

XML Schema Design for the Message Text Format Military Standard

1. Background. The US Message Text Format (MTF) configuration Control Board (CCB) has adopted National Information Exchange Model (NIEM) naming and design rules for USMTF XML Schema. This document provides a complete overview of the resulting product, and how it serves configuration management and implementation purposes.

2. Overview. Reference (1) provides NIEM control measures for human understanding. Reference (2) provides the same information in a machine readable format for use in the testing and verification required for secure information exchange, as described in Reference (3). References 4-6 provide a sample XML Schema, annotations schema and local terminology definitions. These resources contain the basis for XML Schema design decisions to support secure information exchange using USMTF.

3. Scope. This document describes features which accurately define data models to support MTF in a manner which includes security tagging capabilities, and can be independently tested using NIEM rules. Efforts that are independent of NIEM will have to meet these testing requirements independently, or assume risk without them. All options create additional costs for the government.

4. MTF Data Design. The logical design of MTF data objects is retained in the XML Schema using the application information elements in annotations. Every data object in USMTF is defined using a globally scoped XML Schema SimpleType, for content, a ComplexType to add security tags, and an Element so that it can be referenced by other ComplexTypes. Substitution Groups are used for alternative content at the namespace level, but Choice is used for implementation. Structural Rules are provided in the annotations application information node, and implemented using Schematron XPath expressions.

a. Single Schema Concept. NIEM incorporates the W3C best practice in defining each namespace using a single namespace. This has many advantages not the least of which is the ability to use older XML parsers which are not namespace aware.

(1) Namespaces. When an XML Schema imports an XML Schema with a different namespace, then the items it uses from that namespace will be validated against it. When an XML Schema is extended, all of the content is included, with additional structures. When an XML Schema is Restricted, content is removed or adjusted, but it will remain valid against the source Schema so a new namespace is not required.

(2) Extension Schema. An extension schema will become the Reference Schema for a new namespace. This must then be restricted for implementation.

(3) Restriction Schema. Restriction Schema are expected to be dynamically generated based on operational requirements in order to meet network capacity and information security requirements. They will retain the namespace of the Reference or Extension Schema that they restrict.

b. Annotations. In accordance with NIEM, all annotations must be defined using a separate XML Schema. This is provided in Appendix (2).

c. Alternative Content. In accordance with NIEM design rules, USMTF XML uses Substitution Groups to model Choice in the Reference Schema, because this allows extension. For normative implementation, using Restriction Schema,

XML Schema Design for the Message Text Format Military Standard

extension of alternative content is not desired, so the non extensible XML Schema Choice model is preferred.

(1) SubstitutionGroup. This involves the creation of an "Abstract" Element, which is never reflected in an instance but can be used as the context for choice items by adding a substitutionGroup attribute to any global element with the name of the abstract element.

(2) Choice. Substitution Groups make it possible to add additional alternatives, whereas the W3C XML Schema Choice element (xs:choice) cannot be extended. For implementation, when extension of alternative content is prohibited, it is expected that Substitution Groups will be converted to Choice elements.

(3) MTF Considerations. When alternative content is represented within an XML Element definition, has a "Position Name" this structure is preserved with an Alternative Content type and element. Unnamed alternatives are represented using Abstract Substitution Groups.

(a) Named Alternatives. In the Reference Schema, the Alternative Content contains a reference to the Abstract Substitution Group. For implementation the Alternative Content type will contain the W3C Choice structure.

(b) Unnamed Alternatives. In the Reference Schema, the Alternative Content contains a reference to the Abstract Substitution Group. For implementation the the W3C Choice structure replaces this reference.

d. Fields. Fields are the data objects in which information is stored. All other data structures are collections of Fields.

(1) Enumerations. Data items with values than can be expressed as a selection of codes are modelled as enumerations. This model does not support selection of multiple values.

(2) Regular Expressions. A regular expression is a sequence of characters that defined a pattern that can be used to restrict a field entry to desired content.

(3) Value Limits. Numeric values are assigned minimum and maximum values.

(4) Length Limits. Text fields are defined with minimum and maximum lengths. If numeric values must have lengths that are not reflected in numeric formats, such as leading zeros, a combination of value limits and regular expression must be used.

e. Composites. Composites are collections of references to globally defined Fields in a specific sequence, with occurrence information provided.

(1) Occurrence. Used to specify optional components, with minimum occurrence zero, required components, with minimum occurrence greater than zero, and to specify limits for repeating items.

XML Schema Design for the Message Text Format Military Standard

(2) References. All Fields, Composites, Sets, Segments and Messages are defined using global Elements, and included in context using references to these global elements. This supports re-use of data objects by implementing software.

f. Sets. Sets are collections of references to globally defined Fields, Composites, and other Sets in a specific sequence, with occurrence information provided.

g. Segments. Segments are collections of references to globally defined Sets, Composites, Fields and other Segments in a specific sequence, with occurrence information provided.

h. Messages. Messages are collections of references to globally defined Sets, Composites, Fields and Segments in a specific sequence, with occurrence information provided.

(1) Structural Rules. These are used to define required values, and occurrence of specific objects based on the occurrence or values of other Fields, Composites, Sets, or Segments.

(2) Restriction. The normative definitions for MTF messages are necessarily broad to allow the further application of required mission specific restrictions for implementation. Subset and Restriction Schema are required for implementation of MTF Messages in a consistent, and testable manner.

5. XML Schema Validation

a. Reference Schema. All instances of a Restriction Schema must be valid against the restricted Reference Schema. Nodes of an Extension Schema that are imported from a Reference Schema must also be valid.

(1) Subset Schema. For validation purposes Subset Schema are equivalent to the parent Reference Schema, and must be Restricted for implementation. Individual messages are represented using Subset Schema.

