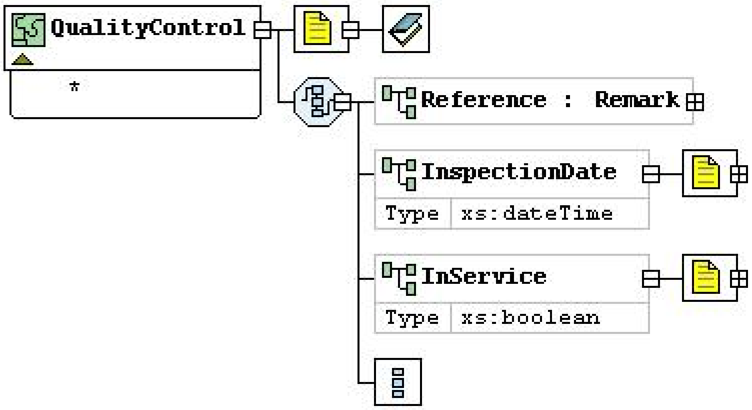
```
<xs:complexType name="PointXY">
  <xs:sequence>
    <xs:element ref="Remark" minOccurs="0"/>
    <xs:element name="X" type="doubleUnc"/>
    <xs:element name="Y" type="doubleUnc"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
    processContents="lax"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

top

Complex Type: QualityControl

Super-types:	None
Sub-types:	None

Name	QualityControl
Abstract	no
Documentation	Quality control information
Diagram	



XML Instance Representation

```
<...
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
   <Remark> ... </Remark> [0..1]
   <InspectionDate> xs:dateTime </InspectionDate> [0..1] ?
   <InService> xs:boolean </InService> [0..1] ?
   Allow any elements from a namespace other than this schema's
   namespace (lax validation). [0..*]
</...>
```

Schema Component Representation

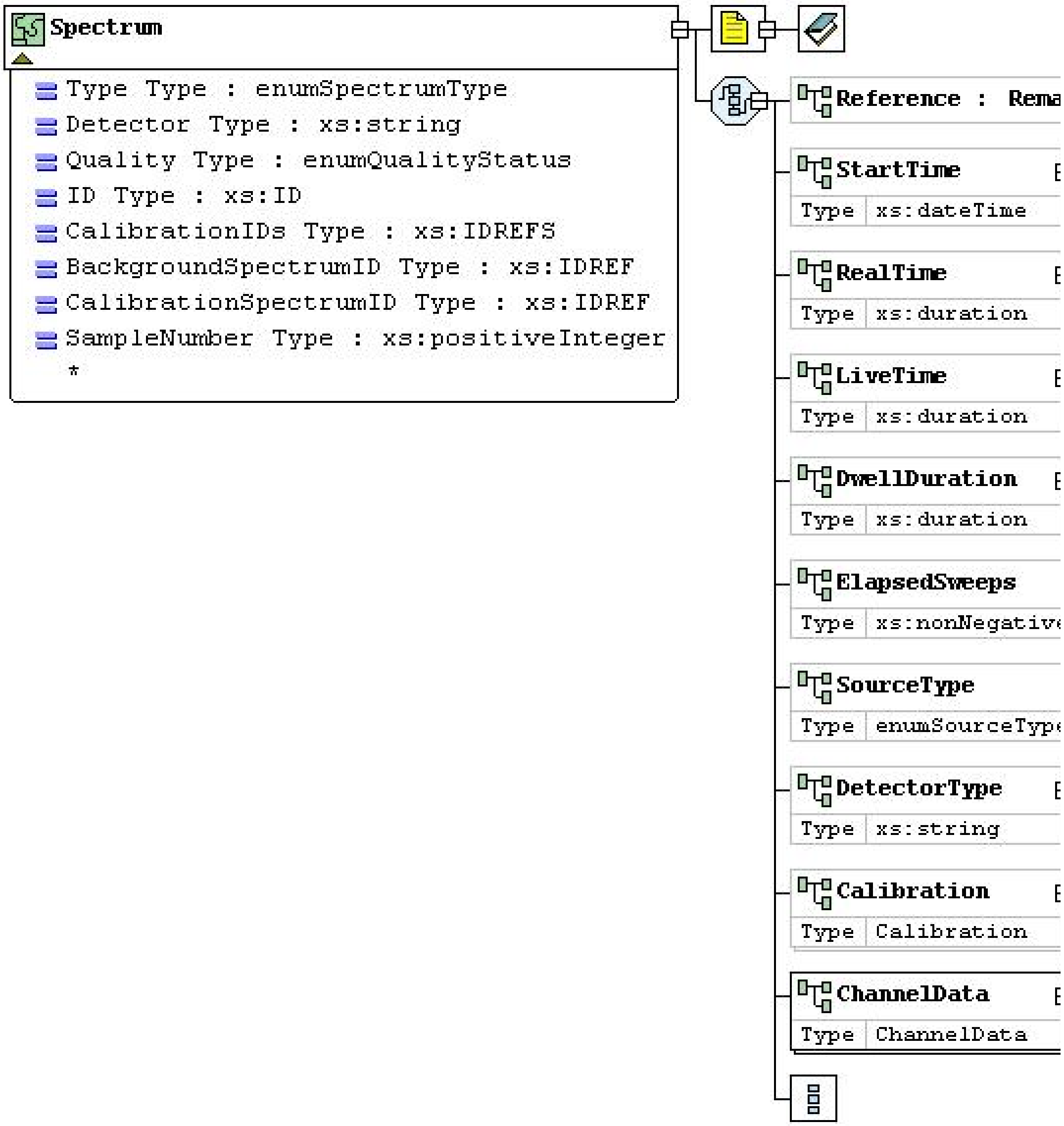
```
<xs:complexType name="QualityControl">
   <xs:sequence>
      <xs:element ref="Remark" minOccurs="0"/>
      <xs:element name="InspectionDate" type="xs:dateTime"
      minOccurs="0"/>
      <xs:element name="InService" type="xs:boolean" minOccurs="0"/>
      <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
      processContents="lax"/>
   </xs:sequence>
   <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

