

HL7 Recommendation

CDA L1 R1 Schema

XML Schemas for CDA Level 1 Release 1

Structured Document Technical Committee
XML Special Interest Group
Informative Document
Revision Document as of 2005-09-21

Editor: Kai U. Heitmann, MD (hl7@kheitmann.nl)

Co-Chairs: Liora Alschuler (liora@the-word-electric.com)
 Calvin Beebe (cbeebe@mayo.edu)
 Paul V. Biron (paul.v.biron@kp.org)
 Sandra Boyer (slboyer@attglobal.net)
 Bob Dolin, MD (robert.h.dolin@kp.org)
 Kai U. Heitmann, MD (hl7@kheitmann.nl)

Contents

CDA L1 R1 Schema XML Schemas for CDA Level 1 Release 1	1
Contents	1
Preface	1
Acknowledgements.....	2
1 Introduction	2
1.1 Background	2
1.2 CDA Level One DTDs and XML Schemas	2
2 XML Schemas for CDA L1 R1	3
2.1 CDA Level One Body (levelone_1.0.xsd)	3
2.2 CDA Header (header_1.0.xsd)	13
2.3 HL7 Version 3 Data Types, Release 1 (v3dt_1.0.xsd)	26
2.4 Sample CDA Documents	51
3 References.....	54

Preface

This informative document presents a set of XML Schemas for the HL7's **Clinical Document Architecture (CDA) Level 1 Release 1** standard to be used as an alternative to the already existing XML 1.0 Document Type Definitions (DTDs). The class of documents validated by these schemas equals the class of documents validated by the DTDs distributed as part of the normative specification.

Acknowledgements

Special thanks to Ralf Schweiger who wrote the XML DTD to schema converting tool called *dtd2xsd* ([rfXSDtool]) and who initially converted and compiled the CDA L1 R1 DTDs to a single corresponding XML schema.

Contributions were made by Amnon Shabo (Israel), Joachim Dudeck (Germany), and Achim Marschall (Germany).

Thanks also to all members of the XML Special Interest Group and the Structured Documents Technical Committee for their input during the development and balloting process.

1 Introduction

1.1 Background

HL7's Clinical Document Architecture (CDA) is part of the HL7 version 3 family of standards for the representation of clinical documents (such as discharge summaries and progress notes). The HL7 CDA Framework, release 1.0 ([rfHL7CDA]), became an ANSI-approved HL7 standard in November 2000 (ANSI/HL7 CDA R1.0-2000). It covers the first ("CDA Level One") of three "levels" as part of an document architecture. For further information see also [rfBJ].

A "schema" is a specification or set of constraints for a class of documents. In other words: schemas are used to describe the structure (i.e. names and hierarchy of elements and attributes) of XML documents. The original CDA specification is expressed using one way of schema description, a collection of XML 1.0 *Document Type Definitions* (DTDs, see [rfXML]).

Since May 2001, the World Wide Web Consortium (W3C) recommends an alternative way, the so called *XML Schema language* ([rfXMLSchema]), which represents a more expressive way of specifying the constraints for a class of documents.

1.2 CDA Level One DTDs and XML Schemas

CDA Level One is specified by three components represented by three DTDs:

- The CDA Header is specified by the *CDA Header DTD*, which is derived by a method that closely parallels the V3 Message Development Framework.
- The CDA Level One Body is specified in the *CDA Level One DTD* and is derived from document analysis, building on the modeling employed by document markup standards.
- The HL7 Version 3 *Data Types, Release 1 DTD* is an XML implementation of the abstract data type specification (as available by that date) used by both the CDA and the HL7 Version 3 message specifications.

XML 1.0 DTD (part of the standard)	XML Schema (informative document)	Description
header_1.0.dtd	header_1.0.xsd	CDA Header
levelone_1.0.dtd	levelone_1.0.xsd	CDA Level One Body
v3DT_1.0.dtd	v3DT_1.0.xsd	HL7 Version 3 <i>Data Types, Release 1</i>

Table 1: Existing schemas for the Clinical Document Architecture

In order to utilize the advantages of the XML Schema language, and to meet the requirements coming from practical experiences with CDA in several projects, for example in Israel and Germany, the original CDA DTDs were converted into XML Schemas. The DTD fractions of the CDA specification were converted using a tool (see [rfXSDtool]). Then the resulting schemas were edited by hand and the comments drawn from the DTDs were added and adopted appropriately. A technical view into the conversion method is given in [rfDTDs].

The **levelone_1.0.xsd** includes the **header_1.0.xsd**. The **header_1.0.xsd** imports the **v3DT_1.0.xsd**. There is no namespace defined, instances do not show any namespaced elements. An CDA instance document have to define **levelone_1.0.xsd** as the schema for validation (see minimal cda document example in section 0).

The comments found in the DTDs were included in the XML Schemas as well, largely unchanged. A few changes have been made to several out-of-date-comments.

The comments are not provided as XML schema documentation-annotations. The HL7 processing rules are left as a comment and not provided as documentation-appinfo elements. This was both done to preserve the comment/definition construction used in the DTDs. Upcoming new XML Schemas for the CDA will make use of the usual documentation-annotations elements.

The xml:lang attributes found in the header and the data types schemas have been deprecated.

The version of the data types used here – the release 1 of the data types – was published together with the CDA DTDs in November 2000. At the time of this writing, the HL7 version 3 data types have had several re-definitions and ballots and are now different in some definitions. Release 1 was used to be consistent with the other CDA schemas. Future releases of the CDA specification and the corresponding schemas will use the most recent data type specifications.

2 XML Schemas for CDA L1 R1

Following the resulting XML Schemas for the Clinical Document Architecture Level 1 Release 1 to be used as an alternative to the already existing XML 1.0 Document Type Definitions (DTDs) for CDA L1R1.

2.1 CDA Level One Body (levelone_1.0.xsd)

This is the CDA Level One Body schema. It includes the CDA Header schema.

```
<?xml version="1.0"?>
<!--
=====
HL7 Clinical Document Architecture, Version 1.0

CDA Levelone XML Schema

Typical usage with an CDA instance document
<levelone
  xmlns="urn:hl7-org/cda"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:hl7-org/cda levelone_1.0.xsd">
  ...
</levelone>

corresponding DTD Last Revised: September 18, 2000
XML Schema last revised:
  2004-01-05 (Kai U. Heitmann):
    Namespace declarations have been eliminated

=====
-->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <!--
  =====
  The following system id must be changed to point to the location of the
  Header file on your system.
  =====
  -->
  <xsd:include schemaLocation="header_1.0.xsd"/>
  <!--
  =====
=====
```

Shared XML attributes

XML element identification

Every XML element within a CDA document has an optional identifier, which must be unique within the document.
(See 3.2.2.1.1 XML element identification). (This attribute is declared in the CDA Header DTD.)

Confidentiality

The confidentiality attribute can occur on any element within the CDA body. The CDA Header contains an optionally repeating element <confidentiality_cd> (see 3.2.2.2.3 Document confidentiality). The confidentiality attribute on CDA Body elements can reference one or more of the confidentiality values in the CDA Header using XML IDREFS. The value(s) referenced must be XML ID(s) in the <confidentiality_cd> element of the CDA Header. Confidentiality is inherited by nested content, unless overridden.

Originators

The originator attribute can occur on any element within the CDA body. The CDA Header contains optionally repeating elements <originator> (see 3.2.2.4.4.1 Originating person) and <originating_device> (see 3.2.2.5.2 Originating device). The originator attribute on an element within the CDA Body can reference one or more of these values using XML IDREFS. The value(s) referenced must be XML ID(s) in the <originator> or <originating_device> element of the CDA Header. Origination is inherited by nested content, unless overridden.

Use of xml:lang for the language property (as in the DTD) deprecated in XML Schema

```
=====
-->
<xsd:attributeGroup name="body_atts">
  <xsd:attributeGroup ref="common_atts"/>
  <xsd:attribute name="originator" type="xsd:IDREFS"/>
  <xsd:attribute name="confidentiality" type="xsd:IDREFS"/>
  <!-- deprecated
  <xsd:attribute ref="xml:lang" use="optional"/>
-->
</xsd:attributeGroup>
<xsd:group name="entries">
  <xsd:choice>
    <xsd:element ref="content"/>
    <xsd:element ref="link"/>
    <xsd:element ref="coded_entry"/>
    <xsd:element ref="observation_media"/>
    <xsd:element ref="local_markup"/>
  </xsd:choice>
</xsd:group>
<xsd:group name="structures">
  <xsd:choice>
    <xsd:element ref="paragraph"/>
    <xsd:element ref="list"/>
    <xsd:element ref="table"/>
  </xsd:choice>
</xsd:group>
<!--
=====
```

Level One Root

The CDA element <levelone> is the root element of a CDA Level One document. The <levelone> element contains a <clinical_document_header> and a <body>. The <clinical_document_header> is derived from the RIM (see 3.2 CDA Header). The <body> is comprised of either <section> elements, or a <non_xml> element, which is used when the document body is in some format other than XML. A CDA <section> can contain "structures", nested <section> elements, and <coded entry> elements. CDA structures include the <paragraph>, <list>, and <table> elements. These structures contain CDA "entries", which include the <content>, <link>, <coded_entry>, <observation_media>, and <local_markup> elements, in addition to plain character data.

```
=====
-->
<xsd:element name="levelone">
  <xsd:complexType>
```

```

        <xsd:sequence>
            <xsd:element ref="clinical_document_header"/>
            <xsd:element ref="body"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="body_atts"/>
    </xsd:complexType>
</xsd:element>
<!--
=====
=====
Document body and sections

The CDA <body> occurs in the <levelone> element. All CDA documents have
exactly one <body>. The <body> contains either one or more <section>
elements (see 3.3.2.2.2 Document sections) or a single non_xml data
segment (see 3.3.2.2.3 Non_xml body).

The CDA <section> is a container used to wrap other containers. A
<section> can occur in the <body>, or can be nested within another
<section>. A <section> has an optional <caption> (see 3.3.2.2.1
Captions), followed by nested <section> elements or structures (see
3.3.2.3 Document Structures), followed by optionally repeating <coded_entry>
elements (see 3.3.2.4.4 Coded entries).

The CDA <non_xml> container represents a document body that is in some
format other than XML. CDA's <non_xml> is an encoded data type (ED),
which is used only to reference data that is stored externally to the
CDA Level One document.
=====
=====
-->
    <xsd:element name="body">
        <xsd:complexType>
            <xsd:choice>
                <xsd:element ref="section" maxOccurs="unbounded"/>
                <xsd:element ref="non_xml"/>
            </xsd:choice>
            <xsd:attributeGroup ref="body_atts"/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="section">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="caption" minOccurs="0"/>
                <xsd:choice minOccurs="0" maxOccurs="unbounded">
                    <xsd:group ref="structures"/>
                    <xsd:element ref="section"/>
                </xsd:choice>
                <xsd:element ref="coded_entry" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:attributeGroup ref="body_atts"/>
        </xsd:complexType>
    </xsd:element>
    <xsd:complexType name="non_xml-cont.model" mixed="true">
        <xsd:complexContent mixed="true">
            <xsd:extension base="ED-cont.model">
                <xsd:attributeGroup ref="common_atts"/>
                <xsd:attribute name="originator" type="xsd:IDREFS"/>
                <xsd:attribute name="confidentiality" type="xsd:IDREFS"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name="non_xml" type="non_xml-cont.model"/>
<!--
=====
=====
Entries:
content, link, coded_entry, observation_media, local_markup

=====
=====
-->
<!--
=====
content

CDA <content> occurs in <local_markup>, table cells (<td>),
<paragraph>, <item>, and nested within <content>. The <content>
element contains zero or more entries (see 3.3.2.4 Document Entries).

The <content> element can nest recursively, which enables wrapping

```

a string of plain text down to as small a chunk as desired. These <content> elements can serve as anchors, and <coded_entry.value> elements can reference these anchors to indicate the original text that supports the use of a coded entry. (See 3.3.2.4.4 Coded entries for more detail.)

```
=====
-->
  <xsd:element name="content">
    <xsd:complexType mixed="true">
      <xsd:sequence minOccurs="0" maxOccurs="unbounded">
        <xsd:group ref="entries"/>
      </xsd:sequence>
      <xsd:attributeGroup ref="body_atts"/>
    </xsd:complexType>
  </xsd:element>
  <!--
```

=====

link

The CDA <link> is a generic referencing mechanism and occurs within <content>, <local_markup>, table cells (<td>), or <caption>. A <link> contains a single required <link_html> element.

The CDA <link_html> can only occur within a <link>. Each <link_html> has an optional local identifier (see 3.3.2.1.1 XML element identification), an optional set of confidentiality status flags (see 3.3.2.1.2 Confidentiality), and an optional set of originators (see 3.3.2.1.3 Originators). The human language of contained character data can be specified using the xml:lang attribute (see 3.3.2.1.4 Language).

The CDA link mechanism is based on the HTML anchor tag. Several groups (see 5.4 References) are actively developing formal link specifications. When a suitable open standard is available and implemented, it will be reviewed with the intent to incorporate it into the CDA Level One specification.

Multimedia that is integral to a document, and part of the attestable content of the document requires the use of <observation_media> (see 3.3.2.4.5 Observation media). Multimedia that is simply referenced by the document and not an integral part of the document should use <link>.

```
=====
-->
  <xsd:element name="link">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="link_html"/>
      </xsd:sequence>
      <xsd:attributeGroup ref="body_atts"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="link_html">
    <xsd:complexType>
      <xsd:simpleContent>
        <xsd:extension base="xsd:string">
          <xsd:attribute name="name" type="xsd:string"/>
          <xsd:attribute name="href" type="xsd:string"/>
          <xsd:attribute name="rel" type="xsd:string"/>
          <xsd:attribute name="rev" type="xsd:string"/>
          <xsd:attribute name="title" type="xsd:string"/>
        </xsd:extension>
      </xsd:simpleContent>
    </xsd:complexType>
  </xsd:element>
  <!--
```

=====

coded_entry

The CDA element <coded_entry> inserts codes from HL7-recognized coding schemes into CDA documents. Where there are no suitable HL7-recognized codes available, locally-defined codes can be used. The use of <coded_entry> in CDA Level One is unrestricted, and the primary intent of <coded_entry> is to facilitate document indexing, search and retrieval, and to provide a standard convention for insertion of locally-meaningful codes.

The <coded_entry.value> element can explicitly reference the original text within the document that supports the use of the code.

```
=====
-->
  <xsd:element name="coded_entry">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="coded_entry.id" minOccurs="0"/>
```

```

        <xsd:element ref="coded_entry.value"/>
        <xsd:element ref="local_markup" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attributeGroup ref="body_atts"/>
</xsd:complexType>
</xsd:element>
<xsd:element name="coded_entry.id">
    <xsd:complexType>
        <xsd:group ref="II-cont.model"/>
        <xsd:attributeGroup ref="common_atts"/>
        <xsd:attributeGroup ref="II-attrib.list"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="coded_entry.value">
    <xsd:complexType>
        <xsd:group ref="CD-cont.model"/>
        <xsd:attributeGroup ref="CD-attrib.list"/>
        <xsd:attributeGroup ref="common_atts"/>
    </xsd:complexType>
</xsd:element>
<!--
=====
observation_media

The <observation_media> element represents media that is logically a
part of a CDA document, but is stored outside the document and incorporated
by reference. Multimedia that is integral to a document, and part of the
attestable content of the document, requires the use of <observation_media>.
Multimedia that is simply referenced by the document and not an integral
part of the document should use <link> (see 3.3.2.4.3 Links). Note that
CDA's <observation_media> is used only to reference data that is stored
externally.

The CDA does not take advantage of ED's ability to Base64 encode images
and other observation media and include them directly in a document
instance file. Several groups (see 5.4 References) are actively
developing formal specifications for packaging binary data within XML
documents. When a suitable open standard for direct incorporation of
binary data is available and implemented, it will be incorporated into
the CDA Level One specification.
=====
-->
<xsd:element name="observation_media">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="observation_media.id" minOccurs="0"/>
            <xsd:element ref="observation_media.value"/>
            <xsd:element ref="local_markup" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="body_atts"/>
        <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="observation"/>
        <xsd:attribute name="T" type="xsd:string" fixed="observation"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="observation_media.id">
    <xsd:complexType>
        <xsd:group ref="II-cont.model"/>
        <xsd:attributeGroup ref="common_atts"/>
        <xsd:attributeGroup ref="II-attrib.list"/>
        <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="id"/>
    </xsd:complexType>
</xsd:element>
<xsd:complexType name="observation_value-cont.model" mixed="true">
    <xsd:complexContent mixed="true">
        <xsd:extension base="ED-cont.model">
            <xsd:attributeGroup ref="common_atts"/>
            <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="value"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:element name="observation_media.value" type="observation_value-cont.model"/>
<!--
=====
local_markup

The implementation of localization in the CDA Level One Body using the
<local_markup> element parallels the implementation described for the
CDA Header (see 3.2.2.6 Localization).

The descriptor attribute describes the element, and the value can be
drawn from a local vocabulary domain. The ignore attribute tells the

```

receiver to ignore just the <local_markup> tag (ignore="markup"), or to ignore the <local_markup> tag and all contained content (ignore="all"). The render attribute indicates how the sender would render the contents. The value can be drawn from a local vocabulary domain. The nested <local_attr> element makes it easier to map local XML attribute values into the CDA.

```
=====
-->
<xsd:element name="local_markup" type="local_markup-cont.model"/>
<xsd:complexType name="local_markup-cont.model" mixed="true">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:group ref="entries"/>
    <xsd:element ref="local_attr"/>
  </xsd:choice>
  <xsd:attribute name="ignore" default="markup">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="all"/>
        <xsd:enumeration value="markup"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
  <xsd:attribute name="descriptor" type="xsd:string"/>
  <xsd:attribute name="render" type="xsd:string"/>
</xsd:complexType>
<!--
```

Structures:
paragraph, list, table

```
=====
-->
<!--
```

paragraph

The CDA <paragraph> can occur in a <section>, <item>, or table cell (<td>). A <paragraph> has an optional <caption> (see 3.3.2.2.1 Captions), followed by zero or more <content> elements (see 3.3.2.4.2 Content).

```
=====
-->
<xsd:element name="paragraph">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="caption" minOccurs="0"/>
      <xsd:element ref="content" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attributeGroup ref="body_atts"/>
  </xsd:complexType>
</xsd:element>
<!--
```

list and item

The CDA <list> can occur in a <section>, <item>, or table cell (<td>). A <list> has an optional <caption> (see 3.3.2.2.1 Captions), and contains one or more <item> elements. The list_type attribute specifies whether the <list> is ordered or unordered (with unordered being the default). Use an ordered list when the ordering of list items is meaningful.