(2) Extension Schema. Any addition or adjustment that will render an instance invalid against the Reference Schema requires the creation of an Extension Schema with a new namespace. This Extension Schema must then be Restricted for implementation.

b. Restriction Schema. Instances of a Restriction Schema must be valid against both the Restriction Schema and the Reference, Subset, or Extension Schema from which it is derived. Restriction Schema are not expected to be further restricted or extended.

c. Default Values. When required fields are not used in Restriction Schema, fixed values are specified in the Restriction Schema.

d. Required Nodes. When required fields cannot be used due to restrictive Distribution Statements, or other reasons, an Extension Schema will be created without the restricted node.

e. Structural Relationship Rules. Conditional rules are provided using XPath statements formatted using the Schematron namespace, and evaluated using

XML Schema Design for the Message Text Format Military Standard

XSLT. This method is used to verify rules applying to any XML document, to include XML Schema, XML instances, and other XSLT products.

6. XSLT Processing. The Extensible Stylesheet Language for Translation (XSLT) is an XML defined language that is used to process conditional rules associated with messages and security tags. XSLT processing is required for validation of MTF XML Schema, and is also used to create XML extension and restriction schema from the reference schema.

7. XML Schema Restriction. This is a derivation that is achieved by removing or altering XML Schema nodes in such a way as to create subsets of the original definition. This is considered a Normative Approach to implementing MTF using XML.

8. XML Schema Extension. This is a derivation that adds or alters information in a way that will support instances that will not be valid against the Extended Schema. This is appropriate for special use-cases. These include backward compatible XML Schema, and alterations required for security purposes.

9. Implementation

a. Code Generation. The machine readable functionality of XML Schema is applied primary to auto-generation of software artifacts for implementation. These include data structures, and tests to verify that valid messages can be generated and parsed.

b. Authority To Operate. All MTF implementations are expected to implement security tags. Proper evaluation and handling of these security tags is required for deployment. The potential for spillage is reduced by restricting messaged content in accordance to mission specifications. Implementations must support continuous, independent evaluation of functional parameters using external test data.

10. References

a. National Information Exchange Model (NIEM) Naming and Design Rules.
<https://reference.niem.gov/niem/specification/naming-and-design-rules/4.0/niem-ndr-4.0.html>

b. NIEM NDR Machine-Readable Rules. <https://github.com/NIEM/NIEM-NDR>

c. MIL STD Implementation Recommendations.
https://github.com/mil-oss/MTFXML/blob/master/USMTF_NIEM/Doc/Implementation/MILSTD-XML-Implementation_Recommendations.pdf

d. MTF Sample Reference Schema Structure.
https://github.com/mil-oss/MTFXML/blob/master/USMTF_NIEM/Doc/SchemaDesign/xsd/SampleReferenceSchema.xsd

e. MTF Sample Restriction Schema Structure.
https://github.com/mil-oss/MTFXML/blob/master/USMTF_NIEM/Doc/SchemaDesign/xsd/SampleRestrictionSchema.xsd

XML Schema Design for the Message Text Format Military Standard

f. MTF Annotations Schema.

https://github.com/mil-oss/MTFXML/blob/master/USMTF_NIEM/Doc/SchemaDesign/ext/niem/mtfappinfo.xsd

g. MTF Local Terminology.

https://github.com/mil-oss/MTFXML/blob/master/USMTF_NIEM/Doc/SchemaDesign/ext/niem/localTerminology.xsd

XML Schema Design for the Message Text Format Military Standard

XML Schema Design for the Message Text Format Military Standard

Appendix A: MTF XML Schema Design. A representative example schema is provided in Reference (4) to demonstrate the design of the MTF data model. Samples of the Restriction schema format are provided inline for comparison.

a. Annotations. The top level annotation demonstrates the inclusion of structural relationship rules. Annotations throughout the document reflect the items defined in Reference (5).

```
<xs:annotation>
  <xs:documentation ism:classification="U"
    ism:ownerProducer="USA"
    ism:noticeType="DoD-Dist-A">A data item for the TEST is used to report on the status of a TEST.</xs:documentation>
  <xs:appinfo>
    <inf:Msg name="TEST MESSAGE"
      mtfid="TEST"
      sponsor="USAF"
      purpose="The TEST is used to report on the status of a TEST."
      MtfRemark="B.1.01.13 was updated by ICP M2017-04, which updated the REF Set."
      version="C.0.01.00"
      date="October 2018"
      remark="Created by ICP M2018-02"/>
    <inf:StructuralRelationships>
      <sch:schema queryBinding="xslt">
        <sch:ns uri="urn:mtf:mil:6040b:niem:mtf" prefix="mtf"/>
        <sch:ns uri="urn:us:gov:ic:ism" prefix="ism"/>
        <sch:title>Structural Relationship Rules</sch:title>
        <sch:pattern id="TestMessage">
          <sch:title>Structural Relationship Rules</sch:title>
          <sch:rule context="mtf:TestMessage/mtf:MessageIdentifierSet">
            <sch:assert fpi="[3]F1 A /TEST/"
              test="mtf:MessageTextFormatIdentifierText = 'TEST'">Field 1 in MSGID (Set 3) is assigned the value "TEST".</sch:assert>
            <sch:assert fpi="[3]F2 A /MIL-STD-6040(SERIES)/"
              test="mtf:StandardOfMessageTextFormatCode = 'MIL-STD-6040(SERIES)'">Field 2 in MSGID (Set 3) is assigned the value "MIL-STD-6040(SERIES)".</sch:assert>
            <sch:assert fpi="[3]F3 A /B.1.01.14/"
              test="mtf:VersionOfMessageTextFormatText = 'B.1.01.14'">Field 3 in MSGID (Set 3) is assigned the value "B.1.01.14".</sch:assert>
          </sch:rule>
          <sch:rule context="mtf:TestMessage/mtf:ReferenceSet">
            <sch:assert fpi="[4],AF3 R ([4],AF2 EQ /CHT/ | /DOC/)/"
              test="mtf:InformationProduct and (mtf:TypeCommunicationCode='CHT' or mtf:TypeCommunicationCode='DOC')">Field 3 in REF (Set 4) is required if Field 2 in the same REF (Set 4) lexicographically equals "CHT" or "DOC".</sch:assert>
          </sch:rule>
          <sch:rule context="mtf:TestMessage/mtf:SetOne">
            <sch:assert fpi="[6],AF2 R ([6],AF3 !@)"
              test="mtf:CompositeOne and not(mtf:CompositeTwo)">Field 1 in SETONE (Set 6) is required if Field 2 in the same SETONE (Set 6) does not occur.</sch:assert>
          </sch:rule>
        </sch:pattern>
      </sch:schema>
    </inf:StructuralRelationships>
  </xs:appinfo>
</xs:annotation>
```