top

Complex Type: Spectrum

Super-types:	None
Sub-types:	None

Name	Spectrum
Abstract	no
Documentation	Spectral data
Diagram	



XML Instance Representation

```
<...
 Type="enumSpectrumType [0..1]  ? "
 Detector="xs:string [0..1]  ? "
 Quality="enumQualityStatus [0..1]  ? "
 ID="xs:ID [0..1]  ? "
 CalibrationIDs="xs:IDREFS [0..1]  ? "
 BackgroundSpectrumID="xs:IDREF [0..1]  ? "
 CalibrationSpectrumID="xs:IDREF [0..1]  ? "
 SampleNumber="xs:positiveInteger [0..1]  ? "
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
   <Remark> ... </Remark> [0..1]
   <StartTime> xs:dateTime </StartTime> [0..1]  ?
   <RealTime> xs:duration </RealTime> [0..1]  ?
   <LiveTime> xs:duration </LiveTime> [0..1]  ?
   <DwellDuration> xs:duration </DwellDuration> [0..1]  ?
```

```
  <ElapsedSweeps> xs:nonNegativeInteger </ElapsedSweeps> [0..1] ?
  <SourceType> enumSourceType </SourceType> [0..1] ?
  <DetectorType> xs:string </DetectorType> [0..1] ?
  <Calibration> Calibration </Calibration> [0..*] ?
  <ChannelData> ChannelData </ChannelData> [1..*] ?
  Allow any elements from a namespace other than this schema's
  namespace (lax validation). [0..*]
</...>
```

Schema Component Representation

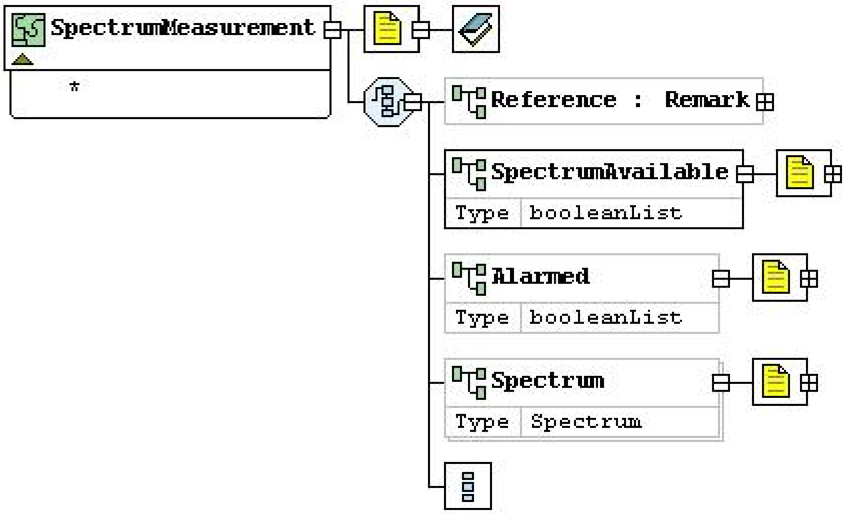
```
<xs:complexType name="Spectrum">
  <xs:sequence>
    <xs:element ref="Remark" minOccurs="0"/>
    <xs:element name="StartTime" type="xs:dateTime" minOccurs="0"/>
    <xs:element name="RealTime" type="xs:duration" minOccurs="0"/>
    <xs:element name="LiveTime" type="xs:duration" minOccurs="0"/>
    <xs:element name="DwellDuration" type="xs:duration"
    minOccurs="0"/>
    <xs:element name="ElapsedSweeps" type="xs:nonNegativeInteger"
    minOccurs="0"/>
    <xs:element name="SourceType" type="enumSourceType"
    minOccurs="0"/>
    <xs:element name="DetectorType" type="xs:string" minOccurs="0"/>
    <xs:element name="Calibration" type="Calibration" minOccurs="0"
    maxOccurs="unbounded"/>
    <xs:element name="ChannelData" type="ChannelData"
    maxOccurs="unbounded"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
    processContents="lax"/>
  </xs:sequence>
  <xs:attribute name="Type" type="enumSpectrumType"/>
  <xs:attribute name="Detector" type="xs:string"/>
  <xs:attribute name="Quality" type="enumQualityStatus"/>
  <xs:attribute name="ID" type="xs:ID"/>
  <xs:attribute name="CalibrationIDs" type="xs:IDREFS"/>
  <xs:attribute name="BackgroundSpectrumID" type="xs:IDREF"/>
  <xs:attribute name="CalibrationSpectrumID" type="xs:IDREF"/>
  <xs:attribute name="SampleNumber" type="xs:positiveInteger"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

top

Complex Type: SpectrumMeasurement

Super-types:	None
Sub-types:	None

Name	SpectrumMeasurement
Abstract	no
Documentation	Spectroscopic portal monitor measurements
Diagram	



XML Instance Representation

