This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!--  ALTO: Analyzed Layout and Text Object   -->
<!--  This document is available under the Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0 - https://creativecommons.org/licenses/by-sa/4.0/ ).
The ALTO Editorial Board has waived all rights to it worldwide under copyright law with confirmation of the original creating authors, including all related and neighboring rights, to the extent allowed by law.
For the full text see https://creativecommons.org/licenses/by-sa/4.0/legalcode.  -->
<!--  Originally created during the EU-funded Project METAe, the Metadata Engine Project (2001 - 2003), by Alexander Egger (1), Birgit Stehno (2) and Gregor Retti (2), (1) University of Graz and (2) University of Innsbruck, Austria with contributions of Ralph Tiede, CCS GmbH, Germany  -->
<!--  Prepared for the Library of Congress by Ralph Tiede, CCS GmbH, with the assistance of Justin Littman (Library of Congress).  -->
<!--  Version 4.4  -->
<!--  Change History  -->
<!--  June 22, 2004: Version finalized for docWORKS/METAe  -->
<!--  November 19, 2004: Modifications requested by Justin Littman  -->
<!--  Modifications of November 19, 2004:
        1. add "Description" element
        2. change "InnerMargin/OuterMargin" to "LeftMargin/RightMargin", add "POSITION" attribute to "PAGE" element
        3. add "PROCESSING" attribute to "PAGE" element
        4. internal changes to validate with Xerces parser
        5. define fontstyles by enumerations
        6. change "WC" (word confidence) attribute to xsd:float in range of "0" to "1".
        7. Add "ALTERNATIVE" as children to "STRING" element
        8. Add "language" attribute to "Textblock" and "STRING" element
 -->
<!--  Modifications of December 02, 2004:
        1. fixed problem with multiple use of blockgroup
        2. add measurement enumeration 'inch1200'
 -->
<!--  Modifications of December 14, 2004:
        1. "FILEID" (attribute of "ComposedBlock"): change type from xsd:IDREF to xsd:string
        2. include minor changes requested by JDL
        3. change "ZORDER" to "IDNEXT" (attribute of "BlockType")
 -->
<!--  Modifications of February 24, 2006:
        1. ACCURACY attribute added to PAGE element to store information on OCR accuracy
        2. CS attribute added to TEXTLINE element to indicate manual correction status
 -->
<!--  Modifications of June 20, 2007 (version 1.3):
        1. Adaption of xlink namespace and schema location to prevent conflicts on XSL transformations in combination with used namespace in original METS file
 -->
<!--  Modifications of August 27, 2007 (version 1.4):
        1. add "QUALITY_DETAIL" attribute to "PAGE" element (gives more details about the page quality, is a free string comparing with QUALITY attribute which is a restrictive one)
        2. add "Cover" to "POSITION" attribute of "PAGE" element
        3. specification of interpretation of confidence values (CC, WC, PC and ACCURACY)
 -->
<!--  Modifications of August 7, 2009:
        1. Change namespace from old CCS URI to LC-based URI.
        2. Use standard LC XLink Schema.
        3. Push version to 2.0 to reflect change in maintenance agency.
        4. Remove CCS copyright statement.
        5. Rollback to model used in 1.4 schema except with the changes itemized in 1-4 of this change note.  An incorrect version of the 2.0 alpha schema was public until 2010-01-11.  The incorrect version was a derivative of the Library of Congress's custom ALTO XML Schema that introduced new elements and attributes.
 -->
<!--  Modifications of January 11, 2010:
        1. Rollback to model used in 1.4 schema except with the changes itemized in 1-4 of the previous change note of August 7, 2009.  An incorrect version of the 2.0 alpha schema was public until 2010-01-11.  The incorrect version was a derivative of the Library of Congress's custom ALTO XML Schema that introduced new elements and attri
 -->
<!--  February 20, 2014, version 2.1:
        1. Page and BlockType element HEIGHT, WIDTH, HPOS, VPOS attribute types changed to xsd:float from xsd:int.
        2. CircleType  HPOS, VPOS and RADIUS attribute type definitions added as xsd:float and made mandatory. Element annotation clarified.
        3. EllipseType HPOS,VPOS,HLENGTH and VLENGTH attribute type definitions added as xsd:float and made mandatory. Element annotation clarified.
        4. MeasurementUnit defined as mandatory and element annotation clarified.
        5. HYP element's CONTENT attribute type definition added as xsd:string.
        6. Tags (LayoutTag/StructureTag/RoleTag/NamedEntityTag/OtherTag) added to allow for tagging content. TAGREFS attribute added to BlockTypes, TextLine and String
        7. CS attribute added to String and Block.
        8. LANG attribute added to String, TextLine and TextBlock. "language" attribute in TextBlock deprecated.
        9. HEIGHT attribute added to HYP and SP elements.
 -->
<!--  April, 2014, version 2.2 DRAFT:
        1. Anonymous types changed to named types (to allow use of xsd:redefine mechanism)
 -->
<!--  July 2014, version 2.2 DRAFT
        1. Version added to xsd:schema.
        2. SCHEMAVERSION attribute added to <alto> element.
        3. documentIdentifier element added to <sourceImageInformationType> element (+ documentIdentifierLocation attribute)
 -->
<!--  August 2014, version 3.0
        1. Changed namespace and targetNamespace to http://www.loc.gov/standards/alto/ns-v3#
        2. Changed schema version to 3.0

        ALTO schemas will be updated by whole numbers upon making changes that break backward compatibility (version 1 to version 2),
        and decimals for changes that will not (2.0 to 2.1). The namespace itself will also only change on major versions (ns-v2 to ns-v3).