Figure A-1-Top Level Annotation

b. Alternative Content. In accordance with NIEM design rules, USMTF XML uses Substitution Groups to model Choice in the Reference Schema, because this

XML Schema Design for the Message Text Format Military Standard

allows extension. For normative implementation, using Restriction Schema, extension of alternative content is not desired, so the non extensible XML Schema Choice model is preferred.

(1) SubstitutionGroups. This involves the creation of an "Abstract" Element, which is never reflected in an instance but can be used as the context for choice items by adding a substitutionGroup attribute to any global element with the name of the abstract element.

```
<xs:complexType name="AlternativeContentType">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data type for alternative content.</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="structures:ObjectType">
      <xs:sequence>
        <xs:element ref="AlternativeContentAbstract" minOccurs="0">
          <xs:annotation>
            <xs:documentation ism:classification="U" ism:ownerProducer="USA"
              ism:noticeType="DoD-Dist-A">A data concept for a choice of a data item for
              the instructions for acknowledging or the force and/or unit required to
              acknowledge.</xs:documentation>
          </xs:annotation>
          <xs:appinfo>
            <inf:Field positionName="ALTERNATIVE CONTENT"
              type="AlternativeContentAbstractType" fieldseq="2"/>
            <inf:Choice substitutionGroup="AlternativeContentAbstract">
              <inf:Element name="EnumerationAlternative" type="EnumerationCodeType"/>
              <inf:Element name="IntegerAlternative" type="IntegerType"/>
              <inf:Element name="SetAlternative" type="SetType"/>
              <inf:Element name="SegmentAlternative" type="SegmentType"/>
            </inf:Choice>
          </xs:appinfo>
        </xs:element>
        <xs:element ref="AlternativeContentAugmentationPoint"
          minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:element name="AlternativeContentAugmentationPoint" abstract="true">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">An augmentation point for AlternativeContentType</xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="AlternativeContent" type="AlternativeContentType" abstract="true">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data item for Alternative Content</xs:documentation>
  </xs:annotation>
</xs:element>
```

Figure B-2-Substitution Group Element Definition for Named Alternatives

XML Schema Design for the Message Text Format Military Standard

(1) SubstitutionGroups. (cont.)

```
<xs:element name="AlternativeContentAbstract" abstract="true">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data concept for a choice of data items.</xs:documentation>
  </xs:annotation>
  <xs:appinfo>
    <inf:Field positionName="ALTERNATIVE CONTENT"
      type="AlternativeContentAbstractType"/>
    <inf:Choice substitutionGroup="AlternativeContentAbstract">
      <inf:Element name="EnumerationAlternative" type="EnumerationCodeType"/>
      <inf:Element name="IntegerAlternative" type="IntegerType"/>
      <inf:Element name="SetAlternative" type="SetType"/>
      <inf:Element name="SegmentAlternative" type="SegmentType"/>
    </inf:Choice>
  </xs:appinfo>
</xs:annotation>
</xs:element>

<xs:element name="EnumerationAlternative" type="EnumerationCodeType" substitutionGroup="AlternativeContentAbstract"
  nillable="true">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data item for Enumeration Code</xs:documentation>
  </xs:annotation>
  <xs:appinfo>
    <inf:Field positionName="ENUMERATION" version="00" ffirm="0000" fud="00"
      type="EnumerationCodeType"/>
  </xs:appinfo>
</xs:annotation>
</xs:element>
<xs:element name="CompositeAlternative" type="CompositeType" substitutionGroup="AlternativeContentAbstract" nillable="true">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data item for Composite One</xs:documentation>
  </xs:annotation>
  <xs:appinfo>
    <inf:Composite positionName="COMPOSITE ONE" version="00" ffirm="000" fud="00"
      type="CompositeComplexType" fieldseq="1"/>
  </xs:appinfo>
</xs:annotation>
</xs:element>
<xs:element name="SetAlternative" type="SetType" substitutionGroup="AlternativeContentAbstract" nillable="true">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data item for Composite Two</xs:documentation>
  </xs:annotation>
  <xs:appinfo>
    <inf:Set positionName="SET TWO" version="00" ffirm="000" fud="00" type="SetType"
      fieldseq="2"/>
  </xs:appinfo>
</xs:annotation>
</xs:element>
```