```
<...
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
   <Remark> ... </Remark> [0..1]
   <SpectrumAvailable> booleanList </SpectrumAvailable> [1] ?
   <Alarmed> booleanList </Alarmed> [0..1] ?
   <Spectrum> Spectrum </Spectrum> [0..*] ?
   Allow any elements from a namespace other than this schema's
   namespace (lax validation). [0..*]
</...>
```

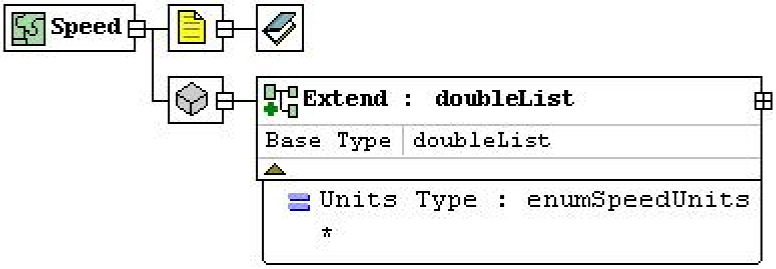
Schema Component Representation

```
<xs:complexType name="SpectrumMeasurement">
   <xs:sequence>
      <xs:element ref="Remark" minOccurs="0"/>
      <xs:element name="SpectrumAvailable" type="booleanList"/>
      <xs:element name="Alarmed" type="booleanList" minOccurs="0"/>
      <xs:element name="Spectrum" type="Spectrum" minOccurs="0"
      maxOccurs="unbounded"/>
      <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
      processContents="lax"/>
   </xs:sequence>
   <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

top

Complex Type: Speed

Super-types:	doubleList (by restriction) < **Speed** (by extension)
Sub-types:	None

Name	Speed
Abstract	no
Documentation	Speed of vehicle (can be the average, an entrance/exit speed pair, or per sample)
Diagram	

XML Instance Representation

```
<...
 Units="enumSpeedUnits [1] ?"
 Allow any attributes from a namespace other than this schema's
 namespace (lax validation).
>
   doubleList
</...>
```

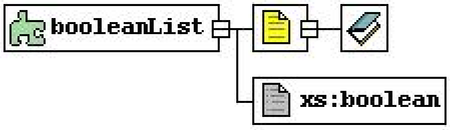
Schema Component Representation

```
<xs:complexType name="Speed">
   <xs:simpleContent>
      <xs:extension base="doubleList">
         <xs:attribute name="Units" type="enumSpeedUnits"
         use="required"/>
         <xs:anyAttribute namespace="##other" processContents="lax"/>
      </xs:extension>
   </xs:simpleContent>
</xs:complexType>
```

top

Simple Type: booleanList

Super-types:	None
Sub-types:	None

Name	booleanList
Content	• List of: xs:boolean
Documentation	List of booleans
Diagram	

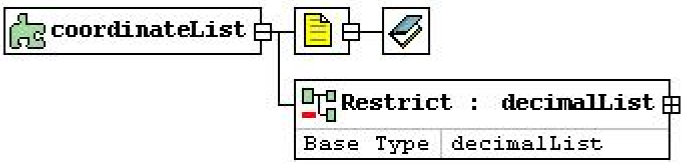
Schema Component Representation

```
<xs:simpleType name="booleanList">
  <xs:list itemType="xs:boolean"/>
</xs:simpleType>
```

top

Simple Type: coordinateList

Super-types:	decimalList (by restriction) < **coordinateList** (by restriction)
Sub-types:	- Coordinates (by extension)