 -->
<!--  January 2016, version 3.1
        1. Changed schema version to 3.1
        2. Added support for using different shapes for the elements String, TextLine, all PageSpaceType elements and on all BlockType elements.
        3. The description of the attribute ROTATION is changed to the rotation of the contents of a block and not the block itself. The attribute is inherited by all sub elements.
 -->
<!--  January 2018, version 4.0
        1. Changed schema version to 4.0
        2. Changed namespace and targetNamespace to http://www.loc.gov/standards/alto/ns-v4#
        3. Clarification and definition of the licensing to common standard "CC BY-SA 4.0" for this ALTO standard (with agreement of the authors)
        4. Added character based text description with new Glyph element and its subelement Variant (GlyphType, VariantType)
        5. Extended annotation for clarification of the difference of existing element ALTERNATIVE and Glyph/Variant
        6. Introduce generic "Processing" and deprecate "OcrProcessing"
        7. Introduce generic "processingStep" with "ProcessingStepType" and required attribute "ID" and deprecate "preProcessingStep", "ocrProcessingStep", "postProcessingStep"
        8. Add common vocabulary for "processingStep" comprising the "ContentGeneration", "ContentModification", "PreOperation", "PostOperation", "Other"
        9. Fix for the element Shape. The Shape element can now only be used once within a PageSpace or a TextLine as it was intended.
 -->
<!--  May 2019, version 4.1
        1. Fix for Processing including  processingStepType.
        2. Add missing PROCESSINGREFS to PageType, PageSpaceType, BlockType, TextLine, StringType for referencing Processing history.
 -->
<!--  June/July 2020, version 4.2
        1. Change BASELINE to accommodate a list of points in addition to a single point.
        2. Make FONTSIZE optional.
        3. Add "strikethrough" to list of allowed values for FONTSTYLE.
 -->
<!--  May 2022, version 4.3
        1. Add BASEDIRECTION attribute defining base direction and line orientation to TextLine and BlockType.
        2. Add support for explicit reading order definitions with "ReadingOrder" element containing "UnorderedGroup"s, "OrderedGroup"s, and "ElementRef"s.
 -->
<!--  March 2023, version 4.4
        1. Add LANG attribute on PageType level to describe the default language used in document
        2. Add ROTATION attribute on PageType level to describe the default rotation used in document
        3. Add OTHERLANGS attribute on PageType to summarize all the languages present into a particular document
        4. Adapt "PointsType" documentation
        5. Adapt xLink attribute group documentation on "BlockType"
 -->
<xsd:schema xmlns="http://www.loc.gov/standards/alto/ns-v4#" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xlink="http://www.w3.org/1999/xlink" targetNamespace="http://www.loc.gov/standards/alto/ns-v4#" elementFormDefault="qualified" attributeFormDefault="unqualified" version="4.4">
  <xsd:import namespace="http://www.w3.org/1999/xlink" schemaLocation="http://www.loc.gov/standards/xlink/xlink.xsd"/>
  <xsd:element name="alto" type="altoType">
    <xsd:annotation>
      <xsd:documentation>ALTO (analyzed layout and text object) stores layout information and OCR recognized text of pages of any kind of printed documents like books, journals and newspapers. ALTO is a standardized XML format to store layout and content information. It is designed to be used as an extension schema to METS (Metadata
      Encoding and Transmission Standard), where METS provides metadata and structural information while ALTO contains content and physical information. </xsd:documentation>
    </xsd:annotation>
```

```
</xsd:element>
<xsd:complexType name="altoType">
  <xsd:sequence>
    <xsd:element name="Description" type="DescriptionType" minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>Describes general settings of the alto file like measurement units and metadata</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="Styles" type="StylesType" minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>Styles define properties of layout elements. A style defined in a parent element is used as default style for all related children elements. </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="Tags" type="TagsType" minOccurs="0">
      <xsd:annotation>
        <xsd:documentation> Tag define properties of additional characteristic. The tags are referenced from related content element on Block or String element by attribute TAGREF via the tag ID. This container element contains the individual elements for LayoutTags, StructureTags, RoleTags, NamedEntityTags and OtherTags
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="ReadingOrder" type="ReadingOrderType" minOccurs="0">
      <xsd:annotation>
        <xsd:documentation> Describes alternative hierarchical orderings of the page (i.e. total orders over its segments, for linear text flow), in addition to the explicit flat reading order defined by @IDNEXT on the block level, and the implicit flat reading order implied by the segment element ordering. </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="Layout" type="LayoutType">
      <xsd:annotation>
        <xsd:documentation>The root layout element.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="SCHEMAVERSION" type="xsd:string" use="optional">
    <xsd:annotation>
      <xsd:documentation>Schema version of the ALTO file.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>
<xsd:complexType name="DescriptionType">
  <xsd:sequence>
    <xsd:element name="MeasurementUnit" type="MeasurementUnitType" minOccurs="1"/>
    <xsd:element name="sourceImageInformation" type="sourceImageInformationType" minOccurs="0"/>
    <xsd:element name="OCRProcessing" minOccurs="0" maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>Element deprecated. 'Processing' should be used instead.</xsd:documentation>