The CDA <item> only occurs within a <list>. An <item> has an optional <caption> (see 3.3.2.2.1 Captions), and may contain <content> (see 3.3.2.4.2 Content) and nested structures (see 3.3.2.3 Document Structures).

```
=====
-->
<xsd:element name="list">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="caption" minOccurs="0"/>
      <xsd:element ref="item" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attributeGroup ref="body_atts"/>
    <xsd:attribute name="list_type" default="unordered">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="ordered"/>
          <xsd:enumeration value="unordered"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>
<!--
```



```

        </xsd:simpleType>
      </xsd:attribute>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="item">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="caption" minOccurs="0"/>
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
          <xsd:element ref="content"/>
          <xsd:group ref="structures"/>
        </xsd:choice>
      </xsd:sequence>
      <xsd:attributeGroup ref="body_atts"/>
    </xsd:complexType>
  </xsd:element>
  <!--
=====
table

In CDA Level One, any information can be presented as a table.
The table markup is for presentation purposes only and, unlike a
database table, does not possess meaningful field names. The CDA
<table> can occur in a <section> or <item>. A <table> has an optional
<caption> (see 3.3.2.2.2.1 Captions).

CDA modifies the strict XHTML table model (see 5.4 References and
Appendix 5.3.1 Tables) by removing formatting tags and by setting
the content model of cells to be similar to the contents of other
CDA containers. The <th> element is modeled analogously to the
<caption> element (see 3.3.2.2.2.1 Captions), and like the <caption>
element, the <caption_cd> is optional and non-repeatable, and must
occur first.

Changes to the strict XHTML table model in CDA include
(expressed as DTD):

Change this:
  <!ELEMENT caption %Inline;>
To this:
  <!ELEMENT caption (#PCDATA | link | caption_cd)*>

Change these XML attributes:
  %attrs;
To these:
  ID ID #IMPLIED
  confidentiality IDREFS #IMPLIED
  originator IDREFS #IMPLIED
  xml:lang NMTOKEN #IMPLIED

Change this:
  <!ELEMENT td %Flow;>
to this:
  <!ELEMENT td (#PCDATA | content | link | coded_entry |
  observation_media | paragraph | list | local_markup)*>

change this:
  <!ELEMENT th %Flow;>
to this:
  <!ELEMENT th (#PCDATA | link | caption_cd)*>
=====
-->
  <!--===== XHTML entities used in the XHTML table model =====>
  <xsd:simpleType name="Character">
    <xsd:restriction base="xsd:string"/>
  </xsd:simpleType>
  <!-- a single character from [ISO10646] -->
  <xsd:simpleType name="Length">
    <xsd:restriction base="xsd:string"/>
  </xsd:simpleType>
  <!-- nn for pixels or nn% for percentage length -->
  <xsd:simpleType name="MultiLength">
    <xsd:restriction base="xsd:string"/>
  </xsd:simpleType>
  <!-- pixel, percentage, or relative -->
  <xsd:simpleType name="Number">
    <xsd:restriction base="xsd:string"/>
  </xsd:simpleType>
  <!-- one or more digits -->
  <xsd:simpleType name="Pixels">
    <xsd:restriction base="xsd:string"/>

```

```

</xsd:simpleType>
<!-- integer representing length in pixels -->
<xsd:simpleType name="Text">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<!--===== Tables =====>
<!-- Derived from IETF HTML table standard, see [RFC1942] -->
<!--
The border attribute sets the thickness of the frame around the
table. The default units are screen pixels.

The frame attribute specifies which parts of the frame around
the table should be rendered. The values are not the same as
CALs to avoid a name clash with the valign attribute.
-->
<xsd:simpleType name="TFrame">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="void"/>
    <xsd:enumeration value="above"/>
    <xsd:enumeration value="below"/>
    <xsd:enumeration value="hsides"/>
    <xsd:enumeration value="lhs"/>
    <xsd:enumeration value="rhs"/>
    <xsd:enumeration value="vsides"/>
    <xsd:enumeration value="box"/>
    <xsd:enumeration value="border"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
The rules attribute defines which rules to draw between cells:

If rules is absent then assume:
  "none" if border is absent or border="0" otherwise "all"
-->
<xsd:simpleType name="TRules">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="none"/>
    <xsd:enumeration value="groups"/>
    <xsd:enumeration value="rows"/>
    <xsd:enumeration value="cols"/>
    <xsd:enumeration value="all"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- horizontal alignment attributes for cell contents

char      alignment char, e.g. char=':'
charoff   offset for alignment char
-->
<xsd:attributeGroup name="cellhalign">
  <xsd:attribute name="align">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="left"/>
        <xsd:enumeration value="center"/>
        <xsd:enumeration value="right"/>
        <xsd:enumeration value="justify"/>
        <xsd:enumeration value="char"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
  <xsd:attribute name="char" type="Character"/>
  <xsd:attribute name="charoff" type="Length"/>
</xsd:attributeGroup>
<!-- vertical alignment attributes for cell contents -->
<xsd:attributeGroup name="cellvalign">
  <xsd:attribute name="valign">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="top"/>
        <xsd:enumeration value="middle"/>
        <xsd:enumeration value="bottom"/>
        <xsd:enumeration value="baseline"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
</xsd:attributeGroup>
<xsd:element name="table">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="caption" minOccurs="0"/>
      <xsd:choice>

```

```

        <xsd:element ref="col" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="colgroup" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:choice>
    <xsd:element ref="thead" minOccurs="0"/>
    <xsd:element ref="tfoot" minOccurs="0"/>
    <xsd:choice>
        <xsd:element ref="tbody" maxOccurs="unbounded"/>
        <xsd:element ref="tr" maxOccurs="unbounded"/>
    </xsd:choice>
</xsd:sequence>
<xsd:attributeGroup ref="body_atts"/>
<xsd:attribute name="summary" type="Text"/>
<xsd:attribute name="width" type="Length"/>
<xsd:attribute name="border" type="Pixels"/>
<xsd:attribute name="frame" type="TFrame"/>
<xsd:attribute name="rules" type="TRules"/>
<xsd:attribute name="cellspacing" type="Length"/>
<xsd:attribute name="cellpadding" type="Length"/>
</xsd:complexType>
</xsd:element>
<xsd:element name="caption">
    <xsd:complexType mixed="true">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element ref="link"/>
            <xsd:element ref="caption_cd"/>
        </xsd:choice>
        <xsd:attributeGroup ref="body_atts"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="caption_cd">
    <xsd:complexType>
        <xsd:group ref="CE-cont.model"/>
        <xsd:attributeGroup ref="body_atts"/>
        <xsd:attributeGroup ref="CE-attrib.list"/>
    </xsd:complexType>
</xsd:element>
<!--
colgroup groups a set of col elements. It allows you to group
several semantically related columns together.
-->
<xsd:element name="colgroup">
    <xsd:complexType>
        <xsd:sequence minOccurs="0" maxOccurs="unbounded">
            <xsd:element ref="col"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="body_atts"/>
        <xsd:attribute name="span" type="Number" default="1"/>
        <xsd:attribute name="width" type="MultiLength"/>
        <xsd:attributeGroup ref="cellhalign"/>
        <xsd:attributeGroup ref="cellvalign"/>
    </xsd:complexType>
</xsd:element>
<!--
col elements define the alignment properties for cells in
one or more columns.

The width attribute specifies the width of the columns, e.g.

    width=64          width in screen pixels
    width=0.5*        relative width of 0.5

The span attribute causes the attributes of one
col element to apply to more than one column.
-->
<xsd:element name="col">
    <xsd:complexType>
        <xsd:attributeGroup ref="body_atts"/>
        <xsd:attribute name="span" type="Number" default="1"/>
        <xsd:attribute name="width" type="MultiLength"/>
        <xsd:attributeGroup ref="cellhalign"/>
        <xsd:attributeGroup ref="cellvalign"/>
    </xsd:complexType>
</xsd:element>
<!--
Use thead to duplicate headers when breaking table
across page boundaries, or for static headers when
tbody sections are rendered in scrolling panel.

Use tfoot to duplicate footers when breaking table
across page boundaries, or for static footers when
tbody sections are rendered in scrolling panel.

```

```

    Use multiple tbody sections when rules are needed
    between groups of table rows.
-->
<xsd:element name="thead">
  <xsd:complexType>
    <xsd:sequence maxOccurs="unbounded">
      <xsd:element ref="tr"/>
    </xsd:sequence>
    <xsd:attributeGroup ref="body_atts"/>
    <xsd:attributeGroup ref="cellhalign"/>
    <xsd:attributeGroup ref="cellvalign"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="tfoot">
  <xsd:complexType>
    <xsd:sequence maxOccurs="unbounded">
      <xsd:element ref="tr"/>
    </xsd:sequence>
    <xsd:attributeGroup ref="body_atts"/>
    <xsd:attributeGroup ref="cellhalign"/>
    <xsd:attributeGroup ref="cellvalign"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="tbody">
  <xsd:complexType>
    <xsd:sequence maxOccurs="unbounded">
      <xsd:element ref="tr"/>
    </xsd:sequence>
    <xsd:attributeGroup ref="body_atts"/>
    <xsd:attributeGroup ref="cellhalign"/>
    <xsd:attributeGroup ref="cellvalign"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="tr">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded">
      <xsd:element ref="th"/>
      <xsd:element ref="td"/>
    </xsd:choice>
    <xsd:attributeGroup ref="body_atts"/>
    <xsd:attributeGroup ref="cellhalign"/>
    <xsd:attributeGroup ref="cellvalign"/>
  </xsd:complexType>
</xsd:element>
<!-- Scope is simpler than headers attribute for common tables -->
<xsd:simpleType name="Scope">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="row"/>
    <xsd:enumeration value="col"/>
    <xsd:enumeration value="rowgroup"/>
    <xsd:enumeration value="colgroup"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- th is for headers, td for data and for cells acting as both -->
<xsd:element name="th">
  <xsd:complexType mixed="true">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element ref="link"/>
      <xsd:element ref="caption_cd"/>
    </xsd:choice>
    <xsd:attributeGroup ref="body_atts"/>
    <xsd:attribute name="abbr" type="Text"/>
    <xsd:attribute name="axis" type="xsd:string"/>
    <xsd:attribute name="headers" type="xsd:IDREFS"/>
    <xsd:attribute name="scope" type="Scope"/>
    <xsd:attribute name="rowspan" type="Number" default="1"/>
    <xsd:attribute name="colspan" type="Number" default="1"/>
    <xsd:attributeGroup ref="cellhalign"/>
    <xsd:attributeGroup ref="cellvalign"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="td">
  <xsd:complexType mixed="true">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:group ref="entries"/>
      <xsd:element ref="paragraph"/>
      <xsd:element ref="list"/>
    </xsd:choice>
    <xsd:attributeGroup ref="body_atts"/>
    <xsd:attribute name="abbr" type="Text"/>

```

```

        <xsd:attribute name="axis" type="xsd:string"/>
        <xsd:attribute name="headers" type="xsd:IDREFS"/>
        <xsd:attribute name="scope" type="Scope"/>
        <xsd:attribute name="rowspan" type="Number" default="1"/>
        <xsd:attribute name="colspan" type="Number" default="1"/>
        <xsd:attributeGroup ref="cellhalign"/>
        <xsd:attributeGroup ref="cellvalign"/>
    </xsd:complexType>
</xsd:element>
</xsd:schema>

```

2.2 CDA Header (header_1.0.xsd)

This is the CDA Header schema. It imports the HL7 Version 3 data types, Release 1.

```

<?xml version="1.0"?>
<!--
=====
HL7 Clinical Document Architecture, Version 1.0

CDA Header XML Schema (included by levelone_1.0.xsd)

Derived from HL7 Reference Information Model, Version 0.98

corresponding DTD Last Revised: September 18, 2000
XML Schema last revised:
    2004-01-05 (Kai U. Heitmann):
        Namespace declarations have been eliminated

=====
-->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
    <!--
=====
Import the V3 data type DTD

(The following system id must be changed to point to the location
of the V3DT file on your system.)
=====
-->
    <xsd:include schemaLocation="v3dt_1.0.xsd"/>
    <!--
=====
Common attributes

=====
-->
    <xsd:attributeGroup name="common_atts">
        <xsd:attribute name="ID" type="xsd:ID"/>
    </xsd:attributeGroup>
    <!--
=====
The base RIM class for the DTD is Document_service

=====
-->
    <xsd:element name="clinical_document_header">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="id"/>
                <xsd:element ref="set_id" minOccurs="0"/>
                <xsd:element ref="version_nbr" minOccurs="0"/>
                <xsd:element ref="document_type_cd"/>
                <xsd:element ref="service_tmr" minOccurs="0"/>
                <xsd:element ref="origination_dttm"/>
                <xsd:element ref="copy_dttm" minOccurs="0"/>
                <xsd:element ref="confidentiality_cd" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element ref="document_relationship" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element ref="fulfills_order" minOccurs="0"/>
                <xsd:element ref="patient_encounter" minOccurs="0"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

```

```

<xsd:element ref="authenticator" minOccurs="0" maxOccurs="unbounded"/>
<xsd:element ref="legal_authenticator" minOccurs="0"/>
<xsd:element ref="intended_recipient" minOccurs="0" maxOccurs="unbounded"/>
<xsd:element ref="originator" minOccurs="0" maxOccurs="unbounded"/>
<xsd:element ref="originating_organization" minOccurs="0"/>
<xsd:element ref="transcriptionist" minOccurs="0"/>
<xsd:element ref="provider" maxOccurs="unbounded"/>
<xsd:element ref="service_actor" minOccurs="0" maxOccurs="unbounded"/>
<xsd:element ref="patient"/>
<xsd:element ref="originating_device" minOccurs="0" maxOccurs="unbounded"/>
<xsd:element ref="service_target" minOccurs="0" maxOccurs="unbounded"/>
<xsd:element ref="local_header" minOccurs="0" maxOccurs="unbounded"/>
</xsd:sequence>
<xsd:attributeGroup ref="common_atts"/>
<xsd:attribute name="HL7-NAME" type="xsd:string"
fixed="document_service_as_clinical_document_header"/>
<xsd:attribute name="T" type="xsd:string" fixed="service"/>
<xsd:attribute name="RIM-VERSION" type="xsd:string" fixed="0.98"/>
</xsd:complexType>
</xsd:element>
<!--
=====
=====
RIM components (classes, attributes, associations) nested under
clinical_document_header

There are four logical components of the CDA Header:
(1) Document information;
(2) Encounter data;
(3) Service actors (such as providers);
(4) Service targets (such as patients).

The four components are presented in this order, similar to their
order in the CDA Header Hierarchical Description.
=====
=====
-->
<!--
=====
=====
Document Information

Document information identifies the document, defines confidentiality
status, and describes relationships to other documents and orders.
=====
=====
-->
<!--
=====
=====
Document Information :: Document Identification

Elements declared in this section include:
<id>, <set_id>, <version_nbr>, <document_type_cd>
=====
-->
<xsd:element name="id">
  <xsd:complexType>
    <xsd:group ref="II-cont.model"/>
    <xsd:attributeGroup ref="II-attrib.list"/>
    <xsd:attributeGroup ref="common_atts"/>
    <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="id"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="set_id">
  <xsd:complexType>
    <xsd:group ref="II-cont.model"/>
    <xsd:attributeGroup ref="II-attrib.list"/>
    <xsd:attributeGroup ref="common_atts"/>
    <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="set_id"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="version_nbr">
  <xsd:complexType>
    <xsd:group ref="INT-cont.model"/>
    <xsd:attributeGroup ref="INT-attrib.list"/>
    <xsd:attributeGroup ref="common_atts"/>
    <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="version_nbr"/>
  </xsd:complexType>
</xsd:element>
<!--
=====
=====

```

Document Information :: Document Time Stamps

Elements declared in this section include:

<service_tmr>, <origination_dttm>, <copy_dttm>

```

=====
-->
  <xsd:element name="document_type_cd">
    <xsd:complexType>
      <xsd:group ref="CE-cont.model"/>
      <xsd:attributeGroup ref="CE-attrib.list"/>
      <xsd:attributeGroup ref="common_atts"/>
      <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="service_cd"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="service_tmr">
    <xsd:complexType>
      <xsd:group ref="GTS-cont.model"/>
      <xsd:attributeGroup ref="GTS-attrib.list"/>
      <xsd:attributeGroup ref="common_atts"/>
      <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="service_tmr"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="origination_dttm">
    <xsd:complexType>
      <xsd:group ref="TS-cont.model"/>
      <xsd:attributeGroup ref="TS-attrib.list"/>
      <xsd:attributeGroup ref="common_atts"/>
      <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="origination_dttm"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="copy_dttm">
    <xsd:complexType>
      <xsd:group ref="TS-cont.model"/>
      <xsd:attributeGroup ref="TS-attrib.list"/>
      <xsd:attributeGroup ref="common_atts"/>
      <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="copy_dttm"/>
    </xsd:complexType>
  </xsd:element>
<!--
=====

```

Document Information :: Document Confidentiality

Elements declared in this section include:

<confidentiality_cd>

```

=====
-->
  <xsd:element name="confidentiality_cd">
    <xsd:complexType>
      <xsd:group ref="CE-cont.model"/>
      <xsd:attributeGroup ref="CE-attrib.list"/>
      <xsd:attributeGroup ref="common_atts"/>
      <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="confidentiality_cd"/>
    </xsd:complexType>
  </xsd:element>
<!--
=====

```

Document Information :: Document Relationships

Elements declared in this section include:

<document_relationship>, <document_relationship.type_cd>,
<related_document>, <fulfills_order>, <fulfills_order.type_cd>,
<order>

```

=====
-->
  <xsd:element name="document_relationship">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="document_relationship.type_cd"/>
        <xsd:element ref="related_document"/>
        <xsd:element ref="local_header" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attributeGroup ref="common_atts"/>
      <xsd:attribute name="HL7-NAME" type="xsd:string"
fixed="is_source_for_service_relationship"/>
      <xsd:attribute name="T" type="xsd:string" fixed="service_relationship"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="document_relationship.type_cd">
    <xsd:complexType>
      <xsd:group ref="CS-cont.model"/>
      <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="CS"/>
    </xsd:complexType>
  </xsd:element>

```

```

        <xsd:attribute name="V" use="required">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="APND"/>
              <xsd:enumeration value="RPLC"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="ST"/>
        <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="code"/>
        <xsd:attribute name="DN" type="xsd:string"/>
        <xsd:attribute name="DN-T" type="xsd:NMTOKEN" fixed="ST"/>
        <xsd:attribute name="DN-HL7_NAME" type="xsd:string" fixed="displayName"/>
        <xsd:attributeGroup ref="common_atts"/>
        <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="type_cd"/>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="related_document">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="id"/>
          <xsd:element ref="set_id" minOccurs="0"/>
          <xsd:element ref="version_nbr" minOccurs="0"/>
          <xsd:element ref="local_header" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="common_atts"/>
        <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="has_target_service"/>
        <xsd:attribute name="T" type="xsd:string" fixed="service"/>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="fulfills_order">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="fulfills_order.type_cd"/>
          <xsd:element ref="order" maxOccurs="unbounded"/>
          <xsd:element ref="local_header" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="common_atts"/>
        <xsd:attribute name="HL7-NAME" type="xsd:string"
fixed="is_source_for_service_relationship"/>
        <xsd:attribute name="T" type="xsd:string" fixed="service_relationship"/>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="fulfills_order.type_cd">
      <xsd:complexType>
        <xsd:group ref="CS-cont.model"/>
        <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="CS"/>
        <xsd:attribute name="V" type="xsd:string" fixed="FLFS"/>
        <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="ST"/>
        <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="code"/>
        <xsd:attribute name="DN" type="xsd:string"/>
        <xsd:attribute name="DN-T" type="xsd:NMTOKEN" fixed="ST"/>
        <xsd:attribute name="DN-HL7_NAME" type="xsd:string" fixed="displayName"/>
        <xsd:attributeGroup ref="common_atts"/>
        <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="type_cd"/>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="order">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="id"/>
          <xsd:element ref="local_header" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="common_atts"/>
        <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="has_target_service"/>
        <xsd:attribute name="T" type="xsd:string" fixed="service"/>
      </xsd:complexType>
    </xsd:element>
  <!--
=====
=====
Encounter Data

Encounter data describes the setting in which the documented encounter occurred.

Elements declared in this section include:
<patient_encounter>, <practice_setting_cd>, <encounter_tmr>, <service_location>, <addr>
=====
-->
    <xsd:element name="patient_encounter">

```



```

    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="id" minOccurs="0"/>
        <xsd:element ref="practice_setting_cd" minOccurs="0"/>
        <xsd:element ref="encounter_tmr"/>
        <xsd:element ref="service_location" minOccurs="0"/>
        <xsd:element ref="local_header" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attributeGroup ref="common_atts"/>
      <xsd:attribute name="HL7-NAME" type="xsd:string"
fixed="is_assigned_to_patient_encounter"/>
      <xsd:attribute name="T" type="xsd:string" fixed="patient_encounter"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="practice_setting_cd">
    <xsd:complexType>
      <xsd:group ref="CE-cont.model"/>
      <xsd:attributeGroup ref="CE-attrib.list"/>
      <xsd:attributeGroup ref="common_atts"/>
      <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="practice_setting_cd"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="encounter_tmr">
    <xsd:complexType>
      <xsd:group ref="IVL_TS-cont.model"/>
      <xsd:attributeGroup ref="IVL_TS-attrib.list"/>
      <xsd:attributeGroup ref="common_atts"/>
      <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="encounter_tmr"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="service_location">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="id" minOccurs="0"/>
        <xsd:element ref="addr" minOccurs="0"/>
        <xsd:element ref="local_header" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attributeGroup ref="common_atts"/>
      <xsd:attribute name="HL7-NAME" type="xsd:string"
fixed="has_master_patient_service_location"/>
      <xsd:attribute name="T" type="xsd:string" fixed="master_patient_service_location"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="addr">
    <xsd:complexType>
      <xsd:group ref="AD-cont.model"/>
      <xsd:attributeGroup ref="AD-attrib.list"/>
      <xsd:attributeGroup ref="common_atts"/>
      <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="addr"/>
    </xsd:complexType>
  </xsd:element>
  <!--
=====
=====
Service Actors

Service actors include those who authenticate the document, those intended
to receive a copy of the document, document originators and transcriptionists,
and health care providers who participated in the service(s) being documented.
=====
-->
  <!--
=====
Service Actors :: People Responsible for a Clinical Document

Elements declared in this section include:
<person>, <person_name>, <effective_tmr>, <nm>, <person_name.type_cd>, <telecom>
=====
-->
  <xsd:element name="person">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="id" maxOccurs="unbounded"/>
        <xsd:element ref="person_name" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="addr" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="telecom" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="local_header" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attributeGroup ref="common_atts"/>
      <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="participation_of_person"/>

```

```

        <xsd:attribute name="T" type="xsd:string" fixed="person"/>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="person_name">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="effective_tmr" minOccurs="0"/>
          <xsd:element ref="nm"/>
          <xsd:element ref="person_name.type_cd" minOccurs="0"/>
          <xsd:element ref="local_header" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="common_atts"/>
        <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="has_person_name"/>
        <xsd:attribute name="T" type="xsd:string" fixed="person_name"/>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="effective_tmr">
      <xsd:complexType>
        <xsd:group ref="IVL_TS-cont.model"/>
        <xsd:attributeGroup ref="IVL_TS-attrib.list"/>
        <xsd:attributeGroup ref="common_atts"/>
        <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="effective_tmr"/>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="nm">
      <xsd:complexType>
        <xsd:group ref="PN-cont.model"/>
        <xsd:attributeGroup ref="PN-attrib.list"/>
        <xsd:attributeGroup ref="common_atts"/>
        <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="nm"/>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="person_name.type_cd">
      <xsd:complexType>
        <xsd:group ref="CE-cont.model"/>
        <xsd:attributeGroup ref="CE-attrib.list"/>
        <xsd:attributeGroup ref="common_atts"/>
        <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="type_cd"/>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="telecom">
      <xsd:complexType>
        <xsd:group ref="TEL-cont.model"/>
        <xsd:attributeGroup ref="TEL-attrib.list"/>
        <xsd:attributeGroup ref="common_atts"/>
        <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="phon"/>
      </xsd:complexType>
    </xsd:element>
    <!--
=====
Service Actors :: Authenticators

Elements declared in this section include:
<authenticator>, <authenticator.type_cd>, <participation_tmr>,
<signature_cd>, <legal_authenticator>, <legal_authenticator.type_cd>
=====
-->
    <xsd:element name="authenticator">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="authenticator.type_cd"/>
          <xsd:element ref="participation_tmr"/>
          <xsd:element ref="signature_cd"/>
          <xsd:element ref="person"/>
          <xsd:element ref="local_header" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="common_atts"/>
        <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="has_service_actor"/>
        <xsd:attribute name="T" type="xsd:string" fixed="service_actor"/>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="authenticator.type_cd">
      <xsd:complexType>
        <xsd:group ref="CS-cont.model"/>
        <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="CS"/>
        <xsd:attribute name="V" type="xsd:string" fixed="VRF"/>
        <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="ST"/>
        <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="code"/>
        <xsd:attribute name="DN" type="xsd:string"/>
        <xsd:attribute name="DN-T" type="xsd:NMTOKEN" fixed="ST"/>
        <xsd:attribute name="DN-HL7_NAME" type="xsd:string" fixed="displayName"/>

```

```

        <xsd:attributeGroup ref="common_atts"/>
        <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="type_cd"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="participation_tmr">
    <xsd:complexType>
        <xsd:group ref="IVL_TS-cont.model"/>
        <xsd:attributeGroup ref="IVL_TS-attrib.list"/>
        <xsd:attributeGroup ref="common_atts"/>
        <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="tmr"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="signature_cd">
    <xsd:complexType>
        <xsd:group ref="CS-cont.model"/>
        <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="CS"/>
        <xsd:attribute name="V" default="S">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="S"/>
                    <xsd:enumeration value="X"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="ST"/>
        <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="code"/>
        <xsd:attribute name="DN" type="xsd:string"/>
        <xsd:attribute name="DN-T" type="xsd:NMTOKEN" fixed="ST"/>
        <xsd:attribute name="DN-HL7_NAME" type="xsd:string" fixed="displayName"/>
        <xsd:attributeGroup ref="common_atts"/>
        <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="signature_cd"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="legal_authenticator">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="legal_authenticator.type_cd"/>
            <xsd:element ref="participation_tmr"/>
            <xsd:element ref="signature_cd"/>
            <xsd:element ref="person"/>
            <xsd:element ref="local_header" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="common_atts"/>
        <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="has_service_actor"/>
        <xsd:attribute name="T" type="xsd:string" fixed="service_actor"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="legal_authenticator.type_cd">
    <xsd:complexType>
        <xsd:group ref="CS-cont.model"/>
        <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="CS"/>
        <xsd:attribute name="V" type="xsd:string" fixed="SPV"/>
        <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="ST"/>
        <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="code"/>
        <xsd:attribute name="DN" type="xsd:string"/>
        <xsd:attribute name="DN-T" type="xsd:NMTOKEN" fixed="ST"/>
        <xsd:attribute name="DN-HL7_NAME" type="xsd:string" fixed="displayName"/>
        <xsd:attributeGroup ref="common_atts"/>
        <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="type_cd"/>
    </xsd:complexType>
</xsd:element>
<!--
=====
Service Actors :: Intended Recipients

Elements declared in this section include:
<intended_recipient>, <intended_recipient.type_cd>
=====
-->
    <xsd:element name="intended_recipient">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="intended_recipient.type_cd"/>
                <xsd:element ref="person"/>
                <xsd:element ref="local_header" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:attributeGroup ref="common_atts"/>
            <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="has_service_actor"/>
            <xsd:attribute name="T" type="xsd:string" fixed="service_actor"/>
        </xsd:complexType>
    </xsd:element>

```

```

<xsd:element name="intended_recipient.type_cd">
  <xsd:complexType>
    <xsd:group ref="CS-cont.model"/>
    <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="CS"/>
    <xsd:attribute name="V" type="xsd:string" fixed="TRC"/>
    <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="ST"/>
    <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="code"/>
    <xsd:attribute name="DN" type="xsd:string"/>
    <xsd:attribute name="DN-T" type="xsd:NMTOKEN" fixed="ST"/>
    <xsd:attribute name="DN-HL7_NAME" type="xsd:string" fixed="displayName"/>
    <xsd:attributeGroup ref="common_atts"/>
    <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="type_cd"/>
  </xsd:complexType>
</xsd:element>
<!--
=====
Service Actors :: Originators

Elements declared in this section include:
<originator>, <originator.type_cd>, <originating_organization>,
<originating_organization.type_cd>, <organization>, <organization.nm>
=====
-->
<xsd:element name="originator">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="originator.type_cd"/>
      <xsd:element ref="participation_tmr"/>
      <xsd:element ref="person"/>
      <xsd:element ref="local_header" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attributeGroup ref="common_atts"/>
    <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="has_service_actor"/>
    <xsd:attribute name="T" type="xsd:string" fixed="service_actor"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="originator.type_cd">
  <xsd:complexType>
    <xsd:group ref="CS-cont.model"/>
    <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="CS"/>
    <xsd:attribute name="V" type="xsd:string" fixed="AUT"/>
    <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="ST"/>
    <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="code"/>
    <xsd:attribute name="DN" type="xsd:string"/>
    <xsd:attribute name="DN-T" type="xsd:NMTOKEN" fixed="ST"/>
    <xsd:attribute name="DN-HL7_NAME" type="xsd:string" fixed="displayName"/>
    <xsd:attributeGroup ref="common_atts"/>
    <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="type_cd"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="originating_organization">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="originating_organization.type_cd"/>
      <xsd:element ref="organization"/>
      <xsd:element ref="local_header" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attributeGroup ref="common_atts"/>
    <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="has_service_actor"/>
    <xsd:attribute name="T" type="xsd:string" fixed="service_actor"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="originating_organization.type_cd">
  <xsd:complexType>
    <xsd:group ref="CS-cont.model"/>
    <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="CS"/>
    <xsd:attribute name="V" type="xsd:string" fixed="CST"/>
    <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="ST"/>
    <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="code"/>
    <xsd:attribute name="DN" type="xsd:string"/>
    <xsd:attribute name="DN-T" type="xsd:NMTOKEN" fixed="ST"/>
    <xsd:attribute name="DN-HL7_NAME" type="xsd:string" fixed="displayName"/>
    <xsd:attributeGroup ref="common_atts"/>
    <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="type_cd"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="organization">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="id" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="organization.nm" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>

```

```

        <xsd:element ref="addr" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="local_header" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attributeGroup ref="common_atts"/>
    <xsd:attribute name="HL7-NAME" type="xsd:string"
fixed="participation_of_organization"/>
    <xsd:attribute name="T" type="xsd:string" fixed="organization"/>
</xsd:complexType>
</xsd:element>
<xsd:element name="organization.nm">
    <xsd:complexType>
        <xsd:group ref="ON-cont.model"/>
        <xsd:attributeGroup ref="ON-attrib.list"/>
        <xsd:attributeGroup ref="common_atts"/>
        <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="nm"/>
    </xsd:complexType>
</xsd:element>
<!--
=====
Service Actors :: Transcriptionist

Elements declared in this section include:
<transcriptionist>, <transcriptionist.type_cd>
=====
-->
    <xsd:element name="transcriptionist">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="transcriptionist.type_cd"/>
                <xsd:element ref="participation_tmr" minOccurs="0"/>
                <xsd:element ref="person"/>
                <xsd:element ref="local_header" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:attributeGroup ref="common_atts"/>
            <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="has_service_actor"/>
            <xsd:attribute name="T" type="xsd:string" fixed="service_actor"/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="transcriptionist.type_cd">
        <xsd:complexType>
            <xsd:group ref="CS-cont.model"/>
            <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="CS"/>
            <xsd:attribute name="V" type="xsd:string" fixed="ENT"/>
            <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="ST"/>
            <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="code"/>
            <xsd:attribute name="DN" type="xsd:string"/>
            <xsd:attribute name="DN-T" type="xsd:NMTOKEN" fixed="ST"/>
            <xsd:attribute name="DN-HL7_NAME" type="xsd:string" fixed="displayName"/>
            <xsd:attributeGroup ref="common_atts"/>
            <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="type_cd"/>
        </xsd:complexType>
    </xsd:element>
<!--
=====
Service Actors :: Healthcare providers

Elements declared in this section include:
<provider>, <provider.type_cd>, <function_cd>
=====
-->
    <xsd:element name="provider">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="provider.type_cd"/>
                <xsd:element ref="function_cd" minOccurs="0"/>
                <xsd:element ref="participation_tmr" minOccurs="0"/>
                <xsd:element ref="person"/>
                <xsd:element ref="local_header" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:attributeGroup ref="common_atts"/>
            <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="has_service_actor"/>
            <xsd:attribute name="T" type="xsd:string" fixed="service_actor"/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="provider.type_cd">
        <xsd:complexType>
            <xsd:group ref="CS-cont.model"/>
            <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="CS"/>
            <xsd:attribute name="V" default="PRF"/>
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">

```

```

        <xsd:enumeration value="ASS"/>
        <xsd:enumeration value="CON"/>
        <xsd:enumeration value="PRF"/>
    </xsd:restriction>
</xsd:simpleType>
</xsd:attribute>
<xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="ST"/>
<xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="code"/>
<xsd:attribute name="DN" type="xsd:string"/>
<xsd:attribute name="DN-T" type="xsd:NMTOKEN" fixed="ST"/>
<xsd:attribute name="DN-HL7_NAME" type="xsd:string" fixed="displayName"/>
<xsd:attributeGroup ref="common_atts"/>
<xsd:attribute name="HL7-NAME" type="xsd:string" fixed="type_cd"/>
</xsd:complexType>
</xsd:element>
<xsd:element name="function_cd">
    <xsd:complexType>
        <xsd:group ref="CE-cont.model"/>
        <xsd:attributeGroup ref="CE-attrib.list"/>
        <xsd:attributeGroup ref="common_atts"/>
        <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="function_cd"/>
    </xsd:complexType>
</xsd:element>
<!--
=====
Service Actors :: Other Service Actors

Elements declared in this section include:
<service_actor>, <service_actor.type_cd>
=====
-->
    <xsd:element name="service_actor">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="service_actor.type_cd"/>
                <xsd:element ref="participation_tmr" minOccurs="0"/>
                <xsd:element ref="signature_cd" minOccurs="0"/>
                <xsd:choice>
                    <xsd:element ref="person"/>
                    <xsd:element ref="organization"/>
                </xsd:choice>
                <xsd:element ref="local_header" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:attributeGroup ref="common_atts"/>
            <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="has_service_actor"/>
            <xsd:attribute name="T" type="xsd:string" fixed="service_actor"/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="service_actor.type_cd">
        <xsd:complexType>
            <xsd:group ref="CE-cont.model"/>
            <xsd:attributeGroup ref="CE-attrib.list"/>
            <xsd:attributeGroup ref="common_atts"/>
            <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="type_cd"/>
        </xsd:complexType>
    </xsd:element>
    <!--
=====
Service Targets

Service targets include the patient, other significant participants (such
as family members), and those devices that may have originated portions
of the document.
=====
-->
    <!--
=====
Service Targets :: Patient

Elements declared in this section include:
<patient>, <patient.type_cd>, <assigned_identifier>, <is_known_by>,
<birth_dttm>, <administrative_gender_cd>
=====
-->
    <xsd:element name="patient">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="patient.type_cd"/>
                <xsd:element ref="participation_tmr" minOccurs="0"/>