Name	coordinateList
Content	- List of: xs:decimal - *length* >= 2
Documentation	Geographic coordinates: first two entries are latitude and longitude; third entry (optional) is elevation in meters relative to ellipsoid
Diagram	 coordinateList Restrict : decimalList Base Type decimalList

Schema Component Representation

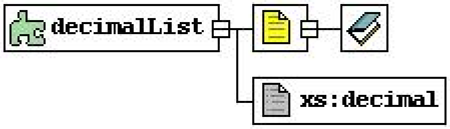
```
<xs:simpleType name="coordinateList">
  <xs:restriction base="decimalList">
    <xs:minLength value="2"/>
    <xs:maxLength value="3"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: decimalList

Super-types:	None
Sub-types:	- coordinateList (by restriction) ○ Coordinates (by extension)

Name	decimalList
Content	- List of: xs:decimal
Documentation	List of decimals
Diagram	



Schema Component Representation

```
<xs:simpleType name="decimalList">
  <xs:list itemType="xs:decimal"/>
</xs:simpleType>
```

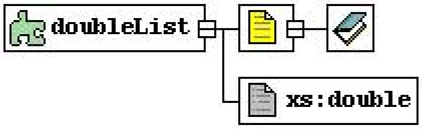
top

Simple Type: doubleList

Super-types: None

Sub-types:

- doubleUnc (by restriction)
  - ItemQuantity (by extension)
- Count (by extension)
- Dose (by extension)
- AbsorbedDose (by extension)
- Exposure (by extension)
- Speed (by extension)
- ChannelData (by extension)
- Coefficients (by extension)

Name	doubleList
Content	• List of: xs:double
Documentation	List of doubles
Diagram	

Schema Component Representation

```
<xs:simpleType name="doubleList">
  <xs:list itemType="xs:double"/>
</xs:simpleType>
```

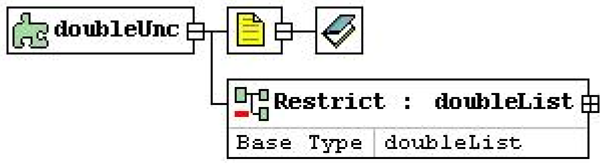
top

Simple Type: doubleUnc

Super-types: doubleList (by restriction) < **doubleUnc** (by restriction)

Sub-types:

- ItemQuantity (by extension)

Name	doubleUnc
Content	- List of: xs:double - *length* = 2
Documentation	Double value with uncertainty: first entry is value, second entry is uncertainty (absolute error, one standard deviation)
Diagram	 doubleUnc Restrict : doubleList Base Type doubleList

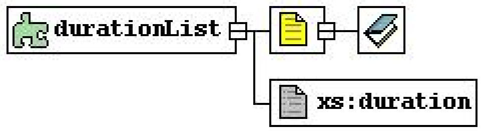
Schema Component Representation

```
<xs:simpleType name="doubleUnc">
   <xs:restriction base="doubleList">
      <xs:length value="2"/>
   </xs:restriction>
</xs:simpleType>
```

top

Simple Type: durationList

Super-types:	None
Sub-types:	- durationUnc (by restriction)

Name	durationList
Content	- List of: xs:duration
Documentation	List of durations
Diagram	

Schema Component Representation

```
<xs:simpleType name="durationList">
   <xs:list itemType="xs:duration"/>
</xs:simpleType>
```

top

Simple Type: durationUnc

Super-types:	durationList (by restriction) < **durationUnc** (by restriction)

Sub-types:	None

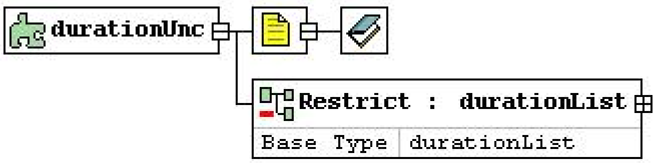
Name durationUnc

Content

- List of: xs:duration
- *length* = 2

Documentation Duration (time) value with uncertainty: first entry is value, second entry is uncertainty (absolute error, one standard deviation)

Diagram



Schema Component Representation

```
<xs:simpleType name="durationUnc">
  <xs:restriction base="durationList">
    <xs:length value="2"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: enumAbsorbedDoseUnits

Super-types:	xs:string < **enumAbsorbedDoseUnits** (by restriction)
Sub-types:	None