      </xsd:annotation>
      <xsd:complexType>
        <xsd:complexContent>
          <xsd:extension base="ocrProcessingType">
            <xsd:attribute name="ID" type="xsd:ID" use="required"/>
          </xsd:extension>
        </xsd:complexContent>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="Processing" minOccurs="0" maxOccurs="unbounded">
      <xsd:complexType>
        <xsd:complexContent>
          <xsd:extension base="processingStepType">
            <xsd:attribute name="ID" type="xsd:ID" use="required"/>
          </xsd:extension>
        </xsd:complexContent>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="StylesType">
  <xsd:sequence>
    <xsd:element name="TextStyle" type="TextStyleType" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="ParagraphStyle" type="ParagraphStyleType" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="TagsType">
  <xsd:annotation>
    <xsd:documentation> There are following variation of tag types available: LayoutTag – criteria about arrangement or graphical appearance StructureTag – criteria about grouping or formation RoleTag – criteria about function or mission NamedEntityTag – criteria about assignment of terms to their relationship / meaning (NER) OtherTag –
    criteria about any other characteristic not listed above, the TYPE attribute is intended to be used for classification within those. </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="LayoutTag" type="TagType"/>
      <xsd:element name="StructureTag" type="TagType"/>
      <xsd:element name="RoleTag" type="TagType"/>
      <xsd:element name="NamedEntityTag" type="TagType"/>
      <xsd:element name="OtherTag" type="TagType"/>
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ReadingOrderType">
  <xsd:annotation>
    <xsd:documentation> Defines one or more reading orders within the page. Groups may be either unordered or ordered and can contain other groups, e.g. a page containing unrelated texts that are ordered individually would be encoded as an UnorderedGroup containing multiple OrderedGroups. The granularity of elements can vary inside
    groups. </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="OrderedGroup" type="OrderedGroupType"/>
      <xsd:element name="UnorderedGroup" type="UnorderedGroupType"/>
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ElementRefType">
  <xsd:annotation>
    <xsd:documentation> A reference to an element such as a block, TextLine, String, or Glyph. </xsd:documentation>
  </xsd:annotation>
  <xsd:attribute name="ID" type="xsd:ID" use="required"/>
  <xsd:attribute name="REF" type="xsd:IDREFS" use="required">
    <xsd:annotation>
      <xsd:documentation> A link to the referenced element. Valid target elements are any block type, TextLine, String, or Glyph. </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="TAGREFS" type="xsd:IDREFS" use="optional">
    <xsd:annotation>
      <xsd:documentation> Optionally annotates the role of the referenced element in the reading order with one or more tags. Examples could be interlinear additions or marginalia. </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>
<xsd:complexType name="OrderedGroupType">
  <xsd:annotation>
    <xsd:documentation> A group containing ordered elements (i.e. the sequence of OrderedGroup, UnorderedGroup or ElementRef subelements is ordered). </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="ElementRef" type="ElementRefType"/>
      <xsd:element name="OrderedGroup" type="OrderedGroupType"/>
      <xsd:element name="UnorderedGroup" type="UnorderedGroupType"/>
    </xsd:choice>
  </xsd:sequence>
```

```
    <xsd:attribute name="ID" type="xsd:ID" use="required"/>
    <xsd:attribute name="TAGREFS" type="xsd:IDREFS" use="optional">
      <xsd:annotation>
        <xsd:documentation> Optionally annotates the role of the group in the reading order with one or more tags. Examples could be distinguishing parallel texts or apparatus criticus and main text. </xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="REF" type="xsd:IDREFS" use="optional">
      <xsd:annotation>
        <xsd:documentation> A link to the referenced element. Valid target elements are any block type, TextLine, or String. </xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:complexType>
  <xsd:complexType name="UnorderedGroupType">
    <xsd:annotation>
      <xsd:documentation> A group containing unordered elements (i.e. the sequence of OrderedGroup, UnorderedGroup or ElementRef subelements is arbitrary). </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="ElementRef" type="ElementRefType"/>
        <xsd:element name="OrderedGroup" type="OrderedGroupType"/>
        <xsd:element name="UnorderedGroup" type="UnorderedGroupType"/>
      </xsd:choice>
    </xsd:sequence>
    <xsd:attribute name="ID" type="xsd:ID" use="required"/>
    <xsd:attribute name="TAGREFS" type="xsd:IDREFS" use="optional"/>
    <xsd:attribute name="REF" type="xsd:IDREFS" use="optional">
      <xsd:annotation>
        <xsd:documentation> A link to the referenced element. Valid target elements are any block type, TextLine, or String. </xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:complexType>
  <xsd:simpleType name="QualityType">
    <xsd:annotation>
      <xsd:documentation>Gives brief information about original page quality</xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="OK"/>
      <xsd:enumeration value="Missing"/>
      <xsd:enumeration value="Missing in original"/>
      <xsd:enumeration value="Damaged"/>
      <xsd:enumeration value="Retained"/>
      <xsd:enumeration value="Target"/>
      <xsd:enumeration value="As in original"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="QualityDetailType">
    <xsd:annotation>