Figure B-1-Abstract Substitution Group

XML Schema Design for the Message Text Format Military Standard

(2) Choice. Substitution Groups make it possible to add additional alternatives, whereas the W3C XML Schema Choice element (xs:choice) cannot be extended. For implementation, when extension of alternative content is prohibited, it is expected that Substitution Groups will be converted to Choice elements.

```
<xs:complexType name="ChoiceAlternativeContentType">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data type for alternative content.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:choice minOccurs="0">
      <xs:annotation>
        <xs:documentation ism:classification="U" ism:ownerProducer="USA"
          ism:noticeType="DoD-Dist-A">A data concept for a choice of a data item for
          the instructions for acknowledging or the force and/or unit required to
          acknowledge.</xs:documentation>
        <xs:appinfo>
          <inf:Field positionName="ALTERNATIVE CONTENT" type="AlternativeContentAbstractType" fieldseq="2"/>
          <inf:Choice substitutionGroup="AlternativeContentAbstract">
            <inf:Element name="EnumerationAlternative" type="EnumerationCodeType"/>
            <inf:Element name="CompositeAlternative" type="CompositeType"/>
          </inf:Choice>
        </xs:appinfo>
      </xs:annotation>
      <xs:element ref="CompositeAlternative">
        <xs:annotation>
          <xs:documentation ism:classification="U" ism:ownerProducer="USA"
            ism:noticeType="DoD-Dist-A">A data item for Composite
            One</xs:documentation>
          <xs:appinfo>
            <inf:Composite positionName="COMPOSITE ONE" ffirm="000" fud="00" type="CompositeComplexType"
fieldseq="1"/>
          </xs:appinfo>
        </xs:annotation>
      </xs:element>
      <xs:element ref="SetAlternative">
        <xs:annotation>
          <xs:documentation ism:classification="U" ism:ownerProducer="USA"
            ism:noticeType="DoD-Dist-A">A data item for Composite
            Two</xs:documentation>
          <xs:appinfo>
            <inf:Set positionName="SET TWO" ffirm="000" fud="00" type="SetType" fieldseq="2"/>
          </xs:appinfo>
        </xs:annotation>
      </xs:element>
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

Figure B-3-Choice Type

c. SimpleTypes. As the base data definitions for the namespace, SimpleTypes remain the same in Reference and Restriction Schema. Further restrictions are accomplished in the ComplexTypes.

(1) FieldTextSimpleType. This example shows the use of a Regular Expression, as well as minimum and maximum lengths. Length restriction scan be included in the Regular expression, but to prevent the need to parse this field to

XML Schema Design for the Message Text Format Military Standard

get the values, MTF fields will always specify length limits explicitly. All text names will have a naming pattern that includes "Text."

```
<xs:simpleType name="FieldTextSimpleType">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data type for the field text.</xs:documentation>
    <xs:appinfo>
      <inf:Field name="FIELD" version="C.0.01.00" ffirm="000" fud="00"
        type="FieldTextType"/>
    </xs:appinfo>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="[\\-\\.\\(\\)\\?A-Z0-9]{1,38}" />
    <xs:minLength value="1" />
    <xs:maxLength value="38" />
  </xs:restriction>
</xs:simpleType>
```

Figure B-1-Text Simple Type

(2) EnumerationCodeSimpleType. This example shows the use of enumerations, which are termed "Code Lists" by NIEM. All enumeration names will have a naming pattern that includes "Code."

```
<xs:simpleType name="EnumerationCodeSimpleType">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data type for the enumeration code.</xs:documentation>
    <xs:appinfo>
      <inf:Field name="ENUMERATION" version="C.0.01.00" ffirm="35" fud="2"
        date="August 2018" remark="Created by ICP M2018-02" type="EnumerationCodeType"/>
    </xs:appinfo>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="ADD">
      <xs:annotation>
        <xs:documentation>Add.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="DEL">
      <xs:annotation>
        <xs:documentation>Delete.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>
```

Figure B-2-Enumeration Simple Type

XML Schema Design for the Message Text Format Military Standard

(3) IntegerSimpleType. This example shows the definition of integer data. A Regular expression can be used to enforce leading zeroes or other formatting requirements. All integer names will have a naming patten that includes "Numeric."

```
<xs:simpleType name="IntegerSimpleType">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data type for integer values.</xs:documentation>
    <xs:appinfo>
      <inf:Field name="Integer" version="C.0.01.00" versiondate="08/01/18"
        remark="Created by ICP M2018-01."/>
    </xs:appinfo>
  </xs:annotation>
  <xs:restriction base="xs:integer">
    <xs:pattern value="[0?]{1,3}"/>
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="100"/>
  </xs:restriction>
</xs:simpleType>
```

Figure B-3-Integer Simple Type

(4) DecimalSimpleType. This example shows the definition of decimal data. totalDigits and fractionDigits elements are used to determine the position of the decimal point. All decimal names will have a naming pattern that includes "Numeric."

```
<xs:simpleType name="DecimalSimpleType">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data type for decimal values.</xs:documentation>
    <xs:appinfo>
      <inf:Field name="Decimal" version="C.0.01.00" versiondate="October 2018"
        remark="Created by ICP M2018-02."/>
    </xs:appinfo>
  </xs:annotation>
  <xs:restriction base="xs:decimal">
    <xs:minInclusive value=".0000001"/>
    <xs:maxInclusive value="99999999"/>
    <xs:totalDigits value="8"/>
    <xs:fractionDigits value="7"/>
  </xs:restriction>
</xs:simpleType>
```

Figure B-4-Decimal Simple Type

XML Schema Design for the Message Text Format Military Standard

d. ComplexTypes. Each ComplexType extends a SimpleType by adding the "structures:SimpleObjectAttributeGroup." This NIEM resource contains a reference to ISM XML Schema for security tags. Restriction Schema used for IEPDs replace this broad option with a specific reference to ISM.

(1) FieldTextType

```
<xs:complexType name="FieldTextType">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data type for the field text.</xs:documentation>
    <xs:appinfo>
      <inf:Field name="FIELD TEXT" version="00" ffirm="00" fud="00" type="FieldTextType"/>
    </xs:appinfo>
  </xs:annotation>
  <xs:simpleContent>
    <xs:extension base="FieldTextSimpleType">
      <xs:attributeGroup ref="structures:SimpleObjectAttributeGroup"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

Figure C-1-Text Reference Type

```
<xs:complexType name="FieldTextType">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data type for the field text.</xs:documentation>
    <xs:appinfo>
      <inf:Field name="FIELD TEXT" version="00" ffirm="00" fud="00" type="FieldTextType"/>
    </xs:appinfo>
  </xs:annotation>
  <xs:simpleContent>
    <xs:extension base="FieldTextSimpleType">
      <xs:attributeGroup ref="ism:SecurityAttributesOptionGroup"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

Figure C-1-Text Restriction Type

(2) EnumerationCodeType