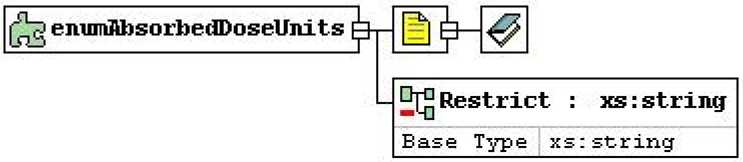
Name enumAbsorbedDoseUnits

Content

- Base XSD Type: string
- *value* comes from list: {'uGy'|'mGy'|'Gy'|'rad'}

Documentation Absorbed dose units

Diagram



Schema Component Representation

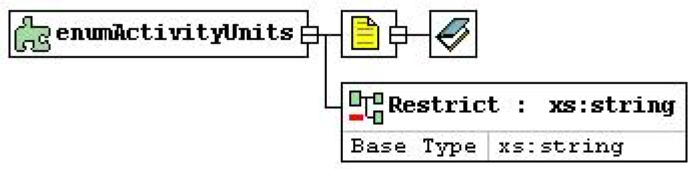
```
<xs:simpleType name="enumAbsorbedDoseUnits">
  <xs:restriction base="xs:string">
    <xs:enumeration value="uGy"/>
    <xs:enumeration value="mGy"/>
    <xs:enumeration value="Gy"/>
    <xs:enumeration value="rad"/>
  </xs:restriction>
```

```
</xs:simpleType>
```

top

Simple Type: enumActivityUnits

Super-types:	xs:string < **enumActivityUnits** (by restriction)
Sub-types:	None

Name	enumActivityUnits
Content	• Base XSD Type: string • *value* comes from list: {'mBq'|'Bq'|'kBq'|'MBq'|'nCi'|'uCi'|'mCi'|'Ci'|'kCi'|'MCi'}
Documentation	Activity units
Diagram	

Schema Component Representation

```
<xs:simpleType name="enumActivityUnits">
  <xs:restriction base="xs:string">
    <xs:enumeration value="mBq"/>
    <xs:enumeration value="Bq"/>
    <xs:enumeration value="kBq"/>
    <xs:enumeration value="MBq"/>
    <xs:enumeration value="nCi"/>
    <xs:enumeration value="uCi"/>
    <xs:enumeration value="mCi"/>
    <xs:enumeration value="Ci"/>
    <xs:enumeration value="kCi"/>
    <xs:enumeration value="MCi"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: enumCalibrationType

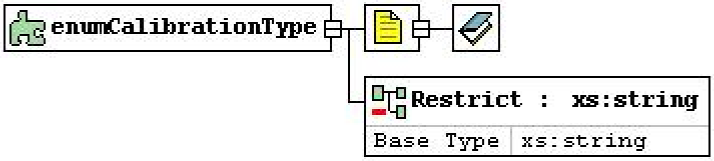
Super-types:	xs:string < **enumCalibrationType** (by restriction)
Sub-types:	None

Name	enumCalibrationType
Content	• Base XSD Type: string

- *value* comes from list: {'Energy'|'AbsoluteEfficiency'|'IntrinsicFullEnergyPeakEfficiency'|'IntrinsicTotalEfficiency'|

Documentation Calibration type

Diagram



Schema Component Representation

```
<xs:simpleType name="enumCalibrationType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Energy"/>
    <xs:enumeration value="AbsoluteEfficiency"/>
    <xs:enumeration value="IntrinsicFullEnergyPeakEfficiency"/>
    <xs:enumeration value="IntrinsicTotalEfficiency"/>
    <xs:enumeration value="FWHM"/>
    <xs:enumeration value="CountsToDose"/>
    <xs:enumeration value="CountsToAbsorbedDose"/>
    <xs:enumeration value="CountsToExposure"/>
    <xs:enumeration value="Other"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: enumCountRateUnits

Super-types:	xs:string < **enumCountRateUnits** (by restriction)
Sub-types:	None

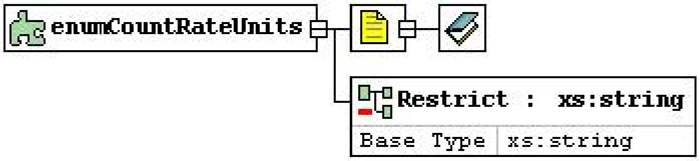
Name enumCountRateUnits

Content

- Base XSD Type: string
- *value* comes from list: {'CPS'|'CPM'}

Documentation Count rate units

Diagram



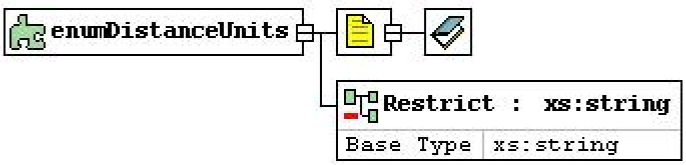
Schema Component Representation

```
<xs:simpleType name="enumCountRateUnits">
  <xs:restriction base="xs:string">
    <xs:enumeration value="CPS"/>
    <xs:enumeration value="CPM"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: enumDistanceUnits

Super-types:	xs:string < **enumDistanceUnits** (by restriction)
Sub-types:	None

Name	enumDistanceUnits
Content	• Base XSD Type: string • *value* comes from list: {'mm'|'cm'|'m'|'km'|'in'|'ft'|'mi'}
Documentation	Distance units
Diagram	 enumDistanceUnits Restrict : xs:string Base Type xs:string

Schema Component Representation

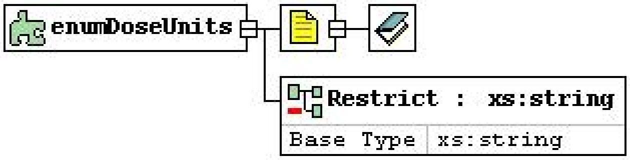
```
<xs:simpleType name="enumDistanceUnits">
  <xs:restriction base="xs:string">
    <xs:enumeration value="mm"/>
    <xs:enumeration value="cm"/>
    <xs:enumeration value="m"/>
    <xs:enumeration value="km"/>
    <xs:enumeration value="in"/>
    <xs:enumeration value="ft"/>
    <xs:enumeration value="mi"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: enumDoseUnits