      <xsd:documentation>Gives more details about the original page quality, since QUALITY attribute gives only brief and restrictive information</xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name="PositionType">
    <xsd:annotation>
      <xsd:documentation>Position of the page. Could be lefthanded, righthanded, cover, foldout or single if it has no special position.</xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Left"/>
      <xsd:enumeration value="Right"/>
      <xsd:enumeration value="Foldout"/>
      <xsd:enumeration value="Single"/>
      <xsd:enumeration value="Cover"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="PCType">
    <xsd:annotation>
      <xsd:documentation>Page Confidence: Confidence level of the ocr for this page. A value between 0 (unsure) and 1 (sure). </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:float">
      <xsd:minInclusive value="0"/>
      <xsd:maxInclusive value="1"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:complexType name="PageType">
    <xsd:annotation>
      <xsd:documentation>One page of a book or journal.</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="TopMargin" type="PageSpaceType" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>The area between the top line of print and the upper edge of the leaf. It may contain page number or running title.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="LeftMargin" type="PageSpaceType" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>The area between the printspace and the left border of a page. May contain margin notes.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="RightMargin" type="PageSpaceType" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>The area between the printspace and the right border of a page. May contain margin notes.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="BottomMargin" type="PageSpaceType" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>The area between the bottom line of letterpress or writing and the bottom edge of the leaf. It may contain a page number, a signature number or a catch word.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="PrintSpace" type="PageSpaceType" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>Rectangle covering the printed area of a page. Page number and running title are not part of the print space. </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="ID" type="PageID" use="required"/>
    <xsd:attribute name="PAGECLASS" type="xsd:string" use="optional">
      <xsd:annotation>
        <xsd:documentation>Any user-defined class like title page.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="STYLEREFS" type="xsd:IDREFS" use="optional"/>
    <xsd:attribute name="PROCESSINGREFS" type="xsd:IDREFS" use="optional"/>
    <xsd:attribute name="HEIGHT" type="xsd:float" use="optional"/>
    <xsd:attribute name="WIDTH" type="xsd:float" use="optional"/>
    <xsd:attribute name="PHYSICAL_IMG_NR" type="xsd:float" use="required">
      <xsd:annotation>
        <xsd:documentation>The number of the page within the document.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="PRINTED_IMG_NR" type="xsd:string" use="optional">
      <xsd:annotation>
        <xsd:documentation>The page number that is printed on the page.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
```

```
    <xsd:attribute name="QUALITY" type="QualityType" use="optional"/>
    <xsd:attribute name="QUALITY_DETAIL" type="QualityDetailType" use="optional"/>
    <xsd:attribute name="POSITION" type="PositionType" use="optional"/>
    <xsd:attribute name="PROCESSING" type="xsd:IDREF" use="optional">
      <xsd:annotation>
        <xsd:documentation>A link to the processing description that has been used for this page.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="ACCURACY" type="xsd:float" use="optional">
      <xsd:annotation>
        <xsd:documentation>Estimated percentage of OCR Accuracy in range from 0 to 100 </xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="PC" type="PCType" use="optional"/>
    <xsd:attribute name="ROTATION" type="xsd:float" use="optional">
      <xsd:annotation>
        <xsd:documentation>Default rotation for text or illustrations on this page. The value is in degree counterclockwise. The default value can be overwritten on lower levels (Textblock, Textline, etc)</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="LANG" type="xsd:language" use="optional">
      <xsd:annotation>
        <xsd:documentation>Default language for text on this page. The default value can be overwritten on lower levels (Textblock, Textline, etc)</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="OTHERLANGS" type="ListOfLanguages" use="optional">
      <xsd:annotation>
        <xsd:documentation>Other languages that appear on this page. Provides a convenient way to summarize all the languages found on a particular page, without parsing the entire file</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:complexType>
  <xsd:simpleType name="ListOfLanguages">
    <xsd:list itemType="xsd:language"/>
  </xsd:simpleType>
  <xsd:complexType name="LayoutType">
    <xsd:sequence>
      <xsd:element name="Page" type="PageType" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="STYLEREFS" type="xsd:IDREFS"/>
  </xsd:complexType>
  <xsd:complexType name="TextStyleType">
    <xsd:annotation>
      <xsd:documentation>A text style defines font properties of text. </xsd:documentation>
    </xsd:annotation>
    <xsd:attribute name="ID" type="xsd:ID"/>
    <xsd:attributeGroup ref="formattingAttributeGroup"/>
  </xsd:complexType>
  <xsd:complexType name="ParagraphStyleType">
    <xsd:annotation>
      <xsd:documentation>A paragraph style defines formatting properties of text blocks.</xsd:documentation>
    </xsd:annotation>
    <xsd:attribute name="ID" type="ParagraphStyleID" use="required"/>
    <xsd:attribute name="ALIGN" use="optional">