```

```

        <xsd:element ref="person"/>
        <xsd:element ref="is_known_by" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="birth_dttm" minOccurs="0"/>
        <xsd:element ref="administrative_gender_cd" minOccurs="0"/>
        <xsd:element ref="local_header" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attributeGroup ref="common_atts"/>
    <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="has_service_target"/>
    <xsd:attribute name="T" type="xsd:string" fixed="service_target"/>
</xsd:complexType>
</xsd:element>
<xsd:element name="patient.type_cd">
    <xsd:complexType>
        <xsd:group ref="CS-cont.model"/>
        <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="CS"/>
        <xsd:attribute name="V" default="PATSBJ">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="PAT"/>
                    <xsd:enumeration value="PATSBJ"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="ST"/>
        <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="code"/>
        <xsd:attribute name="DN" type="xsd:string"/>
        <xsd:attribute name="DN-T" type="xsd:NMTOKEN" fixed="ST"/>
        <xsd:attribute name="DN-HL7_NAME" type="xsd:string" fixed="displayName"/>
        <xsd:attributeGroup ref="common_atts"/>
        <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="type_cd"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="is_known_by">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="id" maxOccurs="unbounded"/>
            <xsd:element ref="is_known_to"/>
            <xsd:element ref="local_header" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="common_atts"/>
        <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="is_known_by"/>
        <xsd:attribute name="T" type="xsd:string" fixed="person_provider_association"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="is_known_to">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="id" maxOccurs="unbounded"/>
            <xsd:element ref="local_header" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="common_atts"/>
        <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="is_known_to"/>
        <xsd:attribute name="T" type="xsd:string" fixed="healthcare_service_provider"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="birth_dttm">
    <xsd:complexType>
        <xsd:group ref="TS-cont.model"/>
        <xsd:attributeGroup ref="TS-attrib.list"/>
        <xsd:attributeGroup ref="common_atts"/>
        <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="birth_dttm"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="administrative_gender_cd">
    <xsd:complexType>
        <xsd:group ref="CE-cont.model"/>
        <xsd:attributeGroup ref="CE-attrib.list"/>
        <xsd:attributeGroup ref="common_atts"/>
        <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="administrative_gender_cd"/>
    </xsd:complexType>
</xsd:element>
<!--
=====
Service Targets :: Originating Device

Elements declared in this section include:
<originating_device>, <originating_device.type_cd>, <device>, <responsibility>,
<responsibility.type_cd>, <responsibility_tmr>
=====
-->
    <xsd:element name="originating_device">

```

```

    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="originating_device.type_cd"/>
        <xsd:element ref="participation_tmr" minOccurs="0"/>
        <xsd:element ref="device"/>
        <xsd:element ref="local_header" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attributeGroup ref="common_atts"/>
      <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="has_service_target"/>
      <xsd:attribute name="T" type="xsd:string" fixed="service_target"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="originating_device.type_cd">
    <xsd:complexType>
      <xsd:group ref="CS-cont.model"/>
      <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="CS"/>
      <xsd:attribute name="V" type="xsd:string" fixed="ODV"/>
      <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="ST"/>
      <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="code"/>
      <xsd:attribute name="DN" type="xsd:string"/>
      <xsd:attribute name="DN-T" type="xsd:NMTOKEN" fixed="ST"/>
      <xsd:attribute name="DN-HL7_NAME" type="xsd:string" fixed="displayName"/>
      <xsd:attributeGroup ref="common_atts"/>
      <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="type_cd"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="device">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="id" maxOccurs="unbounded"/>
        <xsd:element ref="responsibility" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="local_header" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attributeGroup ref="common_atts"/>
      <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="participation_of_material"/>
      <xsd:attribute name="T" type="xsd:string" fixed="device"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="responsibility">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="responsibility.type_cd" minOccurs="0"/>
        <xsd:element ref="responsibility_tmr" minOccurs="0"/>
        <xsd:element ref="person"/>
        <xsd:element ref="local_header" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attributeGroup ref="common_atts"/>
      <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="is_the_responsibility"/>
      <xsd:attribute name="T" type="xsd:string" fixed="responsibility"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="responsibility.type_cd">
    <xsd:complexType>
      <xsd:group ref="CE-cont.model"/>
      <xsd:attributeGroup ref="CE-attrib.list"/>
      <xsd:attributeGroup ref="common_atts"/>
      <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="type_cd"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="responsibility_tmr">
    <xsd:complexType>
      <xsd:group ref="IVL_TS-cont.model"/>
      <xsd:attributeGroup ref="IVL_TS-attrib.list"/>
      <xsd:attributeGroup ref="common_atts"/>
      <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="tmr"/>
    </xsd:complexType>
  </xsd:element>
<!--
=====
Service Targets :: Other Service Targets

Elements declared in this section include:
<service_target>, <service_target.type_cd>
=====
-->
  <xsd:element name="service_target">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="service_target.type_cd"/>
        <xsd:element ref="participation_tmr" minOccurs="0"/>
        <xsd:element ref="person"/>

```



```

        <xsd:element ref="local_header" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attributeGroup ref="common_atts"/>
      <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="has_service_target"/>
      <xsd:attribute name="T" type="xsd:string" fixed="service_target"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="service_target.type_cd">
    <xsd:complexType>
      <xsd:group ref="CE-cont.model"/>
      <xsd:attributeGroup ref="CE-attrib.list"/>
      <xsd:attributeGroup ref="common_atts"/>
      <xsd:attribute name="HL7-NAME" type="xsd:string" fixed="type_cd"/>
    </xsd:complexType>
  </xsd:element>
  <!--
=====
=====
Local Header Information

Locally-defined markup must be used when local semantics have no
corresponding representation in the CDA specification. CDA seeks to
standardize the highest level of shared meaning while providing a
clean and standard mechanism for tagging meaning that is not shared.
This is achieved with the CDA <local_header> element.

The <local_header> element is optionally repeating, and recursive. The
"descriptor" attribute describes the element, and the value can be
drawn from a local vocabulary domain. The "ignore" attribute tells
the receiver to ignore just the <local_header> tag (ignore="markup"),
or to ignore the <local_header> tag and all contained content
(ignore="all"). The "render" attribute indicates how the sender would
render the contents. The value can be drawn from a local vocabulary
domain. The language of contained character data can be specified
using the xml:lang attribute (see 3.3.2.4.1 Character data). The
nested <local_attr> element is provided to make it easier to map
local XML attribute values into local markup.
=====
=====
-->
  <xsd:element name="local_header" type="local_header-cont.model"/>
  <xsd:complexType name="local_header-cont.model" mixed="true">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element ref="local_header"/>
      <xsd:element ref="local_attr"/>
    </xsd:choice>
    <xsd:attribute name="ignore" default="markup">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="all"/>
          <xsd:enumeration value="markup"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
    <xsd:attribute name="descriptor" type="xsd:string"/>
    <xsd:attribute name="render" type="xsd:string"/>
    <xsd:attributeGroup ref="common_atts"/>
    <!-- deprecated
    <xsd:attribute ref="xml:lang" use="optional"/>
    -->
  </xsd:complexType>
  <xsd:element name="local_attr">
    <xsd:complexType>
      <xsd:attribute name="name" type="xsd:NMTOKEN" use="required"/>
      <xsd:attribute name="value" type="xsd:string" use="required"/>
      <xsd:attributeGroup ref="common_atts"/>
      <!-- deprecated
      <xsd:attribute ref="xml:lang" use="optional"/>
      -->
    </xsd:complexType>
  </xsd:element>
</xsd:schema>

```

2.3 HL7 Version 3 Data Types, Release 1 (v3dt_1.0.xsd)

Following the XML schema definition of the HL7 Version 3 data types, release 1. The version used here – the release 1 of the data types – was published together with the CDA DTDs in November 2000. At the time of this writing, the HL7 version 3 data types have had several re-definitions and ballots and are now different in some definitions.

```
<?xml version="1.0"?>
<!--
  These are the v3 data types version 1.0 as of 2000-06-27

  Typical usage within a XML Schema for example for CDA...

  <xsd:import namespace="urn:..." schemaLocation="v3dt_1.0.xsd"/>

  please also refer to the comments in the corresponding DTD
  for the v3 data types version 1.0, named v3dt_1.0.dtd
-->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<!--
  the following type declarations are in the order they are specified
  in the abstract document (except for some of the "subsidiary" types,
  such as ADXP for AD, which are defined before their "principle" type)
-->
<!--
  Each datatype definition contains "HL7 Processing Rules", which are
  statements of required application level validity checks over and above
  what the DTD/Schema expresses. The rules are written as XPath expressions.
  The intention is that the set of rules for each datatype will be
  embedded in a fixed attribute and a receiving application could then
  evaluate them and if any fail the receiving application would know
  that the instance wasn't legal. At present, some of the rules
  are still expressed in the syntax that was used for this purpose
  for the HIMSS demo.
-->
<!--
  The ANY in the HIMSS demo DTD was ANY in the sense of "any type",
  which was implemented as, essentially, the union of properties from
  all datatypes.

  Do we still need an "any type" datatype? My guess is the answer
  is yes (for observation.value, anywhere else?)...but would
  like more input before going there.
-->
<!--
  Revision history

  2004-01-05 (Kai U. Heitmann):
    Namespace declarations have been eliminated

  2002-02-25 (Kai U. Heitmann):
    Separate v3dt schema, used and tested with the CDA schemas
    (levelone_1.0.xsd and header_1.0.xsd). Revisited some
    of the DTD comments

  2001-10 (Kai U. Heitmann):
    First XML Schema draft, created with xsbrowser tool (www.xsbrowser.org)
    amalgamating all CDA DTDs (levelone_1.0.dtd, header_1.0.dtd and v3dt_1.0.dtd)
    into a single XML schema.

  2000-06-27:
    a) A long history in which I didn't explicitly record revision comments.
    b) assigned this revision number 0.9

  2000-06-28:
    a) added this revision history section
    b) added a "usage" comment to the top of the DTD which includes
       the specification of both a full URI and an FPI for this DTD
    c) removed UPDATE MODE (and the param entities needed to
       support it). The functionality will be handled at the
       general XML ITS level.
    d) changed the definition of ED:T to be the enumeration (ST|ED)
       instead of the #FIXED value ED...made ED the default
    e) added X-DOMAIN #FIXED attributes for ED:@COMPEN, ED:@IAC
       with their values being taken from the OID table on page 104
       of the Abstract doc
    f) changed the HL7_NAME for ED:@IC, ED:@IAC, CD:@V, CD:@S, CD:@SN,
       CD:@SV, CD:ORIGTEXT, CS:@V, CS:@DN, CV:@V, CV:@DN, CV:@S, SV:@SN,
```

- CV:@SV, CE:@V, CE:@DN, CE:@S, CE:@SN, CE:@SV to match their spelling in the Abstract doc
- g) fixed a bug in the declaration of ED:@IAC, including adding a param entity which enumerates the domain
 - h) corrected the HL7_NAME for ST:@MT, to be "type" instead of "media type"
 - i) removed ENTITY decl for OID, and replaced all references to that OID back to CDATE
 - j) added producer and quality properties to CD, as per the latest revision of the Abstract doc
 - k) remove CR:@V and associated attributes and added a VALUE sub-element, since the value property is of type CD
 - l) assigned this revision number 0.91
- 2000-06-29:
- a) added an HL7-PR to CE which says that the type of any TRANSLN children must be either CV or CS
 - b) changed the attribute type of CE:@T to the enumeration (CS|CV|CE) which defaults to CE, to handle type demotion for CE
 - c) changed the HL7_NAME for CV:@V, CV:@DN, CV:@S, CV:@SN, CV:@SV, CE:@V, CE:@DN, CE:@S, CE:@SN, CE:@SV, II:@V, II:@AAN to match their spelling in the Abstract doc
 - d) changed II:@VT to II:child::VALIDTIME, since the validTime property is of type IVL<TS>, we can't do it as an attribute
 - e) changed II:@TYPE to II:child::TYPE, since the type property is of type CV, we can't do it as an attribute
 - f) removed tel.use.code.set and tel.use.code.default param entities, since they were not actually correctly used in the definition of the TEL type
 - g) changed the #FIXED value of TEL:@USE-DOMAIN to the OID from the table on p 104
 - h) removed ad.purpose.code.set param entity, since it was not actually used in the definition of the AD type
 - i) added AD:@USE-DOMAIN and #FIXED its value to the OID from the table on p. 104
 - j) removed pnxp.qualifier.code.set param entity, since it was not actually used in the definition of the PNXP type
 - k) added PN:@USE-DOMAIN and #FIXED its value to the OID from the table on p. 104
 - l) changed the required-repeating group in AD and PN content models to optional-repeating, to allow for NULL-able message elements
 - m) assigned this revision number 0.92
- 2000-06-30:
- a) Added ON:@TYPE-DOMAIN, with #FIXED value from the OID table on p 104
 - b) assigned this revision number 0.921
- 2000-07-05:
- a) added ability to specify originalText property of CD as either a child::ORIGTXT or as @ORIGTXT.
 - b) assigned this revision number 0.922
- 2000-07-19:
- a) Removed CD:@Q since quality appears to have disappeared from the final abstract ballot doc
 - b) added processing rule to CV which says that if child::ORIGTXT is present, its @MT must have the value "text/plain"
 - c) added @ORIGTXT to appropriate subtypes of CD
 - d) added AD:@VT
 - e) re-added null.code.set param entity decl, which somehow got removed
 - f) changed QTY:@V-T, INT:@V-T and REAL:@V-T to ST
 - g) added note to INT/REAL about PINF/NINF as null flavors
 - h) added note to REAL about precision and XML Schema
 - i) changed processor rule for RTO to require a non-zero DENOM (if DENOM is not null)
 - j) consolidated the processing rules for PQ into a single rule
 - k) consolidated the processing rules for MO into a single rule
 - l) added MO:@U-DOMAIN and #FIXED its value to the OID from the table in appendix A
 - m) added note on IVL<T> that center property is not represented, but can be calculated
 - n) added preliminary xpath processing rules to IVL<T> (they may not be correct)
 - o) Added @V to all IVL<QTY> to support demotion. Also added note that promotion of a QTY to IVL<QTY> is also not supported.
 - p) added note on all IVL<T> asking whether we want to get rid of the structured rep and go to a simple literal form?
 - q) fixed typo in attlist decl for the TYPE element and %ED-attrib.list;
 - r) changed name of HXIT_AD to HXIT_AD.VALUE and added child::VALIDTIME
 - s) added a processing rule to AD which says that either it is null or at least one address part is given
 - t) added optional child::CONFID to all T content models, so that all T are really UVN<T>
 - u) added @PROB to all T, so that all T are really UVP<T>...question: do we need a processing rule which says that we really have either

```

    a UVN<T> or a UVP<T> but not both?
v) added TS:@CAL-DOMAIN and #FIXED its value to the OID from the table in
   appendix A
w) added note that promotion of a QTY to PPD<QTY> is not supported.
   added @CAL to support newly added demotion to PPD_TS.
x) added QTY:@CAL to support TS as an extension of QTY
y) added ED:@MT-DOMAIN and #FIXED its value to the OID from the table in
   appendix A
z) expanded all param entities used as enumerated attribute type domains
   if the param entity was only used once...removed param entity decl
   as well. Only %null.code.set; and %boolean.code.set; remain.
A) assigned this revision number 0.93
2000-07-20:
  a) added a processing rule to ED which says that when demoted to ST,
     no compression algorithm can be specified and no references are
     allowed.
  b) assigned this revision number 0.931
2000-07-21:
  a) removed CD&CR:@P, they were added mistakenly (they're from an
     earlier draft of the abstract ballot)
  b) per GS's comments, redefined CR as essentially an extension of
     CD which adds child::NAME and @INV
  c) assigned this revision number 0.932
2000-07-23:
  a) made ED into a mixed content model, with "value" in PCDATA; added
     ED:@ENC (B64|TXT) [TXT is default]
  b) removed ED:@ICA
  c) removed structured form of IVL_TS, now uses literal form from 7.3.4
     of ballot doc
  d) removed II:child::VALIDTIME and replaced it with II:@VT
  e) changed II:@V to II:@EX

-->
<!--
.....Code Set null.code.set.....
NI      no information
NA      not applicable
UNK     unknown
NASK    not asked
ASKU    asked but unknown
NAV     not available
OTH     other
PINF    positive infinity
NINF    negative infinity

Note 1: NP (not present) does not appear in this list, because it
should never be sent in an instance. The ITS layer should give
the application layer a null flavor of NP for all XML elements/attributes
not present in the instance

Note 2: too bad we can't use schema's NULL facilities...but since
they don't allow nulls on attributes AND they don't allow
flavors of null, we're stuck defining this ourselves.
-->
<xsd:simpleType name="null.code.set">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="NI"/>
    <xsd:enumeration value="NA"/>
    <xsd:enumeration value="UNK"/>
    <xsd:enumeration value="NASK"/>
    <xsd:enumeration value="ASKU"/>
    <xsd:enumeration value="NAV"/>
    <xsd:enumeration value="OTH"/>
    <xsd:enumeration value="PINF"/>
    <xsd:enumeration value="NINF"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
===== Boolean (BL) =====
HL7 processing rules
  @NULL or @V and not(@NULL and @V)

Examples
  <someBL V="true"/>
  <someOtherBL V="false"/>

.....Code Set boolean.code.set.....
true      TRUE
false     FALSE
=====