```
<xs:complexType name="EnumerationCodeType">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data type for the enumeration code.</xs:documentation>
    <xs:appinfo>
      <inf:Field name="ENUMERATION" version="C.0.00.00" ffirm="000" fud="00"
        type="EnumerationCodeSimpleType"/>
    </xs:appinfo>
  </xs:annotation>
  <xs:simpleContent>
    <xs:extension base="EnumerationCodeSimpleType">
      <xs:attributeGroup ref="structures:SimpleObjectAttributeGroup"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

Figure C-2-Enumeration Reference Type

(2) EnumerationCodeType (cont.)

```

<xs:complexType name="EnumerationCodeType">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data type for the enumeration code.</xs:documentation>
    <xs:appinfo>
      <inf:Field name="ENUMERATION" version="C.0.00.00" ffirm="000" fud="00"
        type="EnumerationCodeSimpleType"/>
    </xs:appinfo>
  </xs:annotation>
  <xs:simpleContent>
    <xs:extension base="EnumerationCodeSimpleType">
      <xs:attributeGroup ref="ism:SecurityAttributesOptionGroup"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

```

Figure C-2-Enumeration Restriction Type

(3) IntegerType

```

<xs:complexType name="IntegerType">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data type for integer values.</xs:documentation>
    <xs:appinfo>
      <inf:Field name="INTEGER" version="00" ffirm="00" fud="00" type="IntegerType"/>
    </xs:appinfo>
  </xs:annotation>
  <xs:simpleContent>
    <xs:extension base="IntegerSimpleType">
      <xs:attributeGroup ref="structures:SimpleObjectAttributeGroup"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

```

Figure C-3-Integer Reference Type

```

<xs:complexType name="IntegerType">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data type for integer values.</xs:documentation>
    <xs:appinfo>
      <inf:Field name="INTEGER" version="00" ffirm="00" fud="00" type="IntegerType"/>
    </xs:appinfo>
  </xs:annotation>
  <xs:simpleContent>
    <xs:extension base="IntegerSimpleType">
      <xs:attributeGroup ref="ism:SecurityAttributesOptionGroup"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

```

Figure C-3-Integer Restriction Type

(4) DecimalType

```

<xs:complexType name="DecimalType">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data type for decimal values.</xs:documentation>
    <xs:appinfo>
      <inf:Field name="DECIMAL" version="00" ffirm="000" fud="00" date="October 2018"
        type="DecimalType"/>
    </xs:appinfo>
  </xs:annotation>
  <xs:simpleContent>
    <xs:extension base="DecimalSimpleType">
      <xs:attributeGroup ref="structures:SimpleObjectAttributeGroup"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

```

Figure C-4-Decimal Reference Type

```

<xs:complexType name="DecimalType">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data type for decimal values.</xs:documentation>
    <xs:appinfo>
      <inf:Field name="DECIMAL" version="00" ffirm="000" fud="00" date="October 2018"
        type="DecimalType"/>
    </xs:appinfo>
  </xs:annotation>
  <xs:simpleContent>
    <xs:extension base="DecimalSimpleType">
      <xs:attributeGroup ref="ism:SecurityAttributesOptionGroup"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

```

Figure C-4-Decimal Restriction Type

(5) CompositeType

```

<xs:complexType name="CompositeType">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA" ism:noticeType="DoD-Dist-A">
A data type for composite complex type.</xs:documentation>
    <xs:appinfo>
      <inf:Composite name="COMPOSITE" version="00" type="CompositeType"/>
    </xs:appinfo>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="structures:ObjectType">
      <xs:sequence>
        <xs:element ref="FieldText">
          <xs:annotation>
            <xs:documentation>A data item for Field</xs:documentation>
            <xs:appinfo>
              <inf:Field name="FIELD" fieldseq="1"/>
            </xs:appinfo>
          </xs:annotation>
        </xs:element>
        <xs:element ref="EnumerationCode">
          <xs:annotation>
            <xs:documentation>A data item for Enumeration Code</xs:documentation>
            <xs:appinfo>
              <inf:Field name="ENUMERATION" type="EnumerationCodeType" fieldseq="2"/>
            </xs:appinfo>
          </xs:annotation>
        </xs:element>
        <xs:element ref="IntegerNumeric">
          <xs:annotation>
            <xs:documentation>A data item for Integer</xs:documentation>
            <xs:appinfo>
              <inf:Field name="INTEGER" type="IntegerType" fieldseq="3"/>
            </xs:appinfo>
          </xs:annotation>
        </xs:element>
        <xs:element ref="DecimalNumeric" minOccurs="0" maxOccurs="unbounded">
          <xs:annotation>
            <xs:documentation>A data item for Decimal</xs:documentation>
            <xs:appinfo>
              <inf:Field name="DECIMAL" type="DecimalType" fieldseq="4"/>
            </xs:appinfo>
          </xs:annotation>
        </xs:element>
        <xs:element ref="CompositeAugmentationPoint" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:element name="CompositeAugmentationPoint" abstract="true">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA" ism:noticeType="DoD-Dist-A">
An augmentation point for CompositeType</xs:documentation>
    </xs:annotation>
  </xs:element>