      <xsd:annotation>
        <xsd:documentation>Indicates the alignment of the paragraph. Could be left, right, center or justify.</xsd:documentation>
      </xsd:annotation>
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Left"/>
          <xsd:enumeration value="Right"/>
          <xsd:enumeration value="Center"/>
          <xsd:enumeration value="Block"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
    <xsd:attribute name="LEFT" type="xsd:float" use="optional">
      <xsd:annotation>
        <xsd:documentation>Left indent of the paragraph in relation to the column.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="RIGHT" type="xsd:float" use="optional">
      <xsd:annotation>
        <xsd:documentation>Right indent of the paragraph in relation to the column.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="LINESPACE" type="xsd:float" use="optional">
      <xsd:annotation>
        <xsd:documentation>Line spacing between two lines of the paragraph. Measurement calculated from baseline to baseline.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="FIRSTLINE" type="xsd:float" use="optional">
      <xsd:annotation>
        <xsd:documentation>Indent of the first line of the paragraph if this is different from the other lines. A negative value indicates an indent to the left, a positive value indicates an indent to the right.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:complexType>
  <xsd:simpleType name="SPTypeID">
    <xsd:restriction base="xsd:ID"/>
  </xsd:simpleType>
  <xsd:simpleType name="PageSpaceTypeID">
    <xsd:restriction base="xsd:ID"/>
  </xsd:simpleType>
  <xsd:simpleType name="ParagraphStyleID">
    <xsd:restriction base="xsd:ID"/>
  </xsd:simpleType>
  <xsd:simpleType name="PageID">
    <xsd:restriction base="xsd:ID"/>
  </xsd:simpleType>
  <xsd:simpleType name="BlockTypeID">
    <xsd:restriction base="xsd:ID"/>
  </xsd:simpleType>
  <xsd:simpleType name="StringTypeID">
    <xsd:restriction base="xsd:ID"/>
  </xsd:simpleType>
  <xsd:simpleType name="TextLineID">
    <xsd:restriction base="xsd:ID"/>
  </xsd:simpleType>
  <xsd:group name="BlockGroup">
    <xsd:annotation>
      <xsd:documentation>Group of available block types</xsd:documentation>
    </xsd:annotation>
    <xsd:choice>
      <xsd:element name="TextBlock" type="TextBlockType">
        <xsd:annotation>
          <xsd:documentation>A block of text.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="Illustration" type="IllustrationType">
        <xsd:annotation>
          <xsd:documentation>A picture or image.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="GraphicalElement" type="GraphicalElementType">
        <xsd:annotation>
```

```
          <xsd:documentation>A graphic used to separate blocks. Usually a line or rectangle.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="ComposedBlock" type="ComposedBlockType">
        <xsd:annotation>
          <xsd:documentation>A block that consists of other blocks</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:choice>
  </xsd:group>
  <xsd:complexType name="BlockType">
    <xsd:annotation>
      <xsd:documentation>Base type for any kind of block on the page.</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence minOccurs="0">
      <xsd:element name="Shape" type="ShapeType"/>
    </xsd:sequence>
    <xsd:attribute name="ID" type="BlockTypeID" use="required"/>
    <xsd:attribute name="STYLEREFS" type="xsd:IDREFS"/>
    <xsd:attribute name="TAGREFS" type="xsd:IDREFS" use="optional"/>
    <xsd:attribute name="PROCESSINGREFS" type="xsd:IDREFS" use="optional"/>
    <xsd:attribute name="HEIGHT" type="xsd:float" use="optional"/>
    <xsd:attribute name="WIDTH" type="xsd:float" use="optional"/>
    <xsd:attribute name="HPOS" type="xsd:float" use="optional"/>
    <xsd:attribute name="VPOS" type="xsd:float" use="optional"/>
    <xsd:attribute name="ROTATION" type="xsd:float" use="optional">
      <xsd:annotation>
        <xsd:documentation>Tells the rotation of e.g. text or illustration within the block. The value is in degree counterclockwise.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="IDNEXT" type="xsd:IDREF" use="optional">
      <xsd:annotation>
        <xsd:documentation>The next block in reading order of the page (if ReadingOrder is not specified, and elements are not in order).</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="CS" type="xsd:boolean" use="optional">
      <xsd:annotation>
        <xsd:documentation>Correction Status. Indicates whether manual correction has been done or not. The correction status should be recorded at the highest level possible (Block, TextLine, String).</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attributeGroup ref="xlink:simpleLink">
      <xsd:annotation>
        <xsd:documentation>Attribute group deprecated. Planned to be removed in future versions due to issues created on mixed validation and because in practice it is not used very often</xsd:documentation>
      </xsd:annotation>
    </xsd:attributeGroup>
  </xsd:complexType>
  <xsd:complexType name="SPType">
    <xsd:annotation>
      <xsd:documentation>A white space.</xsd:documentation>
    </xsd:annotation>
    <xsd:attribute name="ID" type="SPTypeID" use="optional"/>
    <xsd:attribute name="HEIGHT" type="xsd:float" use="optional"/>
    <xsd:attribute name="WIDTH" type="xsd:float" use="optional"/>
    <xsd:attribute name="HPOS" type="xsd:float" use="optional"/>
    <xsd:attribute name="VPOS" type="xsd:float" use="optional"/>
  </xsd:complexType>
  <xsd:simpleType name="SUBS_TYPEType">
    <xsd:annotation>
      <xsd:documentation>Type of the substitution (if any).</xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="HypPart1"/>
      <xsd:enumeration value="HypPart2"/>
      <xsd:enumeration value="Abbreviation"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="CONTENTType">