```

```

-->
<xsd:simpleType name="boolean.code.set">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="true"/>
    <xsd:enumeration value="false"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:group name="BL-cont.model">
  <xsd:sequence>
    <xsd:element ref="NOTE" minOccurs="0"/>
    <xsd:element ref="CONFID" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
<xsd:attributeGroup name="BL-attrib.list">
  <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="BL"/>
  <xsd:attribute name="NULL" type="null.code.set"/>
  <xsd:attribute name="V" type="boolean.code.set"/>
  <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="BL"/>
  <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="value"/>
  <xsd:attribute name="VT" type="xsd:string"/>
  <xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="IVL TS"/>
  <xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
  <xsd:attribute name="PROB" type="xsd:string"/>
  <xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
  <xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
</xsd:attributeGroup>
<!--
===== Binary Data (BIN) =====
The XML ITS does not need to define this datatype, since the
only use of it is within the ED datatype, and a complete definition
of the datatype is not needed for that purpose
=====
-->
<!--
===== Encapsulated Data (ED) =====

HL7 Processing Rules
@V or child::REF
@NULL or @MT
THUMBNAIL[not(child::THUMBNAIL) or child::THUMBNAIL/@NULL]
@T='BIN' and string-length(@V)>0
not(@IAC) or @IA
@PROB >= 0 and @PROB <= 1
@T='ST' and (not(@COMPN) or not(child::REF) or not(child::THUMBNAIL))

Examples:
<someED>cellulitis of the left foot</someED>
<someED MT="image/png"
  IC="aA5mb7c8TXtu392KMsSa2MKkAwL5LKAo2d99azAs3MdUdw">
  <REF V="http://radiology.iuic.edu/xrays/128s8d9ej229se32s.jpg"
    VT="200007200845-0820845" />
  <THUMBNAIL MT="image/jpeg" ENC="B64">
    MNYD83jmMdomSJUEdmde9j44zmMir6edjzMMIjdMDSsWdIJdksIJR3373jeu83
    6edjzMMIjdMDSsWdIJdksIJR3373jeu83MNYD83jmMdomSJUEdmde9j44zmMir
    ...
    omSJUEdmde9j44zmMiromSJUEdmde9j44zmMirdMDSsWdIJdksIJR3373jeu83
    4zmMir6edjzMMIjdMDSsWdIJdksIJR3373jeu83==
  </THUMBNAIL>
</someED>
<someED MT="application/msword" ENC="B64" COMPN="GZ">
  omSJUEdmde9j44zmMiromSJUEdmde9j44zmMirdMDSsWdIJdksIJR3373jeu83
  6edjzMMIjdMDSsWdIJdksIJR3373jeu83MNYD83jmMdomSJUEdmde9j44zmMir
  ...
  MNYD83jmMdomSJUEdmde9j44zmMir6edjzMMIjdMDSsWdIJdksIJR3373jeu83
  4zmMir6edjzMMIjdMDSsWdIJdksIJR3373jeu83==
</someED>

If present, the value of the IA attribute shall be base64 encoded

See note below on the declaration of THUMBNAIL, regarding inherited
values for ED properties

if @IC is valued, the we know that integrityCheckAlgorithm is "SHA-1"

value is text() node ONLY if that text() node is the first child
only 1st occurrence of REF, THUMBNAIL, NOTE and CONFID are significant

@ENC does not correspond to any property

use of xml:lang for the language property (as in the DTD)
deprecated in XML Schema

```

```

=====
-->
<xsd:complexType name="ED-cont.model" mixed="true">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element ref="REF"/>
    <xsd:element ref="THUMBNAIL"/>
    <xsd:element ref="NOTE"/>
    <xsd:element ref="CONFID"/>
  </xsd:choice>
  <xsd:attributeGroup ref="ED-attrib.list"/>
</xsd:complexType>
<xsd:attributeGroup name="ED-attrib.list">
  <xsd:attribute name="T" default="ED">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="ST"/>
        <xsd:enumeration value="ED"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
  <xsd:attribute name="NULL" type="null.code.set"/>
  <xsd:attribute name="ENC" default="TXT">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="B64"/>
        <xsd:enumeration value="TXT"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
  <xsd:attribute name="MT" type="xsd:string" default="text/plain"/>
  <xsd:attribute name="MT-T" type="xsd:NMTOKEN" fixed="CS"/>
  <xsd:attribute name="MT-DOMAIN" type="xsd:NMTOKEN" fixed="2.16.840.1.113883.6.10"/>
  <xsd:attribute name="MT-HL7_NAME" type="xsd:string" fixed="type"/>
  <xsd:attribute name="COMPN">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="DF"/>
        <xsd:enumeration value="GZ"/>
        <xsd:enumeration value="ZL"/>
        <xsd:enumeration value="Z"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
  <xsd:attribute name="COMPN-T" type="xsd:NMTOKEN" fixed="CS"/>
  <xsd:attribute name="COMPN-HL7_NAME" type="xsd:string" fixed="compression"/>
  <xsd:attribute name="COMPN-DOMAIN" type="xsd:NMTOKEN" fixed="2.16.840.1.113883.5.1009"/>
  <xsd:attribute name="IC" type="xsd:string"/>
  <xsd:attribute name="IC-T" type="xsd:NMTOKEN" fixed="BIN"/>
  <xsd:attribute name="IC-HL7_NAME" type="xsd:string" fixed="integrityCheck"/>
  <xsd:attribute name="VT" type="xsd:string"/>
  <xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="IVL TS"/>
  <xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
  <xsd:attribute name="PROB" type="xsd:string"/>
  <xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
  <xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
  <!-- deprecated
  <xsd:attribute ref="xml:lang" use="optional"/>
  <xsd:attribute ref="xml:lang-T" use="optional"/>
  <xsd:attribute ref="xml:lang-HL7_NAME" use="optional"/>
  -->
</xsd:attributeGroup>
<!--
===== String (ST) =====

HL7 Processing Rules
  @NULL or length(@V)>0
  @PROB >= 0 and @PROB <= 1

Examples:
  <someST V="required ST"/>

The value of the charset property should be
obtained from the encoding psuedo attribute of the XML declaration
...if the XML declaration or the encoding psuedo attribute is not
present in the instance, then the CHARSET is assumed to be UTF-8
(as per the XML 1.0 Rec)

Declares the xml namespace as a fixed attribute, which is in scope
for this element and its children

```

```

use of xml:lang for the language property (as in the DTD)
deprecated in XML Schema
=====
-->
<xsd:complexType name="ST-cont.model" mixed="true">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element ref="NOTE"/>
    <xsd:element ref="CONFID"/>
  </xsd:choice>
  <xsd:attributeGroup ref="ST-attrib.list"/>
</xsd:complexType>
<xsd:attributeGroup name="ST-attrib.list">
  <xsd:attribute name="T" type="xsd:string" fixed="ST"/>
  <xsd:attribute name="NULL" type="null.code.set"/>
  <xsd:attribute name="MT" type="xsd:string" fixed="text/plain"/>
  <xsd:attribute name="MT-T" type="xsd:NMTOKEN" fixed="CS"/>
  <xsd:attribute name="MT-HL7_NAME" type="xsd:string" fixed="type"/>
  <xsd:attribute name="VT" type="xsd:string"/>
  <xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="IVL_TS"/>
  <xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
  <xsd:attribute name="PROB" type="xsd:string"/>
  <xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
  <xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
  <!-- deprecated
  <xsd:attribute ref="xml:lang" use="optional"/>
  <xsd:attribute ref="xml:lang-T" use="optional"/>
  <xsd:attribute ref="xml:lang-HL7_NAME" use="optional"/>
  -->
</xsd:attributeGroup>
<!--
===== Concept Descriptor (CD) =====

HL7 Processing Rules
  @NULL or @V
  not(@S) or @V
  not(@V) or @DN
  not(@ORIGTXT) or not(child::ORIGTXT)
  @PROB >= 0 and @PROB <= 1

Examples:
  <someCD T="CD" V="10.3" S="ICD" SV="99" DN="The meaning of the code"/>

need more examples!!!!

Still need processing rules for type demotions

rather than have a "code" attribute, simply uses the V attribute
(but V-HL7_NAME is still "code")

When S and SV appear in a containing element, they are the default
coding system and version for subordinate codes.

You can have EITHER an ORIGTXT child element OR an ORIGTXT attribute,
but not both. If @ORIGTXT is present, then its value is assumed to
be the ID of some other element in the current message/document, in
which case the value of the originalText property is the PCDATA content
of that element. If @ORIGTXT is present, but doesn't resolve to
the ID of some element or the element it resolves to has no PCDATA content,
then the value of the originalText property is NULL with the
default flavor. Note: with this mechanism, @ORIGTXT can only
point to originalText that, essentially, has a media type of
"text/plain".
=====
-->
<xsd:group name="CD-cont.model">
  <xsd:sequence>
    <xsd:element ref="TRANSLTN" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="ORIGTXT" minOccurs="0"/>
    <xsd:element ref="MODIFIER" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="NOTE" minOccurs="0"/>
    <xsd:element ref="CONFID" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
<xsd:attributeGroup name="CD-attrib.list">
  <xsd:attribute name="T" default="CD">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="CD"/>
        <xsd:enumeration value="CE"/>
        <xsd:enumeration value="CV"/>
        <xsd:enumeration value="CS"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>

```

```

        </xsd:restriction>
    </xsd:simpleType>
</xsd:attribute>
<xsd:attribute name="NULL" type="null.code.set"/>
<xsd:attribute name="V" type="xsd:string"/>
<xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="ST"/>
<xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="code"/>
<xsd:attribute name="DN" type="xsd:string"/>
<xsd:attribute name="DN-T" type="xsd:NMTOKEN" fixed="ST"/>
<xsd:attribute name="DN-HL7_NAME" type="xsd:string" fixed="displayName"/>
<xsd:attribute name="S" type="xsd:string"/>
<xsd:attribute name="S-T" type="xsd:NMTOKEN" fixed="OID"/>
<xsd:attribute name="S-HL7_NAME" type="xsd:string" fixed="codeSystem"/>
<xsd:attribute name="SN" type="xsd:string"/>
<xsd:attribute name="SN-T" type="xsd:NMTOKEN" fixed="ST"/>
<xsd:attribute name="SN-HL7_NAME" type="xsd:string" fixed="codeSystemName"/>
<xsd:attribute name="SV" type="xsd:string"/>
<xsd:attribute name="SV-T" type="xsd:NMTOKEN" fixed="ST"/>
<xsd:attribute name="SV-HL7_NAME" type="xsd:string" fixed="codeSystemVersion"/>
<xsd:attribute name="ORIGTXT" type="xsd:IDREF"/>
<xsd:attribute name="ORIGTXT-T" type="xsd:NMTOKEN" fixed="ST"/>
<xsd:attribute name="ORIGTXT-HL7_NAME" type="xsd:string" fixed="originalText"/>
<xsd:attribute name="VT" type="xsd:string"/>
<xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="IVL TS"/>
<xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
<xsd:attribute name="PROB" type="xsd:string"/>
<xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
<xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
</xsd:attributeGroup>
<!--
===== Concept Role (CR) =====

HL7 Processing Rules
  @NULL or child::VALUE
  @PROB >= 0 and @PROB <= 1

Examples: needed!!

as suggested in the abstract doc, the default value for the
inverted property is "false"

CR is modeled essentially as an extension of CD which
adds the name and inverse properties (but the ITS restricts
the use of @S, @SN and @SV)
=====
-->
<xsd:group name="CR-cont.model">
  <xsd:sequence>
    <xsd:element ref="NAME" minOccurs="0"/>
    <xsd:group ref="CD-cont.model"/>
  </xsd:sequence>
</xsd:group>
<!--
Unfortunately, we can't use the CD-attrib.list param entity
for CR's attribute list for a number of reasons: 1) the @T's don't
match; 2) CD contains @PROB which CR should have; 3) we need to
supress @S, @SN and @SV. So, we'll just dup here what we need.
-->
<xsd:attributeGroup name="CR-attrib.list">
  <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="CR"/>
  <xsd:attribute name="NULL" type="null.code.set"/>
  <xsd:attribute name="V" type="xsd:string"/>
  <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="ST"/>
  <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="code"/>
  <xsd:attribute name="DN" type="xsd:string"/>
  <xsd:attribute name="DN-T" type="xsd:NMTOKEN" fixed="ST"/>
  <xsd:attribute name="DN-HL7_NAME" type="xsd:string" fixed="displayName"/>
  <xsd:attribute name="ORIGTXT" type="xsd:IDREF"/>
  <xsd:attribute name="ORIGTXT-T" type="xsd:NMTOKEN" fixed="ST"/>
  <xsd:attribute name="ORIGTXT-HL7_NAME" type="xsd:string" fixed="originalText"/>
  <xsd:attribute name="INV" type="boolean.code.set" default="false"/>
  <xsd:attribute name="INV-T" type="xsd:NMTOKEN" fixed="BL"/>
  <xsd:attribute name="INV-HL7_NAME" type="xsd:string" fixed="inverted"/>
  <xsd:attribute name="VT" type="xsd:string"/>
  <xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="IVL TS"/>
  <xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
  <xsd:attribute name="PROB" type="xsd:string"/>
  <xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
  <xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
</xsd:attributeGroup>
<!--

```



```

===== Coded Simple Value (CS) =====

HL7 Processing Rules
  @NULL or @V
  @PROB >= 0 and @PROB <= 1

Examples: needed!!

rather than have a "code" attribute, simply uses the V attribute

Question: where does the application get the CONTEXT, to
supply the values for S, SV and SN (especially when the instance
of CS is the value of an attribute for some other element)...
one option is illustrated in the declaration of TEL (see USE-DOMAIN)

typo in the abstract doc...duplicate prose describing CNE status of CS
=====
-->
<xsd:group name="CS-cont.model">
  <xsd:sequence>
    <xsd:element ref="NOTE" minOccurs="0"/>
    <xsd:element ref="CONFID" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
<xsd:attributeGroup name="CS-attrib.list">
  <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="CS"/>
  <xsd:attribute name="NULL" type="null.code.set"/>
  <xsd:attribute name="V" type="xsd:string"/>
  <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="ST"/>
  <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="code"/>
  <xsd:attribute name="DN" type="xsd:string"/>
  <xsd:attribute name="DN-T" type="xsd:NMTOKEN" fixed="ST"/>
  <xsd:attribute name="DN-HL7_NAME" type="xsd:string" fixed="displayName"/>
  <xsd:attribute name="VT" type="xsd:string"/>
  <xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="IVL_TS"/>
  <xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
  <xsd:attribute name="PROB" type="xsd:string"/>
  <xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
  <xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
</xsd:attributeGroup>
<!--
===== Coded Value (CV) =====

HL7 Processing Rules
  @NULL or @V
  not(@S) or @V
  child::ORIGTXT[@MT='text/plain']
  @PROB >= 0 and @PROB <= 1

Examples: needed!!

Still need processing rules for type demotion

rather than have a "code" attribute, simply uses the V attribute

When S and SV appear in a containing element, they are the default coding
system and version for subordinate codes.

You can have EITHER an ORIGTXT child element OR an ORIGTXT attribute,
but not both. If @ORIGTXT is present, then its value is assumed to
be the ID of some other element in the current message/document, in
which case the value of the originalText property is the PCDATA content
of that element. If @ORIGTXT is present, but doesn't resolve to
the ID of some element or the element it resolves to has no PCDATA content,
then the value of the originalText property is NULL with the
default flavor. Note: with this mechanism, @ORIGTXT can only
point to originalText that, essentially, has a media type of
"text/plain".
=====
-->
<xsd:group name="CV-cont.model">
  <xsd:sequence>
    <xsd:element ref="ORIGTXT" minOccurs="0"/>
    <xsd:element ref="NOTE" minOccurs="0"/>
    <xsd:element ref="CONFID" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
<xsd:attributeGroup name="CV-attrib.list">
  <xsd:attribute name="T" default="CV">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">

```

```

        <xsd:enumeration value="CS"/>
        <xsd:enumeration value="CV"/>
    </xsd:restriction>
</xsd:simpleType>
</xsd:attribute>
<xsd:attribute name="NULL" type="null.code.set"/>
<xsd:attribute name="V" type="xsd:string"/>
<xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="ST"/>
<xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="code"/>
<xsd:attribute name="DN" type="xsd:string"/>
<xsd:attribute name="DN-T" type="xsd:NMTOKEN" fixed="ST"/>
<xsd:attribute name="DN-HL7_NAME" type="xsd:string" fixed="displayName"/>
<xsd:attribute name="S" type="xsd:string"/>
<xsd:attribute name="S-T" type="xsd:NMTOKEN" fixed="OID"/>
<xsd:attribute name="S-HL7_NAME" type="xsd:string" fixed="codeSystem"/>
<xsd:attribute name="SN" type="xsd:string"/>
<xsd:attribute name="SN-T" type="xsd:NMTOKEN" fixed="ST"/>
<xsd:attribute name="SN-HL7_NAME" type="xsd:string" fixed="codeSystemName"/>
<xsd:attribute name="SV" type="xsd:string"/>
<xsd:attribute name="SV-T" type="xsd:NMTOKEN" fixed="OID"/>
<xsd:attribute name="SV-HL7_NAME" type="xsd:string" fixed="codeSystemVersion"/>
<xsd:attribute name="ORIGTXT" type="xsd:IDREF"/>
<xsd:attribute name="ORIGTXT-T" type="xsd:NMTOKEN" fixed="ST"/>
<xsd:attribute name="ORIGTXT-HL7_NAME" type="xsd:string" fixed="originalText"/>
<xsd:attribute name="VT" type="xsd:string"/>
<xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="IVL_TS"/>
<xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
<xsd:attribute name="PROB" type="xsd:string"/>
<xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
<xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
</xsd:attributeGroup>
<!--
===== Coded With Equivalents (CE) =====

HL7 Processing Rules
  @NULL or @V
  not(@S) or @V
  child::TRANSTN[@T='CV' or @T='CS']
  @PROB >= 0 and @PROB <= 1

Examples: needed!!

Still need processing rules for type demotions

rather than have a "code" attribute, simply uses the V attribute

When S and SV appear in a containing element, they are the default coding
system and version for subordinate codes.

You can have EITHER an ORIGTXT child element OR an ORIGTXT attribute,
but not both. If @ORIGTXT is present, then its value is assumed to
be the ID of some other element in the current message/document, in
which case the value of the originalText property is the PCDATA content
of that element. If @ORIGTXT is present, but doesn't resolve to
the ID of some element or the element it resolves to has no PCDATA content,
then the value of the originalText property is NULL with the
default flavor. Note: with this mechanism, @ORIGTXT can only
point to originalText that, essentially, has a media type of
"text/plain".
=====
-->
<xsd:group name="CE-cont.model">
  <xsd:sequence>
    <xsd:element ref="ORIGTXT" minOccurs="0"/>
    <xsd:element ref="TRANSLTN" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="NOTE" minOccurs="0"/>
    <xsd:element ref="CONFID" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
<xsd:attributeGroup name="CE-attrib.list">
  <xsd:attribute name="T" default="CE">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="CS"/>
        <xsd:enumeration value="CV"/>
        <xsd:enumeration value="CE"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
  <xsd:attribute name="NULL" type="null.code.set"/>
  <xsd:attribute name="V" type="xsd:string"/>