```

Figure C-5-Composite Reference Type

(6) CompositeType (cont.)

```

<xs:complexType name="CompositeType">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data type for composite complex
      type.</xs:documentation>
    <xs:appinfo>
      <inf:Composite name="COMPOSITE" version="00" type="CompositeType"/>
    </xs:appinfo>
  </xs:annotation>
  <xs:sequence>
    <xs:element ref="FieldText">
      <xs:annotation>
        <xs:documentation>A data item for Field</xs:documentation>
        <xs:appinfo>
          <inf:Field xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="FIELD"
            version="00" type="FieldTextType" fieldseq="1"/>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
    <xs:element ref="EnumerationCode">
      <xs:annotation>
        <xs:documentation>A data item for Enumeration Code</xs:documentation>
        <xs:appinfo>
          <inf:Field name="ENUMERATION" version="00"
            type="EnumerationCodeType" fieldseq="2"/>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
    <xs:element ref="IntegerNumeric">
      <xs:annotation>
        <xs:documentation>A data item for Integer</xs:documentation>
        <xs:appinfo>
          <inf:Field name="INTEGER" version="00" type="IntegerType"
            fieldseq="3"/>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
    <xs:element ref="DecimalNumeric" minOccurs="0" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>A data item for Decimal</xs:documentation>
        <xs:appinfo>
          <inf:Field name="DECIMAL" version="00" type="DecimalType"
            fieldseq="4"/>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

Figure C-5-Composite Restriction Type

```

<xs:complexType name="SetType">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data type for set complex type.</xs:documentation>
    <xs:appinfo>
      <inf:Composite name="SET" version="00" type="SetType"/>
    </xs:appinfo>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="structures:ObjectType">
      <xs:sequence>
        <xs:element ref="CompositeTwo">
          <xs:annotation>
            <xs:documentation>A data item for Composite Two</xs:documentation>
            <xs:appinfo>
              <inf:Composite name="COMPOSITE TWO" type="CompositeType" fieldseq="2"/>
            </xs:appinfo>
          </xs:annotation>
        </xs:element>
        <xs:element ref="FieldText">
          <xs:annotation>
            <xs:documentation>A data item for Field Text</xs:documentation>
            <xs:appinfo>
              <inf:Field name="FIELD TEXT" type="FieldTextType" fieldseq="3"/>
            </xs:appinfo>
          </xs:annotation>
        </xs:element>
        <xs:element ref="EnumerationCode">
          <xs:annotation>
            <xs:documentation>A data item for Enumeration Code</xs:documentation>
            <xs:appinfo>
              <inf:Field name="ENUMERATION" type="EnumerationCodeType" fieldseq="4"/>
            </xs:appinfo>
          </xs:annotation>
        </xs:element>
        <xs:element ref="IntegerNumeric">
          <xs:annotation>
            <xs:documentation>A data item for Integer</xs:documentation>
            <xs:appinfo>
              <inf:Field name="INTEGER" type="IntegerType" fieldseq="5"/>
            </xs:appinfo>
          </xs:annotation>
        </xs:element>
        <xs:element ref="SetAugmentationPoint" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:element name="SetAugmentationPoint" abstract="true">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">An augmentation point for SetType</xs:documentation>
  </xs:annotation>
</xs:element>

```

Figure C-6-Set Reference Type

(6) SetType (cont.)

```

<xs:complexType name="SetType">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data type for set complex type.</xs:documentation>
    <xs:appinfo>
      <inf:Composite name="SET" type="SetType"/>
    </xs:appinfo>
  </xs:annotation>
  <xs:sequence>
    <xs:element ref="CompositeOne">
      <xs:annotation>
        <xs:documentation>A data item for Composite One</xs:documentation>
        <xs:appinfo>
          <inf:Composite name="COMPOSITE ONE" type="CompositeType" fieldseq="1"/>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
    <xs:element ref="FieldText">
      <xs:annotation>
        <xs:documentation>A data item for Field Text</xs:documentation>
        <xs:appinfo>
          <inf:Field name="FIELD TEXT" type="FieldTextType" fieldseq="3"/>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
    <xs:element ref="EnumerationCode">
      <xs:annotation>
        <xs:documentation>A data item for Enumeration Code</xs:documentation>
        <xs:appinfo>
          <inf:Field name="ENUMERATION" type="EnumerationCodeType" fieldseq="4"/>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
    <xs:element ref="IntegerNumeric">
      <xs:annotation>
        <xs:documentation>A data item for Integer</xs:documentation>
        <xs:appinfo>
          <inf:Field name="INTEGER" type="IntegerType" fieldseq="5"/>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

Figure C-6-Set Restriction Type

(7) SegmentType

```

<xs:complexType name="SegmentType">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data type for set complex type.</xs:documentation>
    <xs:appinfo>
      <inf:Composite name="SET" version="00" type="SetType"/>
    </xs:appinfo>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="structures:ObjectType">
      <xs:sequence>
        <xs:element ref="SetOne">
          <xs:annotation>
            <xs:documentation>A data item for Set One</xs:documentation>
            <xs:appinfo>
              <inf:Set name="SET ONE" version="00" type="SetType" fieldseq="1"/>
            </xs:appinfo>
          </xs:annotation>
        </xs:element>
        <xs:element ref="SetTwo">
          <xs:annotation>
            <xs:documentation>A data item for Set Two</xs:documentation>
            <xs:appinfo>
              <inf:Set name="SET TWO" version="00" type="SetType" fieldseq="2"/>
            </xs:appinfo>
          </xs:annotation>
        </xs:element>
        <xs:element ref="SegmentAugmentationPoint" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

Figure C-7-Segment Reference Type

```

<xs:complexType name="SegmentType">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA" ism:noticeType="DoD-Dist-A">A data type for
set complex type.</xs:documentation>
    <xs:appinfo>
      <inf:Segment name="SEGMENT" type="SetType"/>
    </xs:appinfo>
  </xs:annotation>
  <xs:sequence>
    <xs:element ref="SetOne">
      <xs:annotation>
        <xs:documentation>A data item for Set One</xs:documentation>
        <xs:appinfo>
          <inf:Set name="SET ONE" version="00" type="SetType" fieldseq="1"/>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
    <xs:element ref="SetTwo">
      <xs:annotation>
        <xs:documentation>A data item for Set Two</xs:documentation>
        <xs:appinfo>
          <inf:Set name="SET TWO" type="SetType" fieldseq="2"/>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