    <xsd:restriction base="xsd:string">
      <xsd:whiteSpace value="preserve"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="WCType">
    <xsd:annotation>
      <xsd:documentation>Word Confidence: Confidence level of the ocr for this string. A value between 0 (unsure) and 1 (sure). </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:float">
      <xsd:minInclusive value="0"/>
      <xsd:maxInclusive value="1"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:complexType name="ALTERNATIVEType">
    <xsd:annotation>
      <xsd:documentation> Any alternative for the word. Alternative can outline a variant of writing by new typing / spelling rules, typically manually done or by dictionary replacements. The above sample is an old composed character "Ǻ" of ancient time, which is replaced now by "Ä". As variant are meant alternatives of the real printed
      content which are options outlined by the text recognition process. Similar sample: "Straße" vs. "Strasse". Such alternatives are not coming from text recognition. </xsd:documentation>
    </xsd:annotation>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="PURPOSE" type="xsd:string" use="optional">
          <xsd:annotation>
            <xsd:documentation>Identifies the purpose of the alternative.</xsd:documentation>
          </xsd:annotation>
        </xsd:attribute>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
  <xsd:complexType name="StringType" mixed="false">
    <xsd:annotation>
      <xsd:documentation>A sequence of chars. Strings are separated by white spaces or hyphenation chars.</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence minOccurs="0">
      <xsd:element name="Shape" type="ShapeType" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="ALTERNATIVE" type="ALTERNATIVEType" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="Glyph" type="GlyphType" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="ID" type="StringTypeID" use="optional"/>
    <xsd:attribute name="STYLEREFS" type="xsd:IDREFS" use="optional"/>
    <xsd:attribute name="TAGREFS" type="xsd:IDREFS" use="optional"/>
    <xsd:attribute name="PROCESSINGREFS" type="xsd:IDREFS" use="optional"/>
    <xsd:attribute name="HEIGHT" type="xsd:float" use="optional"/>
    <xsd:attribute name="WIDTH" type="xsd:float" use="optional"/>
    <xsd:attribute name="HPOS" type="xsd:float" use="optional"/>
    <xsd:attribute name="VPOS" type="xsd:float" use="optional"/>
    <xsd:attribute name="CONTENT" type="CONTENTType" use="required"/>
    <xsd:attribute name="STYLE" type="fontStylesType" use="optional"/>
    <xsd:attribute name="SUBS_TYPE" type="SUBS_TYPEType" use="optional"/>
    <xsd:attribute name="SUBS_CONTENT" type="xsd:string" use="optional">
      <xsd:annotation>
        <xsd:documentation>Content of the substitution.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="WC" type="WCType" use="optional"/>
    <xsd:attribute name="CC" type="xsd:string" use="optional">
      <xsd:annotation>
```

```
      <xsd:documentation>Confidence level of each character in that string. A list of numbers, one number between 0 (sure) and 9 (unsure) for each character.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="CS" type="xsd:boolean" use="optional">
    <xsd:annotation>
      <xsd:documentation>Correction Status. Indicates whether manual correction has been done or not. The correction status should be recorded at the highest level possible (Block, TextLine, String).</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="LANG" type="xsd:language" use="optional">
    <xsd:annotation>
      <xsd:documentation>Attribute to record language of the string. The language should be recorded at the highest level possible.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>
<xsd:complexType name="PageSpaceType">
  <xsd:annotation>
    <xsd:documentation>A region on a page</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="Shape" type="ShapeType" minOccurs="0" maxOccurs="1"/>
    <xsd:sequence minOccurs="0" maxOccurs="unbounded">
      <xsd:group ref="BlockGroup"/>
    </xsd:sequence>
  </xsd:sequence>
  <xsd:attribute name="ID" type="PageSpaceTypeID" use="optional"/>
  <xsd:attribute name="STYLEREFS" type="xsd:IDREFS" use="optional"/>
  <xsd:attribute name="PROCESSINGREFS" type="xsd:IDREFS" use="optional"/>
  <xsd:attribute name="HEIGHT" type="xsd:float" use="optional"/>
  <xsd:attribute name="WIDTH" type="xsd:float" use="optional"/>
  <xsd:attribute name="HPOS" type="xsd:float" use="optional"/>
  <xsd:attribute name="VPOS" type="xsd:float" use="optional"/>
</xsd:complexType>
<xsd:simpleType name="PointsType">
  <xsd:annotation>
    <xsd:documentation>A list of coordinate-pairs that are absolute to the upper-left corner of a page.</xsd:documentation>
    <xsd:documentation>The upper left corner of the page is defined as x=0 and y=0</xsd:documentation>
    <xsd:documentation>Currently there are no rules to enforce a particular format for a points list but in future versions is planned to restrict it to following options:</xsd:documentation>
    <xsd:documentation>"x1,y1 x2,y2 ... xn,yn" - highly recommended as widely used and easy to read by both human and machine</xsd:documentation>
    <xsd:documentation>"x1 y1 x2 y2 ... xn yn" - kept for back compatibility, since currently there are tools using this format</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:complexType name="ShapeType">
  <xsd:annotation>
    <xsd:documentation>Describes the bounding shape of a block, if it is not rectangular.</xsd:documentation>
  </xsd:annotation>
  <xsd:choice>
    <xsd:element name="Polygon" type="PolygonType"/>
    <xsd:element name="Ellipse" type="EllipseType"/>
    <xsd:element name="Circle" type="CircleType"/>
  </xsd:choice>
</xsd:complexType>
<xsd:simpleType name="InlineDirType">
  <xsd:annotation>