```

```

<xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="ST"/>
<xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="code"/>
<xsd:attribute name="DN" type="xsd:string"/>
<xsd:attribute name="DN-T" type="xsd:NMTOKEN" fixed="ST"/>
<xsd:attribute name="DN-HL7_NAME" type="xsd:string" fixed="displayName"/>
<xsd:attribute name="S" type="xsd:string"/>
<xsd:attribute name="S-T" type="xsd:NMTOKEN" fixed="OID"/>
<xsd:attribute name="S-HL7_NAME" type="xsd:string" fixed="codeSystem"/>
<xsd:attribute name="SN" type="xsd:string"/>
<xsd:attribute name="SN-T" type="xsd:NMTOKEN" fixed="ST"/>
<xsd:attribute name="SN-HL7_NAME" type="xsd:string" fixed="codeSystemName"/>
<xsd:attribute name="SV" type="xsd:string"/>
<xsd:attribute name="SV-T" type="xsd:NMTOKEN" fixed="OID"/>
<xsd:attribute name="SV-HL7_NAME" type="xsd:string" fixed="codeSystemVersion"/>
<xsd:attribute name="ORIGTXT" type="xsd:IDREF"/>
<xsd:attribute name="ORIGTXT-T" type="xsd:NMTOKEN" fixed="ST"/>
<xsd:attribute name="ORIGTXT-HL7_NAME" type="xsd:string" fixed="originalText"/>
<xsd:attribute name="VT" type="xsd:string"/>
<xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="IVL TS"/>
<xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
<xsd:attribute name="PROB" type="xsd:string"/>
<xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
<xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
</xsd:attributeGroup>
<!--
===== ISO Object Identifier (OID) =====
no need for a separate OID type, since it is never used
by itself, but only as the type of property of some other type,
in which case it is also used just as a CDATA attribute
=====
-->
<!--
===== Instance Identifier (II) =====

HL7 Processing Rules
  @NULL or @RT
  @PROB >= 0 and @PROB <= 1

Examples:
  <someII V="optional ST" RT="required OID" ANN="optional ST"
  VT='2000-06-23-2000-07-24' />
=====
-->
<xsd:group name="II-cont.model">
  <xsd:sequence>
    <xsd:element ref="TYPE" minOccurs="0"/>
    <xsd:element ref="NOTE" minOccurs="0"/>
    <xsd:element ref="CONFID" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
<xsd:attributeGroup name="II-attrib.list">
  <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="II"/>
  <xsd:attribute name="NULL" type="null.code.set"/>
  <xsd:attribute name="EX" type="xsd:string"/>
  <xsd:attribute name="EX-T" type="xsd:NMTOKEN" fixed="ST"/>
  <xsd:attribute name="EX-HL7_NAME" type="xsd:string" fixed="extension"/>
  <xsd:attribute name="RT" type="xsd:string"/>
  <xsd:attribute name="RT-T" type="xsd:NMTOKEN" fixed="OID"/>
  <xsd:attribute name="RT-HL7_NAME" type="xsd:string" fixed="root"/>
  <xsd:attribute name="AAN" type="xsd:string"/>
  <xsd:attribute name="AAN-T" type="xsd:NMTOKEN" fixed="ST"/>
  <xsd:attribute name="AAN-HL7_NAME" type="xsd:string" fixed="assigningAuthorityName"/>
  <xsd:attribute name="VT" type="xsd:string"/>
  <xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="IVL TS"/>
  <xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
  <xsd:attribute name="PROB" type="xsd:string"/>
  <xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
  <xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
</xsd:attributeGroup>
<!--
===== telecommunication address (TEL) =====

HL7 Processing Rules
  @NULL or @V
  @PROB >= 0 and @PROB <= 1

Examples:
  <someTEL V="http://example.com/somePath" USE="WP"/>
  <someTEL V="tel:(358)555-1234" USE="HP EC"/>

```

The latest draft of the Abstract doc mentions a resolvedData property, although it doesn't describe that property in detail. I believe the property is ill-defined and should not be present, hence I haven't represented it here.

```

=====
-->
<xsd:group name="TEL-cont.model">
  <xsd:sequence>
    <xsd:element ref="NOTE" minOccurs="0"/>
    <xsd:element ref="CONFID" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
<xsd:attributeGroup name="TEL-attrib.list">
  <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="TEL"/>
  <xsd:attribute name="NULL" type="null.code.set"/>
  <xsd:attribute name="V" type="xsd:string"/>
  <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="URL"/>
  <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="value"/>
  <xsd:attribute name="USE" type="xsd:NMTOKENS"/>
  <xsd:attribute name="USE-T" type="xsd:NMTOKEN" fixed="SET_CS"/>
  <xsd:attribute name="USE-DOMAIN" type="xsd:string" fixed="2.16.840.1.113883.5.1011"/>
  <xsd:attribute name="USE-HL7_NAME" type="xsd:string" fixed="use"/>
  <xsd:attribute name="VT" type="xsd:string"/>
  <xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="GTS"/>
  <xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
  <xsd:attribute name="PROB" type="xsd:string"/>
  <xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
  <xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
</xsd:attributeGroup>
<!--
===== address part (ADXP) =====
      address part type codes are represented by sub-element GIs in the
      enclosing AD
=====
-->
<xsd:group name="ADXP-cont.model">
  <xsd:sequence>
    <xsd:element ref="NOTE" minOccurs="0"/>
    <xsd:element ref="CONFID" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
<xsd:attributeGroup name="ADXP-attrib.list">
  <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="ADXP"/>
  <xsd:attribute name="NULL" type="null.code.set"/>
  <xsd:attribute name="V" type="xsd:string"/>
  <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="ST"/>
  <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="value"/>
  <xsd:attribute name="VT" type="xsd:string"/>
  <xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="IVL_TS"/>
  <xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
  <xsd:attribute name="PROB" type="xsd:string"/>
  <xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
  <xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
</xsd:attributeGroup>
<!--
===== Postal and Residential Address (AD) =====

HL7 Processing Rules
  @NULL or child::LIT or child::DEL or child::CNT etc.
  @PROB >= 0 and @PROB <= 1

Examples:
  <someAD USE="PST"/>
    <HNR V="970"/>
    <STR V="Post St"/>
    <DIR V="NE"/>
    <CTY V="Alameda"/>
    <STA V="CA"/>
    <ZIP V="94501"/>
  </someAD>

formatted property is represented as @V

LIT as an ADXP type code has been removed from the latest draft of
the Abstract doc, with the notion being that what was formerly the content
of LIT should now become PCDATA...this requires a mixed content model, which
I'm not quite ready to introduce...so, LIT is still used here. The problem
with a mixed content model in this case is that child::NOTE's could be
sprinkled anywhere and the semantics of that aren't defined. Having a
mixed content model would also make it harder (tho not impossible) to
define a good content model for HXIT<AD> (unless of course, we switch

```

```

to a literal for for IVL<TS>, which would simplify the transfer syntax
but complicate the applications)
=====
-->
<xsd:group name="AD-cont.model">
  <xsd:sequence>
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element ref="LIT"/>
      <xsd:element ref="DEL"/>
      <xsd:element ref="CNT"/>
      <xsd:element ref="STA"/>
      <xsd:element ref="CTY"/>
      <xsd:element ref="ZIP"/>
      <xsd:element ref="STR"/>
      <xsd:element ref="HNR"/>
      <xsd:element ref="DIR"/>
      <xsd:element ref="ADL"/>
      <xsd:element ref="POB"/>
    </xsd:choice>
    <xsd:element ref="NOTE" minOccurs="0"/>
    <xsd:element ref="CONFID" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
<xsd:attributeGroup name="AD-attrib.list">
  <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="AD"/>
  <xsd:attribute name="NULL" type="null.code.set"/>
  <xsd:attribute name="V" type="xsd:string"/>
  <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="ST"/>
  <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="formatted"/>
  <xsd:attribute name="USE" type="xsd:NMTOKENS"/>
  <xsd:attribute name="USE-T" type="xsd:NMTOKEN" fixed="SET_CS"/>
  <xsd:attribute name="USE-DOMAIN" type="xsd:string" fixed="2.16.840.1.113883.5.1012"/>
  <xsd:attribute name="USE-HL7_NAME" type="xsd:string" fixed="use"/>
  <xsd:attribute name="VT" type="xsd:string"/>
  <xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="GTS"/>
  <xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
  <xsd:attribute name="PROB" type="xsd:string"/>
  <xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
  <xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
</xsd:attributeGroup>
<!--
===== Person Name Part (PNXP) =====

HL7 Processing Rules
  @NULL or @V

  name part type codes are represented by sub-element GIs in the
  enclosing PN
=====
-->
<xsd:group name="PNXP-cont.model">
  <xsd:sequence>
    <xsd:element ref="NOTE" minOccurs="0"/>
    <xsd:element ref="CONFID" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
<xsd:attributeGroup name="PNXP-attrib.list">
  <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="PNXP"/>
  <xsd:attribute name="NULL" type="null.code.set"/>
  <xsd:attribute name="V" type="xsd:string"/>
  <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="ST"/>
  <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="value"/>
  <xsd:attribute name="QUAL" type="xsd:NMTOKENS"/>
  <xsd:attribute name="QUAL-T" type="xsd:NMTOKEN" fixed="SET_CS"/>
  <xsd:attribute name="QUAL-DOMAIN" type="xsd:string" fixed="2.16.840.1.113883.5.1014"/>
  <xsd:attribute name="QUAL-HL7_NAME" type="xsd:string" fixed="qualifier"/>
  <xsd:attribute name="VT" type="xsd:string"/>
  <xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="IVL_TS"/>
  <xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
  <xsd:attribute name="PROB" type="xsd:string"/>
  <xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
  <xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
</xsd:attributeGroup>
<!--
===== Person Name (PN) =====

HL7 Processing Rules

The order in which the name parts are sent should represent the natural
order for displaying the name. It is not required to break names down
using all of the elements, although individual nations may require

```

specific elements. Displaying a name depends on the white space rules included in the V3DT report. It may be necessary to preserve white space in certain elements.

@PROB >= 0 and @PROB <= 1

Examples:

```
<somePN NOTE="This is an example of 'Wesley Rishel'">
  <FAM V="Rishel" QUAL="BR RE"/>
  <GIV V="Wesley" QUAL="BR RE"/>
</somePN>

<somePN NOTE="This is an example of 'Irma Corine Jongeneel-de Haas'">
  <GIV V="Irma" QUAL="RE"/>
  <GIV V="Corine" QUAL="RE"/>
  <FAM V="Jongeneel" QUAL="RE SP"/>
  <DEL V="-"/>
  <FAM V="de Haas" QUAL="RE BR"/>
</somePN>
```

uses @V to represent the formatted property

```
-----
-->
<xsd:group name="PN-cont.model">
  <xsd:sequence>
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element ref="GIV"/>
      <xsd:element ref="MID"/>
      <xsd:element ref="FAM"/>
      <xsd:element ref="PFX"/>
      <xsd:element ref="SFX"/>
      <xsd:element ref="DEL"/>
    </xsd:choice>
    <xsd:element ref="NOTE" minOccurs="0"/>
    <xsd:element ref="CONFID" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
<xsd:attributeGroup name="PN-attrib.list">
  <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="PN"/>
  <xsd:attribute name="NULL" type="null.code.set"/>
  <xsd:attribute name="V" type="xsd:string"/>
  <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="ST"/>
  <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="formatted"/>
  <xsd:attribute name="VT" type="xsd:string"/>
  <xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="IVL_TS"/>
  <xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
  <xsd:attribute name="PROB" type="xsd:string"/>
  <xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
  <xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
</xsd:attributeGroup>
<!--
```

----- Organization Name (ON) -----

HL7 Processing Rules
 @NULL or @V
 @PROB >= 0 and @PROB <= 1

Examples:

```
<someON TYPE="optional CS" V="optional ST"/>
```

should this be modeled closer to AD and PN?

```
-----
-->
<xsd:group name="ON-cont.model">
  <xsd:sequence>
    <xsd:element ref="NOTE" minOccurs="0"/>
    <xsd:element ref="CONFID" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
<xsd:attributeGroup name="ON-attrib.list">
  <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="ON"/>
  <xsd:attribute name="NULL" type="null.code.set"/>
  <xsd:attribute name="V" type="xsd:string"/>
  <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="ST"/>
  <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="value"/>
  <xsd:attribute name="TYPE">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="L"/>
        <xsd:enumeration value="A"/>
        <xsd:enumeration value="ST"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
</xsd:attributeGroup>
```

```

        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
    <xsd:attribute name="TYPE-T" type="xsd:NMTOKEN" fixed="CS"/>
    <xsd:attribute name="TYPE-DOMAIN" type="xsd:string" fixed="2.16.840.1.113883.5.1015"/>
    <xsd:attribute name="TYPE-HL7_NAME" type="xsd:string" fixed="type"/>
    <xsd:attribute name="VT" type="xsd:string"/>
    <xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="IVL_TS"/>
    <xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
    <xsd:attribute name="PROB" type="xsd:string"/>
    <xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
    <xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
  </xsd:attributeGroup>
<!--
===== Quantity (QTY) =====

HL7 Processing Rules:
  @NULL or @V
  ###not xpath### INT|REAL|TS ::= V or PQ|MO ::= V, U?
  not(@T='PQ' or @T='MO' or @T='TS') or @U ### I think does the HL7-PR above
  @PROB >= 0 and @PROB <= 1

Examples:
  <someQTY T="INT|REAL|PQ|MO" V="123.75" U="mg/dL"/>

Unless @T=TS, @CAL is meaningless and should be ignored

The cardinality/default value for QTY is defined as a param entity
whose starting value is "#IMPLIED". We do it this way so that the
NUM and DENOM properties of RTO can take on the default value
"1" as specified in the abstract ballot.
=====
-->
<xsd:group name="QTY-cont.model">
  <xsd:sequence>
    <xsd:element ref="NOTE" minOccurs="0"/>
    <xsd:element ref="CONFID" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
<xsd:attributeGroup name="QTY-attrib.list">
  <xsd:attribute name="T" default="INT">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="INT"/>
        <xsd:enumeration value="REAL"/>
        <xsd:enumeration value="PQ"/>
        <xsd:enumeration value="MO"/>
        <xsd:enumeration value="TS"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
  <xsd:attribute name="NULL" type="null.code.set"/>
  <xsd:attribute name="V" type="xsd:string"/>
  <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="ST"/>
  <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="value"/>
  <xsd:attribute name="U" type="xsd:string"/>
  <xsd:attribute name="U-T" type="xsd:NMTOKEN" fixed="CS"/>
  <xsd:attribute name="U-HL7_NAME" type="xsd:string" fixed="unit"/>
  <xsd:attribute name="VT" type="xsd:string"/>
  <xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="IVL_TS"/>
  <xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
  <xsd:attribute name="PROB" type="xsd:string"/>
  <xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
  <xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
</xsd:attributeGroup>
<!--
===== Integer (INT) =====

HL7 Processing Rules:
  @NULL or @V
  @PROB >= 0 and @PROB <= 1

Examples:
  <someINT V="required ST"/>

The "exceptional" values positive and negative infinity are
represented as null flavors
=====
-->
<xsd:group name="INT-cont.model">
  <xsd:sequence>

```

```

        <xsd:element ref="NOTE" minOccurs="0"/>
        <xsd:element ref="CONFID" minOccurs="0"/>
    </xsd:sequence>
</xsd:group>
<xsd:attributeGroup name="INT-attrib.list">
    <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="INT"/>
    <xsd:attribute name="NULL" type="null.code.set"/>
    <xsd:attribute name="V" type="xsd:string"/>
    <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="ST"/>
    <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="value"/>
    <xsd:attribute name="VT" type="xsd:string"/>
    <xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="IVL_TS"/>
    <xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
    <xsd:attribute name="PROB" type="xsd:string"/>
    <xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
    <xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
</xsd:attributeGroup>
<!--
===== Real Number (REAL) =====

HL7 Processing Rules:
    @NULL or @V
    @PROB >= 0 and @PROB <= 1

Examples:
    <someREAL V="required REAL"/>

The "exceptional" values positive and negative infinity are
represented as null flavors

Note: XML Schema currently allows trailing zeros in its literals
of type decimal, hence we will be able to get the precision property
from the literals. However, there is discussion within the WG
to disallow trailing zeros, and if this happens, we will have to
add @PREC or some such
=====
-->
<xsd:group name="REAL-cont.model">
    <xsd:sequence>
        <xsd:element ref="NOTE" minOccurs="0"/>
        <xsd:element ref="CONFID" minOccurs="0"/>
    </xsd:sequence>
</xsd:group>
<xsd:attributeGroup name="REAL-attrib.list">
    <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="REAL"/>
    <xsd:attribute name="NULL" type="null.code.set"/>
    <xsd:attribute name="V" type="xsd:string"/>
    <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="ST"/>
    <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="value"/>
    <xsd:attribute name="VT" type="xsd:string"/>
    <xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="IVL_TS"/>
    <xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
    <xsd:attribute name="PROB" type="xsd:string"/>
    <xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
    <xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
</xsd:attributeGroup>
<!--
===== Ratio of Quantities (RTO) =====

HL7 Processing Rules:
    @NULL or (child::NUM and not(child::NUM/@NULL))
    @NULL or (child::DENOM[@V!='0'] and not(child::DENOM/@NULL))
    @PROB >= 0 and @PROB <= 1

Examples:
    <someRTO>
        <NUM T="INT|REAL|PQ|MO" V="123.75" U="mg/dL"/>
        <DENOM T="INT|REAL|PQ|MO" V="123.75" U="mg/dL"/>
    </someRTO>

How should the demotion to PQ and REAL be handled?
=====
-->
<xsd:group name="RTO-cont.model">
    <xsd:sequence>
        <xsd:sequence minOccurs="0">
            <xsd:element ref="NUM"/>
            <xsd:element ref="DENOM"/>
        </xsd:sequence>
        <xsd:element ref="NOTE" minOccurs="0"/>
        <xsd:element ref="CONFID" minOccurs="0"/>
    </xsd:sequence>