Figure C-7-Segment Restriction Type

(8) MessageType

```

<xs:complexType name="MessageType">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data type for message complex type.</xs:documentation>
    <xs:appinfo>
      <inf:Msg mtfid="MSG1" name="MESSAGE ONE" version="00" type="MessageType"/>
    </xs:appinfo>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="structures:ObjectType">
      <xs:sequence>
        <xs:element ref="SetOne">
          <xs:annotation>
            <xs:documentation>A data item for Set</xs:documentation>
            <xs:appinfo>
              <inf:Set name="SET ONE" version="00" type="SetType" fieldseq="1"/>
            </xs:appinfo>
          </xs:annotation>
        </xs:element>
        <xs:element ref="SegmentOne">
          <xs:annotation>
            <xs:documentation>A data item for Segment One</xs:documentation>
            <xs:appinfo>
              <inf:Segment name="SEGMENT ONE" version="00" type="SegmentType"
                fieldseq="3"/>
            </xs:appinfo>
          </xs:annotation>
        </xs:element>
        <xs:element ref="CompositeOne">
          <xs:annotation>
            <xs:documentation>A data item for Field</xs:documentation>
            <xs:appinfo>
              <inf:Composite name="COMPOSITE ONE" version="00"
                type="CompositeType" fieldseq="5"/>
            </xs:appinfo>
          </xs:annotation>
        </xs:element>
        <xs:element ref="EnumerationCode">
          <xs:annotation>
            <xs:documentation>A data item for Enumeration Code</xs:documentation>
            <xs:appinfo>
              <inf:Field name="ENUMERATION" version="00"
                type="EnumerationCodeType" fieldseq="8"/>
            </xs:appinfo>
          </xs:annotation>
        </xs:element>
        <xs:element ref="MessageAugmentationPoint" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

Figure C-7-Message Reference Type

(8) MessageType (cont.)

```

<xs:complexType name="MessageType">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data type for message complex type.</xs:documentation>
    <xs:appinfo>
      <inf:Msg mtfid="MSG1" name="MESSAGE ONE" type="MessageType"/>
    </xs:appinfo>
  </xs:annotation>
  <xs:sequence>
    <xs:element ref="SetOne">
      <xs:annotation>
        <xs:documentation>A data item for Set</xs:documentation>
        <xs:appinfo>
          <inf:Set name="SET ONE" type="SetType" fieldseq="1"/>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
    <xs:element ref="SegmentOne">
      <xs:annotation>
        <xs:documentation>A data item for Segment One</xs:documentation>
        <xs:appinfo>
          <inf:Segment name="SEGMENT ONE" type="SegmentType" fieldseq="3"/>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
    <xs:element ref="CompositeOne">
      <xs:annotation>
        <xs:documentation>A data item for Field</xs:documentation>
        <xs:appinfo>
          <inf:Composite name="COMPOSITE ONE" type="CompositeType" fieldseq="5"/>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
    <xs:element ref="EnumerationCode">
      <xs:annotation>
        <xs:documentation>A data item for Enumeration Code</xs:documentation>
        <xs:appinfo>
          <inf:Field name="ENUMERATION" type="EnumerationCodeType" fieldseq="8"/>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
    <xs:element ref="MessageAugmentationPoint" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

```

Figure C-7-Message Restriction Type

XML Schema Design for the Message Text Format Military Standard

e. AttributeGroups. The only use of Attribute Groups in NIEM is from IC-ISM XML Schema.

```
<!-- ***** -->
<!-- SecurityAttributesGroup ..... -->
<!-- ***** -->
<xsd:attributeGroup name="SecurityAttributesGroup">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      The group of Information Security Marking attributes in which the use of attributes 'classification' and
      'ownerProducer' is required. This group is to be contrasted with group 'SecurityAttributesOptionGroup' in which
      use of those attributes is optional.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:attribute ref="classification" use="required"/>
  <xsd:attribute ref="ownerProducer" use="required"/>
  <xsd:attribute ref="SCIcontrols" use="optional"/>
  <xsd:attribute ref="SARIdentifier" use="optional"/>
  <xsd:attribute ref="disseminationControls" use="optional"/>
  <xsd:attribute ref="FGIsourceOpen" use="optional"/>
  <xsd:attribute ref="FGIsourceProtected" use="optional"/>
  <xsd:attribute ref="releasableTo" use="optional"/>
  <xsd:attribute ref="nonICmarkings" use="optional"/>
  <xsd:attribute ref="classifiedBy" use="optional"/>
  <xsd:attribute ref="derivativelyClassifiedBy" use="optional"/>
  <xsd:attribute ref="classificationReason" use="optional"/>
  <xsd:attribute ref="derivedFrom" use="optional"/>
  <xsd:attribute ref="declassDate" use="optional"/>
  <xsd:attribute ref="declassEvent" use="optional"/>
  <xsd:attribute ref="declassException" use="optional"/>
  <xsd:attribute ref="typeOfExemptedSource" use="optional"/>
  <xsd:attribute ref="dateOfExemptedSource" use="optional"/>
  <xsd:attribute ref="declassManualReview" use="optional"/>
</xsd:attributeGroup>
```

Figure D-1-Attribution Group

f. Elements. Each of the Complex Types above has references to Elements defined at the Global level. Element definitions are the same for Reference and Restriction Schema. Global elements do not have position names.

(1) FieldText

```
<xs:element name="FieldText" type="FieldTextType" nillable="true">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data item for Field Text</xs:documentation>
  <xs:appinfo>
    <inf:Field name="TEXT" ffirm="00" fud="00" type="FieldTextType"/>
  </xs:appinfo>
</xs:annotation>
</xs:element>
```

Figure E-1-Text Field Element

XML Schema Design for the Message Text Format Military Standard

(2) EnumerationCode

```
<xs:element name="EnumerationCode" type="EnumerationCodeType" nillable="true">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data item for Enumeration Code</xs:documentation>
    <xs:appinfo>
      <inf:Field name="ENUMERATION" version="00" ffirm="0000" fud="00" type="EnumerationCodeType"/>
    </xs:appinfo>
  </xs:annotation>
</xs:element>
```

Figure E-1-Enumeration Code Element

(3) IntegerNumeric