    <xsd:documentation>Describes the inline base direction and line orientation of a line or of all lines inside a text block.</xsd:documentation>
    <xsd:documentation>
      The meaning of these terms is defined by the W3C writing modes document:
      <a href="https://www.w3.org/TR/css-writing-modes-3/#writing-mode"/>
    </xsd:documentation>
    <xsd:documentation>These values should correspond to the base direction set in the BiDi algorithm to the respective elements during Unicode encoding. A value of "ttb" (top-to-bottom) implies a base direction of left-to-right, a value of "btt" (bottom-to-top) a base direction of right-to-left.</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="ltr"/>
    <xsd:enumeration value="rtl"/>
    <xsd:enumeration value="ttb"/>
    <xsd:enumeration value="btt"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="PolygonType">
  <xsd:annotation>
    <xsd:documentation>A polygon shape.</xsd:documentation>
  </xsd:annotation>
  <xsd:attribute name="POINTS" type="PointsType" use="required"/>
</xsd:complexType>
<xsd:complexType name="EllipseType">
  <xsd:annotation>
    <xsd:documentation>An ellipse shape. HPOS and VPOS describe the center of the ellipse. HLENGTH and VLENGTH are the width and height of the described ellipse.</xsd:documentation>
    <xsd:documentation>The attribute ROTATION tells the rotation of the e.g. text or illustration within the block. The value is in degrees counterclockwise. </xsd:documentation>
  </xsd:annotation>
  <xsd:attribute name="HPOS" type="xsd:float" use="required"/>
  <xsd:attribute name="VPOS" type="xsd:float" use="required"/>
  <xsd:attribute name="HLENGTH" type="xsd:float" use="required"/>
  <xsd:attribute name="VLENGTH" type="xsd:float" use="required"/>
  <xsd:attribute name="ROTATION" type="xsd:float" use="optional"/>
</xsd:complexType>
<xsd:complexType name="CircleType">
  <xsd:annotation>
    <xsd:documentation>A circle shape. HPOS and VPOS describe the center of the circle.</xsd:documentation>
  </xsd:annotation>
  <xsd:attribute name="HPOS" type="xsd:float" use="required"/>
  <xsd:attribute name="VPOS" type="xsd:float" use="required"/>
  <xsd:attribute name="RADIUS" type="xsd:float" use="required"/>
</xsd:complexType>
<xsd:attributeGroup name="formattingAttributeGroup">
  <xsd:annotation>
    <xsd:documentation>Formatting attributes. Note that these attributes are assumed to be inherited from ancestor elements of the document hierarchy.</xsd:documentation>
  </xsd:annotation>
  <xsd:attribute name="FONTFAMILY" type="xsd:string" use="optional">
    <xsd:annotation>
      <xsd:documentation>The font name.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="FONTTYPE" type="fontTypeType" use="optional"/>
  <xsd:attribute name="FONTWIDTH" type="fontWidthType" use="optional"/>
  <xsd:attribute name="FONTSIZE" type="xsd:float" use="optional">
    <xsd:annotation>
      <xsd:documentation>The font size, in points (1/72 of an inch).</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="FONTCOLOR" type="xsd:hexBinary" use="optional">
    <xsd:annotation>
      <xsd:documentation>Font color as RGB value</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="FONTSTYLE" type="fontStylesType" use="optional"/>
</xsd:attributeGroup>
<xsd:simpleType name="fontTypeType">
  <xsd:annotation>
    <xsd:documentation>Serif or Sans-Serif</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="serif"/>
    <xsd:enumeration value="sans-serif"/>
```

```
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="fontWidthType">
    <xsd:annotation>
      <xsd:documentation>fixed or proportional</xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="proportional"/>
      <xsd:enumeration value="fixed"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="MeasurementUnitType">
    <xsd:annotation>
      <xsd:documentation> All measurement values inside the alto file are related to this unit, except the font size. Coordinates as being used in HPOS and VPOS are absolute coordinates referring to the upper-left corner of a page. The upper left corner of the page is defined as coordinate (0/0). values meaning: mm10: 1/10th of millimeter
      inch1200: 1/1200th of inch pixel: 1 pixel The values for pixel will be related to the resolution of the image based on which the layout is described. In case the original image is not known the scaling factor can be calculated based on total width and height of the image and the according information of the PAGE element.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="pixel"/>
      <xsd:enumeration value="mm10"/>
      <xsd:enumeration value="inch1200"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:complexType name="sourceImageInformationType">
    <xsd:annotation>
      <xsd:documentation>Information to identify the image file from which the OCR text was created.</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="fileName" type="fileNameType" minOccurs="0"/>
      <xsd:element name="fileIdentifier" type="fileIdentifierType" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="documentIdentifier" type="documentIdentifierType" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:simpleType name="fileNameType">
    <xsd:restriction base="xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name="fileIdentifierValueType">
    <xsd:restriction base="xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name="fileIdentifierLocationValueType">
    <xsd:restriction base="xsd:string"/>
  </xsd:simpleType>
  <xsd:complexType name="fileIdentifierType">
    <xsd:annotation>
      <xsd:documentation>A unique identifier for the image file. This is drawn from MIX.</xsd:documentation>
      <xsd:documentation> This identifier must be unique within the local system. To facilitate file sharing or interoperability with other systems, fileIdentifierLocation may be added to designate the system or application where the identifier is unique.</xsd:documentation>