```



```

    </xsd:sequence>
  </xsd:group>
  <xsd:attributeGroup name="RTO-attrib.list">
    <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="RTO"/>
    <xsd:attribute name="NULL" type="null.code.set"/>
    <xsd:attribute name="VT" type="xsd:string"/>
    <xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="IVL_TS"/>
    <xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
    <xsd:attribute name="PROB" type="xsd:string"/>
    <xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
    <xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
  </xsd:attributeGroup>
  <!--
===== Physical Quantity (PQ) =====

  HL7 Processing Rules:
    @NULL or (@V and @U)
    @PROB >= 0 and @PROB <= 1

  Examples:
    <somePQ V="1123.37" U="cm"/>

  Does the XML ITS have to say *anything* about about the
  canonical form for a PQ?

  -->
  <xsd:group name="PQ-cont.model">
    <xsd:sequence>
      <xsd:element ref="NOTE" minOccurs="0"/>
      <xsd:element ref="CONFID" minOccurs="0"/>
    </xsd:sequence>
  </xsd:group>
  <xsd:attributeGroup name="PQ-attrib.list">
    <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="PQ"/>
    <xsd:attribute name="NULL" type="null.code.set"/>
    <xsd:attribute name="V" type="xsd:string"/>
    <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="REAL"/>
    <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="value"/>
    <xsd:attribute name="U" type="xsd:string"/>
    <xsd:attribute name="U-T" type="xsd:NMTOKEN" fixed="CS"/>
    <xsd:attribute name="U-HL7_NAME" type="xsd:string" fixed="unit"/>
    <xsd:attribute name="VT" type="xsd:string"/>
    <xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="IVL_TS"/>
    <xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
    <xsd:attribute name="PROB" type="xsd:string"/>
    <xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
    <xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
  </xsd:attributeGroup>
  <!--
===== Monetary Amount (MO) =====

  HL7 Processing Rules:
    @NULL or (@V and @U)
    @PROB >= 0 and @PROB <= 1

  Examples:
    <someMO V="1123.37" U="USD"/>

  Why is there an MO type to begin with...why can't this just be a
  restriction of PQ, where the domain of unit is fixed to ISO 4217?

  -->
  <xsd:group name="MO-cont.model">
    <xsd:sequence>
      <xsd:element ref="NOTE" minOccurs="0"/>
      <xsd:element ref="CONFID" minOccurs="0"/>
    </xsd:sequence>
  </xsd:group>
  <xsd:attributeGroup name="MO-attrib.list">
    <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="MO"/>
    <xsd:attribute name="NULL" type="null.code.set"/>
    <xsd:attribute name="V" type="xsd:string"/>
    <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="REAL"/>
    <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="value"/>
    <xsd:attribute name="U" type="xsd:string"/>
    <xsd:attribute name="U-T" type="xsd:NMTOKEN" fixed="CS"/>
    <xsd:attribute name="U-DOMAIN" type="xsd:string" fixed="2.16.840.1.113883.6.9"/>
    <xsd:attribute name="U-HL7_NAME" type="xsd:string" fixed="currency"/>
    <xsd:attribute name="VT" type="xsd:string"/>
    <xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="IVL_TS"/>

```

```

    <xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
    <xsd:attribute name="PROB" type="xsd:string"/>
    <xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
    <xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
  </xsd:attributeGroup>
<!--
===== Point in Time (TS) =====

HL7 Processing Rules:
  @NULL or @V
  @PROB >= 0 and @PROB <= 1

Examples:
  <someTS V="19990924162403-0800"/>

the timezone property is conveyed as part of value (V)
instead of as a separate property (there is some call w/i the
XML Schema WG that ALL time-related datatypes in the schema spec
should be specified in UTC, I'm fighting that but may loose)

offset is not represented at all (and to editorialize, I believe
should be removed from the abstract type, unless the epoch is also
added...semantically, it does no good to know that some point in
time is 1 day from an epoch if I don't also know what the epoch is).
=====
-->
<xsd:group name="TS-cont.model">
  <xsd:sequence>
    <xsd:element ref="NOTE" minOccurs="0"/>
    <xsd:element ref="CONFID" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
<xsd:attributeGroup name="TS-attrib.list">
  <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="TS"/>
  <xsd:attribute name="NULL" type="null.code.set"/>
  <xsd:attribute name="V" type="xsd:string"/>
  <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="ST"/>
  <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="value"/>
  <xsd:attribute name="VT" type="xsd:string"/>
  <xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="IVL TS"/>
  <xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
  <xsd:attribute name="PROB" type="xsd:string"/>
  <xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
  <xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
</xsd:attributeGroup>
<!--
-->
<!-- We do not need SET<T> for all T -->
<!-- We do not need LIST<T> for all T -->
<!-- We do not need BAG<T> for all T -->
<!--
-->
<!--
===== interval of physical quantities (IVL_PQ) =====

To avoid redundancy and difficulty in processing we factor the
units from both bounds into one unit of the interval. This
requires both low and high bound to have a common unit. Width,
however must have its own unit, since in difference-scale
quantities the width may be of a different unit.

HL7 Processing Rules:
  The following combinations of components are valid:

  @NULL or @V
  @PROB >= 0 and @PROB <= 1

Examples:
  <someIVL_PQ LOW="optional REAL" LOW_CLOSED="optional BL" HIGH="optional REAL"
    HIGH_CLOSED="optional BL" UNIT="optional CV" WID="optional PQ"/>

The XML ITS does not support promotion of a PQ into an IVL_PQ,
although demotion of IVL_PQ to PQ is supported (using @V and @U)
=====
-->
<xsd:group name="IVL_PQ-cont.model">
  <xsd:sequence>
    <xsd:element ref="NOTE" minOccurs="0"/>
    <xsd:element ref="CONFID" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>

```

```

<xsd:attributeGroup name="IVL_PQ-attrib.list">
  <xsd:attribute name="T" default="IVL_PQ">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="IVL_PQ"/>
        <xsd:enumeration value="PQ"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
  <xsd:attribute name="NULL" type="null.code.set"/>
  <xsd:attribute name="V" type="xsd:string"/>
  <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="REAL"/>
  <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="value"/>
  <xsd:attribute name="U" type="xsd:string"/>
  <xsd:attribute name="U-T" type="xsd:NMTOKEN" fixed="CS"/>
  <xsd:attribute name="U-HL7_NAME" type="xsd:string" fixed="unit"/>
  <xsd:attribute name="VT" type="xsd:string"/>
  <xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="IVL_TS"/>
  <xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
  <xsd:attribute name="PROB" type="xsd:string"/>
  <xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
  <xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
</xsd:attributeGroup>
<!--
===== interval of points in time (IVL_TS) =====

  Note that DIF_TS is just an alias for PQ with a dimensional
  constraint on the unit (PQ in the dimension of time.)

  The XML ITS does not support promotion of a PQ into an IVL_PQ,
  although demotion of IVL_PQ to PQ is supported (using @V and @U)

  literals (e.g., values of @V) follow literal form from section
  7.4.3 of abstract ballot
=====
-->
<xsd:group name="IVL_TS-cont.model">
  <xsd:sequence>
    <xsd:element ref="NOTE" minOccurs="0"/>
    <xsd:element ref="CONFID" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
<xsd:attributeGroup name="IVL_TS-attrib.list">
  <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="IVL_TS"/>
  <xsd:attribute name="NULL" type="null.code.set"/>
  <xsd:attribute name="V" type="xsd:string"/>
  <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="TS"/>
  <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="value"/>
  <xsd:attribute name="VT" type="xsd:string"/>
  <xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="IVL_TS"/>
  <xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
  <xsd:attribute name="PROB" type="xsd:string"/>
  <xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
  <xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
</xsd:attributeGroup>
<!--
===== Periodic Interval of Time (PIVL) =====
  no need for a separate PIVL type, since it is never used
  by itself, but only to define the semantics of GTS
=====
-->
<!--
===== Event-Related Periodic Interval of Time (EIVL) =====
  no need for a separate EIVL type, since it is never used
  by itself, but only to define the semantics of GTS
=====
-->
<!--
===== General Timing Specification (GTS) =====

  HL7 Processing Rules:

    The following combinations of components are valid:

    @NULL or @V
    @PROB >= 0 and @PROB <= 1

  HL7 Processing Rules:

    The values of @V follow the Literal Form in section 8.3.1.3 of the
    abstract ballot

```

```

-->
<xsd:group name="GTS-cont.model">
  <xsd:sequence>
    <xsd:element ref="NOTE" minOccurs="0"/>
    <xsd:element ref="CONFID" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
<xsd:attributeGroup name="GTS-attrib.list">
  <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="GTS"/>
  <xsd:attribute name="NULL" type="null.code.set"/>
  <xsd:attribute name="V" type="xsd:string"/>
  <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="GTS"/>
  <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="value"/>
  <xsd:attribute name="VT" type="xsd:string"/>
  <xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="IVL_TS"/>
  <xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
  <xsd:attribute name="PROB" type="xsd:string"/>
  <xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
  <xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
</xsd:attributeGroup>
<!--
-->
<!--
===== ANT<T> =====
no need for ANT<T> since all types defined for the XML ITS
already include an optional note property
=====
-->
<!--
===== HIST<T> =====
no need for HIST<T> since all types defined for the XML ITS
already include an optional validTime property (and hence,
are HXIT<T>), and HIST<T> is simply an optionally repeating
HXIT<T>
=====
-->
<!--
===== Uncertain Value-Narrative (UVN<T>) =====
no need for UVN<T> since all types defined for the XML ITS
already include an optional confidence property
=====
-->
<!--
===== Uncertain Value-Probabilistic (UVP<T>) =====
no need for UVP<T> since all types defined for the XML ITS
already include an optional probability property
=====
-->
<!--
===== Non-Parametric Probability Distribution (NPPD<T>) =====
no need for NPPD<T> since all types defined for the XML ITS
already include an optional probability property (and hence,
are already UVP<T>) and NPPD<T> is simply an optionally
repeating UVP<T>
=====
-->
<!--
-->
<!--
===== PPD<QTY> =====

HL7 Processing Rules:
  @NULL or @V

Examples: needed!!!

The XML ITS does not support promoting a QTY to a PPD<QTY>
=====
-->
<xsd:group name="PPD_QTY-cont.model">
  <xsd:sequence>
    <xsd:element ref="NOTE" minOccurs="0"/>
    <xsd:element ref="CONFID" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
<xsd:attributeGroup name="PPD_QTY-attrib.list">
  <xsd:attribute name="T" default="PPD_INT">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="PPD_INT"/>

```

```

        <xsd:enumeration value="PPD_REAL"/>
        <xsd:enumeration value="PPD_PQ"/>
        <xsd:enumeration value="PPD_MO"/>
        <xsd:enumeration value="PPD_TS"/>
    </xsd:restriction>
</xsd:simpleType>
</xsd:attribute>
<xsd:attribute name="NULL" type="null.code.set"/>
<xsd:attribute name="V" type="xsd:string"/>
<xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="REAL"/>
<xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="value"/>
<xsd:attribute name="U" type="xsd:string"/>
<xsd:attribute name="U-T" type="xsd:NMTOKEN" fixed="ST"/>
<xsd:attribute name="U-HL7_NAME" type="xsd:string" fixed="unit"/>
<xsd:attribute name="SD" type="xsd:string"/>
<xsd:attribute name="SD-T" type="xsd:NMTOKEN" fixed="REAL"/>
<xsd:attribute name="SD-HL7_NAME" type="xsd:string" fixed="standardDeviation"/>
<xsd:attribute name="SDU" type="xsd:string"/>
<xsd:attribute name="SDU-T" type="xsd:NMTOKEN" fixed="CS"/>
<xsd:attribute name="SDU-HL7_NAME" type="xsd:string" fixed="standardDeviation unit"/>
<xsd:attribute name="TY">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="U"/>
            <xsd:enumeration value="N"/>
            <xsd:enumeration value="LN"/>
            <xsd:enumeration value="G"/>
            <xsd:enumeration value="E"/>
            <xsd:enumeration value="X2"/>
            <xsd:enumeration value="T"/>
            <xsd:enumeration value="F"/>
            <xsd:enumeration value="B"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:attribute>
<xsd:attribute name="TY-T" type="xsd:NMTOKEN" fixed="CS"/>
<xsd:attribute name="TY-DOMAIN" type="xsd:string" fixed="2.16.840.1.113883.5.1019"/>
<xsd:attribute name="TY-HL7_NAME" type="xsd:string" fixed="type"/>
<xsd:attribute name="VT" type="xsd:string"/>
<xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="IVL TS"/>
<xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
<xsd:attribute name="PROB" type="xsd:string"/>
<xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
<xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
</xsd:attributeGroup>
<!--
===== PPD<REAL> =====

HL7 Processing Rules:
    @NULL or @V

Examples: needed!!!

The abstract ballot says that @TY is of type CV...shouldn't
it really be of type CS since the domain is fixed and no
exceptions are allowed? Assuming so, none of extra properties
of CV are represented

The XML ITS does not support promoting a REAL to a PPD<REAL>
=====
-->
<xsd:group name="PPD_REAL-cont.model">
    <xsd:sequence>
        <xsd:element ref="NOTE" minOccurs="0"/>
        <xsd:element ref="CONFID" minOccurs="0"/>
    </xsd:sequence>
</xsd:group>
<xsd:attributeGroup name="PPD_REAL-attrib.list">
    <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="PPD_REAL"/>
    <xsd:attribute name="NULL" type="null.code.set"/>
    <xsd:attribute name="V" type="xsd:string"/>
    <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="REAL"/>
    <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="value"/>
    <xsd:attribute name="SD" type="xsd:string"/>
    <xsd:attribute name="SD-T" type="xsd:NMTOKEN" fixed="REAL"/>
    <xsd:attribute name="SD-HL7_NAME" type="xsd:string" fixed="standardDeviation"/>
    <xsd:attribute name="TY">
        <xsd:simpleType>
            <xsd:restriction base="xsd:string">
                <xsd:enumeration value="U"/>
                <xsd:enumeration value="N"/>

```

```

        <xsd:enumeration value="LN"/>
        <xsd:enumeration value="G"/>
        <xsd:enumeration value="E"/>
        <xsd:enumeration value="X2"/>
        <xsd:enumeration value="T"/>
        <xsd:enumeration value="F"/>
        <xsd:enumeration value="B"/>
    </xsd:restriction>
</xsd:simpleType>
</xsd:attribute>
<xsd:attribute name="TY-T" type="xsd:NMTOKEN" fixed="CS"/>
<xsd:attribute name="TY-DOMAIN" type="xsd:string" fixed="2.16.840.1.113883.5.1019"/>
<xsd:attribute name="TY-HL7_NAME" type="xsd:string" fixed="type"/>
<xsd:attribute name="VT" type="xsd:string"/>
<xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="IVL TS"/>
<xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
<xsd:attribute name="PROB" type="xsd:string"/>
<xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
<xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
</xsd:attributeGroup>
<!--
===== PPD<PQ> =====

HL7 Processing Rules:
    @NULL or @V

Examples: needed!!!

The XML ITS does not support promoting a PQ to a PPD<PQ>
=====
-->
<xsd:group name="PPD_PQ-cont.model">
    <xsd:sequence>
        <xsd:element ref="NOTE" minOccurs="0"/>
        <xsd:element ref="CONFID" minOccurs="0"/>
    </xsd:sequence>
</xsd:group>
<xsd:attributeGroup name="PPD_PQ-attrib.list">
    <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="PPD_PQ"/>
    <xsd:attribute name="NULL" type="null.code.set"/>
    <xsd:attribute name="V" type="xsd:string"/>
    <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="REAL"/>
    <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="value"/>
    <xsd:attribute name="U" type="xsd:string"/>
    <xsd:attribute name="U-T" type="xsd:NMTOKEN" fixed="ST"/>
    <xsd:attribute name="U-HL7_NAME" type="xsd:string" fixed="unit"/>
    <xsd:attribute name="SD" type="xsd:string"/>
    <xsd:attribute name="SD-T" type="xsd:NMTOKEN" fixed="REAL"/>
    <xsd:attribute name="SD-HL7_NAME" type="xsd:string" fixed="standardDeviation"/>
    <xsd:attribute name="SDU" type="xsd:string"/>
    <xsd:attribute name="SDU-T" type="xsd:NMTOKEN" fixed="CS"/>
    <xsd:attribute name="SDU-HL7_NAME" type="xsd:string" fixed="standardDeviation unit"/>
    <xsd:attribute name="TY">
        <xsd:simpleType>
            <xsd:restriction base="xsd:string">
                <xsd:enumeration value="U"/>
                <xsd:enumeration value="N"/>
                <xsd:enumeration value="LN"/>
                <xsd:enumeration value="G"/>
                <xsd:enumeration value="E"/>
                <xsd:enumeration value="X2"/>
                <xsd:enumeration value="T"/>
                <xsd:enumeration value="F"/>
                <xsd:enumeration value="B"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
    <xsd:attribute name="TY-T" type="xsd:NMTOKEN" fixed="CS"/>
    <xsd:attribute name="TY-DOMAIN" type="xsd:string" fixed="2.16.840.1.113883.5.1019"/>
    <xsd:attribute name="TY-HL7_NAME" type="xsd:string" fixed="type"/>
    <xsd:attribute name="VT" type="xsd:string"/>
    <xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="IVL TS"/>
    <xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
    <xsd:attribute name="PROB" type="xsd:string"/>
    <xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
    <xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
</xsd:attributeGroup>
<!--
===== PPD<TS> =====

HL7 Processing Rules:

```

```

@NULL or @V

Examples: needed!!!

The XML ITS does not support promoting a TS to a PPD<TS>
=====
-->
<xsd:group name="PPD_TS-cont.model">
  <xsd:sequence>
    <xsd:element ref="NOTE" minOccurs="0"/>
    <xsd:element ref="CONFID" minOccurs="0"/>
  </xsd:sequence>
</xsd:group>
<xsd:attributeGroup name="PPD_TS-attrib.list">
  <xsd:attribute name="T" type="xsd:NMTOKEN" fixed="PPD_TS"/>
  <xsd:attribute name="NULL" type="null.code.set"/>
  <xsd:attribute name="V" type="xsd:string"/>
  <xsd:attribute name="V-T" type="xsd:NMTOKEN" fixed="REAL"/>
  <xsd:attribute name="V-HL7_NAME" type="xsd:string" fixed="value"/>
  <xsd:attribute name="SD" type="xsd:string"/>
  <xsd:attribute name="SD-T" type="xsd:NMTOKEN" fixed="REAL"/>
  <xsd:attribute name="SD-HL7_NAME" type="xsd:string" fixed="standardDeviation"/>
  <xsd:attribute name="SDU" type="xsd:string"/>
  <xsd:attribute name="SDU-T" type="xsd:NMTOKEN" fixed="CS"/>
  <xsd:attribute name="SDU-HL7_NAME" type="xsd:string" fixed="standardDeviation unit"/>
  <xsd:attribute name="TY">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="U"/>
        <xsd:enumeration value="N"/>
        <xsd:enumeration value="LN"/>
        <xsd:enumeration value="G"/>
        <xsd:enumeration value="E"/>
        <xsd:enumeration value="X2"/>
        <xsd:enumeration value="T"/>
        <xsd:enumeration value="F"/>
        <xsd:enumeration value="B"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
  <xsd:attribute name="TY-T" type="xsd:NMTOKEN" fixed="CS"/>
  <xsd:attribute name="TY-DOMAIN" type="xsd:string" fixed="2.16.840.1.113883.5.1019"/>
  <xsd:attribute name="TY-HL7_NAME" type="xsd:string" fixed="type"/>
  <xsd:attribute name="VT" type="xsd:string"/>
  <xsd:attribute name="VT-T" type="xsd:NMTOKEN" fixed="IVL_TS"/>
  <xsd:attribute name="VT-HL7_NAME" type="xsd:string" fixed="validTime"/>
  <xsd:attribute name="PROB" type="xsd:string"/>
  <xsd:attribute name="PROB-T" type="xsd:NMTOKEN" fixed="REAL"/>
  <xsd:attribute name="PROB-HL7_NAME" type="xsd:string" fixed="probability"/>
</xsd:attributeGroup>
<!--
  while some of these elements could be declared up with
  their main datatype, others can't since the para-entties
  need for them wouldn't yet have been defined; hence, we
  just put them all here.

  Also, there are most likely HL7-PR's associated with some
  (all?) of these properties, but I haven't had the time to
  check into that
-->
<!--
  elements required for ANT<T>
-->
<xsd:element name="NOTE">
  <xsd:complexType>
    <xsd:group ref="CE-cont.model"/>
    <xsd:attributeGroup ref="CE-attrib.list"/>
    <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="note"/>
  </xsd:complexType>
</xsd:element>
<!--
  elements required for UVN<T>
-->
<xsd:element name="CONFID">
  <xsd:complexType>
    <xsd:group ref="CV-cont.model"/>
    <xsd:attributeGroup ref="CV-attrib.list"/>
    <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="confidence"/>
  </xsd:complexType>
</xsd:element>
<!--

```

```

elements required for ED properties
-->
<xsd:element name="REF">
  <xsd:complexType>
    <xsd:group ref="TEL-cont.model"/>
    <xsd:attributeGroup ref="TEL-attrib.list"/>
    <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="reference"/>
  </xsd:complexType>
</xsd:element>
<!--
  Following the ITS note in the abstract ballot, any properties
  of the THUMBNAIL not specified in the instance are to be "inherited"
  from the enclosing ED element.

  Note: actually, the ITS note only mentions the type, charset and compression
  properties, but I think it equally applies to all properties
-->
<xsd:element name="THUMBNAIL">
  <xsd:complexType mixed="true">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element ref="REF"/>
      <xsd:element ref="THUMBNAIL"/>
      <xsd:element ref="NOTE"/>
      <xsd:element ref="CONFID"/>
    </xsd:choice>
    <xsd:attributeGroup ref="ED-attrib.list"/>
    <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="thumbnail"/>
  </xsd:complexType>
</xsd:element>
<!--
  elements required for CD properties (and related types)
-->
<xsd:element name="ORIGTXT">
  <xsd:complexType mixed="true">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element ref="REF"/>
      <xsd:element ref="THUMBNAIL"/>
      <xsd:element ref="NOTE"/>
      <xsd:element ref="CONFID"/>
    </xsd:choice>
    <xsd:attributeGroup ref="ED-attrib.list"/>
    <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="originalText"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="MODIFIER">
  <xsd:complexType>
    <xsd:group ref="CR-cont.model"/>
    <xsd:attributeGroup ref="CR-attrib.list"/>
    <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="modifier"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="TRANSLTN">
  <xsd:complexType>
    <xsd:group ref="CD-cont.model"/>
    <xsd:attributeGroup ref="CD-attrib.list"/>
    <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="translation"/>
  </xsd:complexType>
</xsd:element>
<!--
  elements required for CR properties
-->
<xsd:element name="NAME">
  <xsd:complexType>
    <xsd:group ref="CV-cont.model"/>
    <xsd:attributeGroup ref="CV-attrib.list"/>
    <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="name"/>
  </xsd:complexType>
</xsd:element>
<!--
  elements required for II properties
-->
<xsd:element name="TYPE">
  <xsd:complexType>
    <xsd:group ref="CV-cont.model"/>
    <xsd:attributeGroup ref="CV-attrib.list"/>
    <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="type"/>
  </xsd:complexType>
</xsd:element>
<!--
  elements required for AD properties
-->

```



```

<xsd:element name="LIT">
  <xsd:complexType>
    <xsd:group ref="ADXP-cont.model"/>
    <xsd:attributeGroup ref="ADXP-attrib.list"/>
    <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="literal"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="DEL">
  <xsd:complexType>
    <xsd:group ref="ADXP-cont.model"/>
    <xsd:attributeGroup ref="ADXP-attrib.list"/>
    <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="delimiter"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="CNT">
  <xsd:complexType>
    <xsd:group ref="ADXP-cont.model"/>
    <xsd:attributeGroup ref="ADXP-attrib.list"/>
    <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="country"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="CTY">
  <xsd:complexType>
    <xsd:group ref="ADXP-cont.model"/>
    <xsd:attributeGroup ref="ADXP-attrib.list"/>
    <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="city"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="STA">
  <xsd:complexType>
    <xsd:group ref="ADXP-cont.model"/>
    <xsd:attributeGroup ref="ADXP-attrib.list"/>
    <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="state"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="ZIP">
  <xsd:complexType>
    <xsd:group ref="ADXP-cont.model"/>
    <xsd:attributeGroup ref="ADXP-attrib.list"/>
    <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="postal code"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="STR">
  <xsd:complexType>
    <xsd:group ref="ADXP-cont.model"/>
    <xsd:attributeGroup ref="ADXP-attrib.list"/>
    <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="street name"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="HNR">
  <xsd:complexType>
    <xsd:group ref="ADXP-cont.model"/>
    <xsd:attributeGroup ref="ADXP-attrib.list"/>
    <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="house number"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="DIR">
  <xsd:complexType>
    <xsd:group ref="ADXP-cont.model"/>
    <xsd:attributeGroup ref="ADXP-attrib.list"/>
    <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="direction"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="ADL">
  <xsd:complexType>
    <xsd:group ref="ADXP-cont.model"/>
    <xsd:attributeGroup ref="ADXP-attrib.list"/>
    <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="address locator"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="POB">
  <xsd:complexType>
    <xsd:group ref="ADXP-cont.model"/>
    <xsd:attributeGroup ref="ADXP-attrib.list"/>
    <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="post office box"/>
  </xsd:complexType>
</xsd:element>
<!--
  elements required for PN properties
-->
<xsd:element name="FAM">

```

```

    <xsd:complexType>
      <xsd:group ref="PNXP-cont.model"/>
      <xsd:attributeGroup ref="PNXP-attrib.list"/>
      <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="family"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="GIV">
    <xsd:complexType>
      <xsd:group ref="PNXP-cont.model"/>
      <xsd:attributeGroup ref="PNXP-attrib.list"/>
      <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="given"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="MID">
    <xsd:complexType>
      <xsd:group ref="PNXP-cont.model"/>
      <xsd:attributeGroup ref="PNXP-attrib.list"/>
      <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="middle"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="PFX">
    <xsd:complexType>
      <xsd:group ref="PNXP-cont.model"/>
      <xsd:attributeGroup ref="PNXP-attrib.list"/>
      <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="prefix"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="SFX">
    <xsd:complexType>
      <xsd:group ref="PNXP-cont.model"/>
      <xsd:attributeGroup ref="PNXP-attrib.list"/>
      <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="suffix"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="PN.DEL">
    <xsd:complexType>
      <xsd:group ref="PNXP-cont.model"/>
      <xsd:attributeGroup ref="PNXP-attrib.list"/>
      <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="delimiter"/>
    </xsd:complexType>
  </xsd:element>
  <!--
    elements required for RTO properties
  -->
  <xsd:element name="NUM">
    <xsd:complexType>
      <xsd:group ref="QTY-cont.model"/>
      <xsd:attributeGroup ref="QTY-attrib.list"/>
      <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="numerator"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="DENOM">
    <xsd:complexType>
      <xsd:group ref="QTY-cont.model"/>
      <xsd:attributeGroup ref="QTY-attrib.list"/>
      <xsd:attribute name="HL7_NAME" type="xsd:string" fixed="denominator"/>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>

```

2.4 Sample CDA Documents

The following included sample documents are minimal CDA documents. Example 1 contains required elements only. It shows the proper usage of the CDA schemas and namespaces. Example 2 is a little richer CDA instance document.

Example 1

```
<?xml version="1.0" ?>
<levelone xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <clinical_document_header>
    <id EX="53123456" RT="2.16.840.1.113883.3.7.2.34556.1"/>
    <document_type_cd V="11490-0" S="2.16.840.1.113883.6.1" DN="Discharge Note"/>
    <origination_dttm V="20010412"/>
    <provider>
      <provider.type_cd V="CON"/>
      <person>
        <id EX="123456" RT="2.16.840.1.113883.3.7.2.12345.1.2"/>
      </person>
    </provider>
    <patient>
      <patient.type_cd V="PATSBJ"/>
      <person>
        <id EX="993534233" RT="2.16.840.1.113883.3.7.2.12345.1.2"/>
      </person>
    </patient>
  </clinical_document_header>
  <body>
    <section>
      <caption>Discharge Note</caption>
      <paragraph>
        <content> This is a very short note. </content>
      </paragraph>
    </section>
  </body>
</levelone>
```

Example 2

```
<?xml version="1.0"?>
<levelone xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <clinical_document_header>
    <id EX="a123" RT="2.16.840.1.113883.3.933"/>
    <document_type_cd V="11488-4" S="2.16.840.1.113883.6.1" DN="Consultation note"/>
    <origination_dttm V="20000407"/>
    <patient_encounter>
      <id EX="KPENC1332" RT="2.16.840.1.113883.3.933"/>
      <practice_setting_cd V="GIM" S="2.16.840.1.113883.5.10588"
        DN="General internal medicine clinic"/>
      <encounter_tmr V="20000407"/>
    </patient_encounter>
    <legal_authenticator>
      <legal_authenticator.type_cd V="SPV"/>
      <participation_tmr V="20000408"/>
      <signature_cd V="S"/>
      <person>
        <id EX="KP00017" RT="2.16.840.1.113883.3.933"/>
        <person_name>
          <nm>
            <GIV V="Robert"/>
            <FAM V="Dolin"/>
            <SFX V="MD" QUAL="PT"/>
          </nm>
          <person_name.type_cd V="I" S="2.16.840.1.113883.5.200"/>
        </person_name>
      </person>
    </legal_authenticator>
    <originator>
      <originator.type_cd V="AUT"/>
      <participation_tmr V="20000407"/>
      <person>
        <id EX="KP00017" RT="2.16.840.1.113883.3.933"/>
      </person>
    </originator>
```

```

<originating_organization>
  <originating_organization.type_cd V="CST"/>
  <organization>
    <id EX="M345" RT="2.16.840.1.113883.3.933"/>
    <organization.nm V="Good Health Clinic"/>
  </organization>
</originating_organization>
<provider>
  <provider.type_cd V="CON"/>
  <participation_tmr V="20000407"/>
  <person>
    <id EX="KP00017" RT="2.16.840.1.113883.3.933"/>
  </person>
</provider>
<patient>
  <patient.type_cd V="PATSBJ"/>
  <person>
    <id EX="12345" RT="2.16.840.1.113883.3.933"/>
    <person_name>
      <nm>
        <GIV V="Henry"/>
        <FAM V="Levin"/>
        <SFX V="the 7th"/>
      </nm>
      <person_name.type_cd V="I" S="2.16.840.1.113883.5.200"/>
    </person_name>
  </person>
  <birth_dttm V="19320924"/>
  <administrative_gender_cd V="M" S="2.16.840.1.113883.5.1"/>
</patient>
</clinical_document_header>
<body>
  <section>
    <caption>History of Present Illness</caption>
    <paragraph>
      <content>Henry Levin, the 7th is a 67 year old male referred for further
        asthma management. Onset of asthma in his teens. He was hospitalized
        twice last year, and already twice this year. He has not been able to
        be weaned off steroids for the past several months.
      </content>
    </paragraph>
  </section>
  <section>
    <caption>Past Medical History</caption>
    <list>
      <item>
        <content>Asthma</content>
      </item>
      <item>
        <content>Hypertension</content>
      </item>
      <item>
        <content>Osteoarthritis, right knee</content>
      </item>
    </list>
  </section>
  <section>
    <caption>Medications</caption>
    <list>
      <item>
        <content>Theodur 200mg BID</content>
      </item>
      <item>
        <content>Proventil inhaler 2puffs QID PRN</content>
      </item>
      <item>
        <content>Prednisone 20mg qd</content>
      </item>
      <item>
        <content>HCTZ 25mg qd</content>
      </item>
    </list>
  </section>
  <section>
    <caption>Allergies</caption>
    <list>
      <item>
        <content>Penicillin - Hives</content>
      </item>
      <item>
        <content>Aspirin - Wheezing</content>
      </item>
    </list>
  </section>

```

```

    </item>
  </list>
</section>
<section>
  <caption>Social History</caption>
  <list>
    <item>
      <content>
        Smoking :: 1 PPD between the ages of 20 and 55, and then he quit.
      </content>
    </item>
    <item>
      <content>Alcohol :: rare</content>
    </item>
  </list>
</section>
<section>
  <caption>Physical Examination</caption>
  <section>
    <caption>Vital Signs</caption>
    <list>
      <item>
        <content>BP 118/78</content>
      </item>
      <item>
        <content>Resp 16 and unlabored</content>
      </item>
      <item>
        <content>T 98.6F</content>
      </item>
      <item>
        <content>HR 86 and regular</content>
      </item>
    </list>
  </section>
  <section>
    <caption>Skin</caption>
    <paragraph>
      <content>Erythematous rash, palmar surface, left index finger.
        <observation_media>
          <observation_media.value MT="image/jpeg">
            <REF V="rash.jpeg"/>
          </observation_media.value>
        </observation_media>
      </content>
    </paragraph>
  </section>
  <section>
    <caption>Lungs</caption>
    <paragraph>
      <content>Clear with no wheeze. Good air flow.</content>
    </paragraph>
  </section>
  <section>
    <caption>Cardiac</caption>
    <paragraph>
      <content>RRR with no murmur, no S3, no S4.</content>
    </paragraph>
  </section>
</section>
<section>
  <caption>Labs</caption>
  <list>
    <item>
      <content>
        CXR 02/03/1999: Hyperinflated. Normal cardiac silhouette, clear lungs.
      </content>
    </item>
    <item>
      <content>Peak Flow today: 260 l/m</content>
    </item>
  </list>
</section>
<section>
  <caption>Assessment</caption>
  <list>
    <item>
      <content>
        Asthma, with prior smoking
        history. Difficulty weaning off steroids. Will try gradual taper.
        <coded_entry>

```

```

        <coded_entry.value V="D2-51000" S="2.16.840.1.113883.6.5"/>
      </coded_entry>
    </content>
  </item>
  <item>
    <content>Hypertension, well-controlled.</content>
  </item>
  <item>
    <content>Contact dermatitis on finger.</content>
  </item>
</list>
</section>
<section>
  <caption>Plan</caption>
  <list>
    <item>
      <content>Complete PFTs with lung volumes.</content>
    </item>
    <item>
      <content>Chem-7</content>
    </item>
    <item>
      <content>
        Provide educational material on inhaler usage and peak flow self-monitoring.
      </content>
    </item>
    <item>
      <content>Decrease prednisone to 20qOD alternating with 18qOD.</content>
    </item>
    <item>
      <content>Hydrocortisone cream to finger BID.</content>
    </item>
    <item>
      <content>RTC 1 week.</content>
    </item>
  </list>
</section>
</body>
</levelone>

```

3 References

- [rfHL7CDA] Alschuler L, Dolin RH, Boyer S, Beebe C (eds).
HL7 Clinical Document Architecture Framework, Release 1.0.
ANSI approved HL7 Standard, Nov 2000.
<http://www.hl7.org>
- [rfBJ] Dolin RH, Alschuler L, Beebe C, Biron PV, Boyer SL, Essin D, Kimber E,
Lincoln T, Mattison JE
The HL7 Clinical Document Architecture
J Am Med Inform Assoc 2001 Nov-Dec;8(6):552-69
- [rfXML] Extensible Markup Language (XML) 1.0. W3C Recommendation 10-February-
1998.
<http://www.w3.org/TR/1998/REC-xml-19980210.html>
- [rfXSDtool] XML Schema tools (dtd2xsd, xsbrowse), downloadable from the official W3C site
<http://www.w3.org/XML/Schema#Tools>
- [rfDTDrs] Schweiger R, Hoelzer S, Heitmann KU, Dudeck J.
DTDs go XML Schema – A tools perspective
Medical Informatics and the Internet in Medicine, 2002, in press
- [rfXMLSchema] XML Schema, W3C Recommendation May 2, 2001
<http://www.w3.org/XML>
<http://www.w3.org/2001/XMLSchema.html>