Super-types:	xs:string < **enumDoseUnits** (by restriction)
Sub-types:	None

Name	enumDoseUnits
Content	• Base XSD Type: string • *value* comes from list: {'urem'|'mrem'|'rem'|'uSv'|'mSv'|'Sv'}
Documentation	Dose units
Diagram	



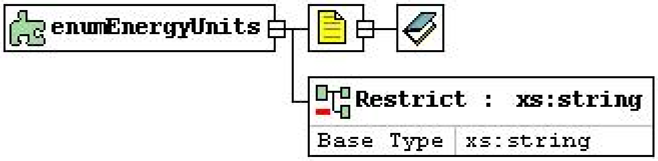
Schema Component Representation

```
<xs:simpleType name="enumDoseUnits">
  <xs:restriction base="xs:string">
    <xs:enumeration value="urem"/>
    <xs:enumeration value="mrem"/>
    <xs:enumeration value="rem"/>
    <xs:enumeration value="uSv"/>
    <xs:enumeration value="mSv"/>
    <xs:enumeration value="Sv"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: enumEnergyUnits

Super-types:	xs:string < **enumEnergyUnits** (by restriction)
Sub-types:	None

Name	enumEnergyUnits
Content	• Base XSD Type: string • *value* comes from list: {'eV'|'keV'|'MeV'}
Documentation	Energy units
Diagram	 enumEnergyUnits Restrict : xs:string Base Type xs:string

Schema Component Representation

```
<xs:simpleType name="enumEnergyUnits">
  <xs:restriction base="xs:string">
    <xs:enumeration value="eV"/>
    <xs:enumeration value="keV"/>
    <xs:enumeration value="MeV"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: enumEquationType

Super-types:	xs:string < **enumEquationType** (by restriction)
Sub-types:	None

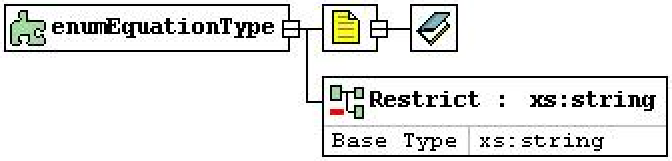
Name enumEquationType

Content

- Base XSD Type: string
- *value* comes from list: {'Polynomial'|'Pade'|'Exponential'|'PolyLogarithmic'|'FullRangeFraction'|'Other'}

Documentation Equation type

Diagram



Schema Component Representation

```
<xs:simpleType name="enumEquationType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Polynomial"/>
    <xs:enumeration value="Pade"/>
    <xs:enumeration value="Exponential"/>
    <xs:enumeration value="PolyLogarithmic"/>
    <xs:enumeration value="FullRangeFraction"/>
    <xs:enumeration value="Other"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: enumExposureUnits

Super-types:	xs:string < **enumExposureUnits** (by restriction)
Sub-types:	None

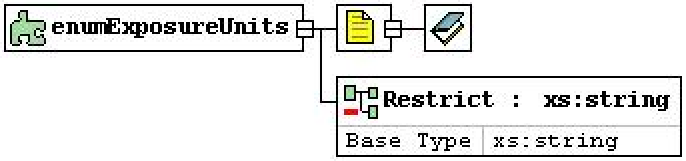
Name enumExposureUnits

Content

- Base XSD Type: string
- *value* comes from list: {'C/kg'|'uR'|'mR'|'R'}

Documentation Exposure units

Diagram



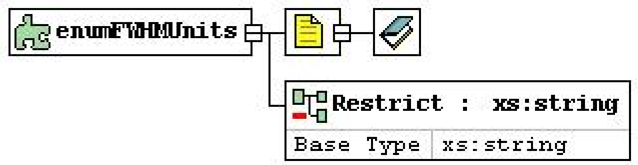
Schema Component Representation