```
<xs:element name="IntegerNumeric" type="IntegerType" nillable="true">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data item for Integer</xs:documentation>
    <xs:appinfo>
      <inf:Field name="INTEGER" version="00" ffirm="00" fud="0" type="IntegerType"
        />
    </xs:appinfo>
  </xs:annotation>
</xs:element>
```

Figure E-1-Integer Element

(4) DecimalNumeric

```
<xs:element name="DecimalNumeric" type="DecimalType" nillable="true">
  <xs:annotation>
    <xs:documentation ism:classification="U" ism:ownerProducer="USA"
      ism:noticeType="DoD-Dist-A">A data item for Decimal</xs:documentation>
    <xs:appinfo>
      <inf:Field name="DECIMAL" version="00" ffirm="00" fud="0" type="DecimalType"/>
    </xs:appinfo>
  </xs:annotation>
</xs:element>
```

Figure E-1-Decimal Element

g. Attributes. Attribute nodes always have a Simple Type. This shows how they are used to define security tags for IC-ISM which uses a Simple Type with enumerations to specify attribute values.

```
<xsd:attribute name="classification" type="ClassificationType">
  <xsd:annotation>
    <xsd:documentation>IC-ISM Classification </xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
```

Figure F-1-IC-ISM Attribute

g. Attributes. (cont.)

```
<xsd:simpleType name="ClassificationType">
  <xsd:annotation>
    <xsd:documentation>IC-ISM Classification </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:NMTOKEN">
    <xsd:enumeration value="U"/>
    <xsd:enumeration value="C"/>
    <xsd:enumeration value="S"/>
    <xsd:enumeration value="TS"/>
    <xsd:enumeration value="R"/>
    <xsd:enumeration value="CTS"/>
    <xsd:enumeration value="CTS-B"/>
    <xsd:enumeration value="CTS-BALK"/>
    <xsd:enumeration value="NU"/>
    <xsd:enumeration value="NR"/>
    <xsd:enumeration value="NC"/>
    <xsd:enumeration value="NS"/>
    <xsd:enumeration value="NS-S"/>
    <xsd:enumeration value="NS-A"/>
    <xsd:enumeration value="CTSA"/>
    <xsd:enumeration value="NSAT"/>
    <xsd:enumeration value="NCA"/>
  </xsd:restriction>
</xsd:simpleType>
```

Figure F-2-IC-ISM Simple Type

h. Structural Relationship Rules. These are provided using the Schematron language, which is designed for processing by XSLT to run the XPath defined assertions.

```
<inf:StructuralRelationships>
  <sch:schema queryBinding="xslt">
    <sch:ns uri="urn:mtf:mil:6040b:niem:mtf" prefix="mtf"/>
    <sch:ns uri="urn:us:gov:ic:ism" prefix="ism"/>
    <sch:title>Structural Relationship Rules</sch:title>
    <sch:pattern id="TestMessage">
      <sch:title>Structural Relationship Rules</sch:title>
      <sch:rule context="mtf:TestMessage/mtf:MessageIdentifierSet">
        <sch:assert fpi="[3]F1 A /TEST/"
          test="mtf:MessageTextFormatIdentifierText = 'TEST'">Field 1 in MSGID (Set 3) is assigned the value
"TEST".</sch:assert>
        <sch:assert fpi="[3]F3 A /B.1.01.14/"
          test="mtf:VersionOfMessageTextFormatText = 'B.1.01.14'">Field 3 in MSGID (Set 3) is assigned the value
"B.1.01.14".</sch:assert>
      </sch:rule>
      <sch:rule context="mtf:TestMessage/mtf:ReferenceSet">
        <sch:assert fpi="[4],AF3 R ([4],AF2 EQ /CHT/ | /DOC/)"
          test="mtf:InformationProduct and (mtf:TypeCommunicationCode='CHT' or
mtf:TypeCommunicationCode='DOC')">Field 3 in REF (Set 4) is required if Field 2 in the same REF (Set 4) lexicographically
equals "CHT" or "DOC".</sch:assert>
      </sch:rule>
      <sch:rule context="mtf:TestMessage/mtf:SetOne">
        <sch:assert fpi="[6],AF2 R ([6],AF3 !@)"
          test="mtf:CompositeOne and not(mtf:CompositeTwo)">Field 1 in SETONE (Set 6) is required if Field 2 in the
same SETONE (Set 6) does not occur.</sch:assert>
      </sch:rule>
    </sch:pattern>
  </sch:schema>
</inf:StructuralRelationships>
```

Figure H-1-Structural Relationship Rules

j. Security Relationship Rules. Security markings are tested using Schematron provided by the IC-ISM XML Schema definitions.

```
<sch:pattern id="ISM-ID-00013" xmlns:sch="http://purl.oclc.org/dsdl/schematron">
  <sch:p class="ruleText">
    [ISM-ID-00013][Error] If ISM_NSI_EO_APPLIES then either attribute classifiedBy or
    derivedFrom must be specified on the ISM_RESOURCE_ELEMENT.

    Human Readable: Documents under E.O. 13526 must have classification authority block information.
  </sch:p>
  <sch:p class="codeDesc">
    If ISM_NSI_EO_APPLIES, we make sure that the ISM_RESOURCE_ELEMENT specifies
    attribute ism:classifiedBy or attribute ism:derivedFrom.
  </sch:p>
  <sch:rule context="*[$ISM_NSI_EO_APPLIES
    and generate-id(.) = generate-id($ISM_RESOURCE_ELEMENT)]">
    <sch:assert
      test="@ism:classifiedBy or @ism:derivedFrom"
      flag="error">
        [ISM-ID-00013][Error] Documents under E.O. 13526 must have classification authority block information.
      </sch:assert>
    </sch:rule>
  </sch:pattern>
```

Figure J-1-Security Relationship Rules