```
<xs:simpleType name="enumExposureUnits">
   <xs:restriction base="xs:string">
      <xs:enumeration value="C/kg"/>
      <xs:enumeration value="uR"/>
      <xs:enumeration value="mR"/>
      <xs:enumeration value="R"/>
   </xs:restriction>
</xs:simpleType>
```

top

Simple Type: enumFWHMUnits

Super-types:	xs:string < **enumFWHMUnits** (by restriction)
Sub-types:	None

Name	enumFWHMUnits
Content	• Base XSD Type: string • *value* comes from list: {'Energy'|'Channels'}
Documentation	FWHM (Full Width at Half-Maximum) (i.e., resolution) units
Diagram	 enumFWHMUnits Restrict : xs:string Base Type xs:string

Schema Component Representation

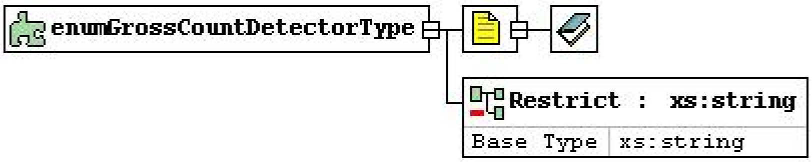
```
<xs:simpleType name="enumFWHMUnits">
   <xs:restriction base="xs:string">
      <xs:enumeration value="Energy"/>
      <xs:enumeration value="Channels"/>
   </xs:restriction>
</xs:simpleType>
```

top

Simple Type: enumGrossCountDetectorType

Super-types:	xs:string < **enumGrossCountDetectorType** (by restriction)
Sub-types:	None

Name	enumGrossCountDetectorType
Content	• Base XSD Type: string • *value* comes from list: {'Alpha'|'Beta'|'Gamma'|'Neutron'|'Other'}

Documentation	Gross counting detector type
Diagram	 enumGrossCountDetectorType Restrict : xs:string Base Type xs:string

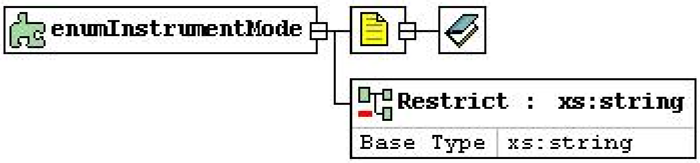
Schema Component Representation

```
<xs:simpleType name="enumGrossCountDetectorType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Alpha"/>
    <xs:enumeration value="Beta"/>
    <xs:enumeration value="Gamma"/>
    <xs:enumeration value="Neutron"/>
    <xs:enumeration value="Other"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: enumInstrumentMode

Super-types:	xs:string < **enumInstrumentMode** (by restriction)
Sub-types:	None

Name	enumInstrumentMode
Content	• Base XSD Type: string • *value* comes from list: {'Measure'|'Calibrate'|'Test'|'Other'}
Documentation	Instrument mode
Diagram	 enumInstrumentMode Restrict : xs:string Base Type xs:string

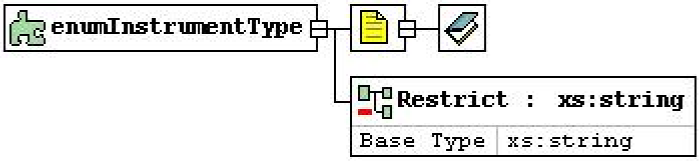
Schema Component Representation

```
<xs:simpleType name="enumInstrumentMode">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Measure"/>
    <xs:enumeration value="Calibrate"/>
    <xs:enumeration value="Test"/>
    <xs:enumeration value="Other"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: enumInstrumentType

Super-types:	xs:string < **enumInstrumentType** (by restriction)
Sub-types:	None

Name	enumInstrumentType
Content	● Base XSD Type: string ● *value* comes from list: {'PortalMonitor'|'SpecPortal'|'RadionuclideIdentifier'|'PersonalRadiationDetector'|'SurveyM
Documentation	Instrument type
Diagram	 enumInstrumentType Restrict : xs:string Base Type xs:string

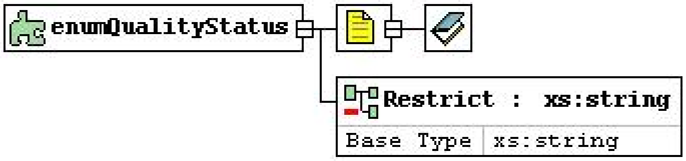
Schema Component Representation

```
<xs:simpleType name="enumInstrumentType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="PortalMonitor"/>
    <xs:enumeration value="SpecPortal"/>
    <xs:enumeration value="RadionuclideIdentifier"/>
    <xs:enumeration value="PersonalRadiationDetector"/>
    <xs:enumeration value="SurveyMeter"/>
    <xs:enumeration value="Spectrometer"/>
    <xs:enumeration value="Other"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: enumQualityStatus

Super-types:	xs:string < **enumQualityStatus** (by restriction)
Sub-types:	None

Name	enumQualityStatus
Content	● Base XSD Type: string ● *value* comes from list: {'Good'|'Bad'|'Suspect'|'Missing'}
Documentation	Data quality
Diagram	 enumQualityStatus Restrict : xs:string Base Type xs:string

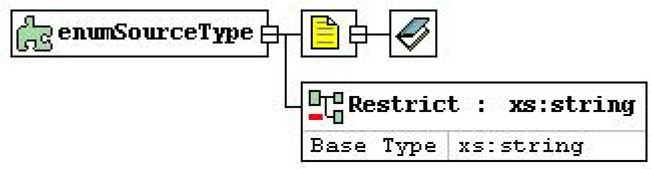
Schema Component Representation

```
<xs:simpleType name="enumQualityStatus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Good"/>
    <xs:enumeration value="Bad"/>
    <xs:enumeration value="Suspect"/>
    <xs:enumeration value="Missing"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: enumSourceType

Super-types:	xs:string < **enumSourceType** (by restriction)
Sub-types:	None

Name	enumSourceType
Content	• Base XSD Type: string • *value* comes from list: {'Item'|'Background'|'Calibration'|'Other'}
Documentation	Measurement type
Diagram	

Schema Component Representation

```
<xs:simpleType name="enumSourceType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Item"/>
    <xs:enumeration value="Background"/>
    <xs:enumeration value="Calibration"/>
    <xs:enumeration value="Other"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: enumSpectrumCompressionType

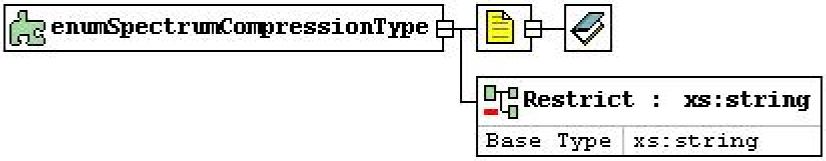
Super-types:	xs:string < **enumSpectrumCompressionType** (by restriction)
Sub-types:	None

Name	enumSpectrumCompressionType

Content
- Base XSD Type: string
- *value* comes from list: {'None'|'CountedZeroes'}

Documentation The algorithm (if any) by which the spectrum has been compressed

Diagram



Schema Component Representation

```
<xs:simpleType name="enumSpectrumCompressionType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="None"/>
    <xs:enumeration value="CountedZeroes"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: enumSpectrumType

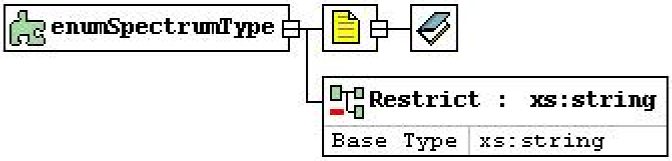
Super-types:	xs:string < **enumSpectrumType** (by restriction)
Sub-types:	None

Name enumSpectrumType

Content
- Base XSD Type: string
- *value* comes from list: {'PHA'|'MCS'|'Other'}

Documentation Spectrum type

Diagram



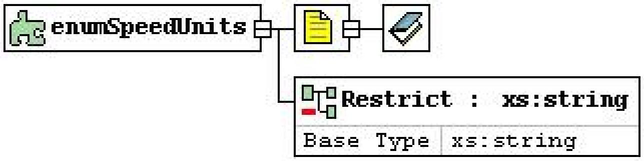
Schema Component Representation

```
<xs:simpleType name="enumSpectrumType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="PHA"/>
    <xs:enumeration value="MCS"/>
    <xs:enumeration value="Other"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: enumSpeedUnits

Super-types:	xs:string < **enumSpeedUnits** (by restriction)
Sub-types:	None

Name	enumSpeedUnits
Content	• Base XSD Type: string • *value* comes from list: {'kph'|'m/s'|'mph'|'fps'}
Documentation	Speed units
Diagram	 enumSpeedUnits Restrict : xs:string Base Type xs:string

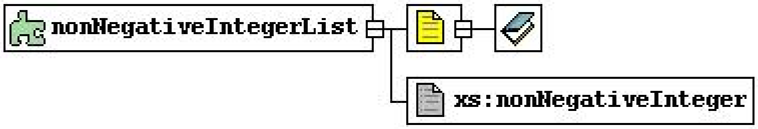
Schema Component Representation

```
<xs:simpleType name="enumSpeedUnits">
  <xs:restriction base="xs:string">
    <xs:enumeration value="kph"/>
    <xs:enumeration value="m/s"/>
    <xs:enumeration value="mph"/>
    <xs:enumeration value="fps"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: nonNegativeIntegerList

Super-types:	None
Sub-types:	None

Name	nonNegativeIntegerList
Content	• List of: xs:nonNegativeInteger
Documentation	List of nonNegativeIntegers
Diagram	

Schema Component Representation

```
<xs:simpleType name="nonNegativeIntegerList">
  <xs:list itemType="xs:nonNegativeInteger"/>
</xs:simpleType>
```

top

Generated by xs3p. Last modified: 08/18/2006 17:07:18