    </xsd:annotation>
    <xsd:simpleContent>
      <xsd:extension base="fileIdentifierValueType">
        <xsd:attribute name="fileIdentifierLocation" type="fileIdentifierLocationValueType">
          <xsd:annotation>
            <xsd:documentation>A location qualifier, i.e., a namespace.</xsd:documentation>
          </xsd:annotation>
        </xsd:attribute>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
  <xsd:simpleType name="documentIdentifierValueType">
    <xsd:restriction base="xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name="documentIdentifierLocationValueType">
    <xsd:restriction base="xsd:string"/>
  </xsd:simpleType>
  <xsd:complexType name="documentIdentifierType">
    <xsd:annotation>
      <xsd:documentation>A unique identifier for the document.</xsd:documentation>
      <xsd:documentation> This identifier must be unique within the local system. To facilitate file sharing or interoperability with other systems, documentIdentifierLocation may be added to designate the system or application where the identifier is unique.</xsd:documentation>
    </xsd:annotation>
    <xsd:simpleContent>
      <xsd:extension base="documentIdentifierValueType">
        <xsd:attribute name="documentIdentifierLocation" type="documentIdentifierLocationValueType">
          <xsd:annotation>
            <xsd:documentation>A location qualifier, i.e., a namespace.</xsd:documentation>
          </xsd:annotation>
        </xsd:attribute>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
  <xsd:complexType name="ocrProcessingType">
    <xsd:annotation>
      <xsd:documentation>Deprecated. processingStepType should be used instead.</xsd:documentation>
      <xsd:documentation>Information on how the text was created, including preprocessing, OCR processing, and postprocessing steps. Where possible, this draws from MIX's change history.</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="preProcessingStep" type="processingStepType" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="ocrProcessingStep" type="processingStepType"/>
      <xsd:element name="postProcessingStep" type="processingStepType" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="processingStepType">
    <xsd:annotation>
      <xsd:documentation>Description of the processing step.</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="processingCategory" type="processingCategoryType" minOccurs="0" maxOccurs="1">
        <xsd:annotation>
          <xsd:documentation>Classification of the category of operation, how the file was created, including generation, modification, preprocessing, postprocessing or any other steps.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="processingDateTime" type="dateTimeType" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>Date or DateTime the image was processed.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="processingAgency" type="xsd:string" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>Identifies the organizationlevel producer(s) of the processed image.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="processingStepDescription" type="xsd:string" minOccurs="0" maxOccurs="unbounded">
        <xsd:annotation>
          <xsd:documentation>An ordinal listing of the image processing steps performed. For example, "image despeckling."</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="processingStepSettings" type="xsd:string" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>A description of any setting of the processing application. For example, for a multi-engine OCR application this might include the engines which were used. Ideally, this description should be adequate so that someone else using the same application can produce identical results.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="processingSoftware" type="processingSoftwareType" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:simpleType name="processingCategoryType">
```

```
    <xsd:list>
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="contentGeneration"/>
          <xsd:enumeration value="contentModification"/>
          <xsd:enumeration value="preOperation"/>
          <xsd:enumeration value="postOperation"/>
          <xsd:enumeration value="other"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:list>
  </xsd:simpleType>
  <xsd:complexType name="processingSoftwareType">
    <xsd:annotation>
      <xsd:documentation>Information about a software application. Where applicable, the preferred method for determining this information is by selecting Help -- About.</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="softwareCreator" type="xsd:string" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>The name of the organization or company that created the application.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="softwareName" type="xsd:string" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>The name of the application.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="softwareVersion" type="xsd:string" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>The version of the application.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="applicationDescription" type="xsd:string" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>A description of any important characteristics of the application, especially for non-commercial applications. For example, if a non-commercial application is built using commercial components, e.g., an OCR engine SDK. Those components should be mentioned here.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:simpleType name="dateTimeType">
    <xsd:union memberTypes="xsd:date xsd:dateTime xsd:gYear xsd:gYearMonth"/>
  </xsd:simpleType>
  <xsd:simpleType name="fontStylesType">
    <xsd:annotation>
      <xsd:documentation>List of any combination of font styles</xsd:documentation>
    </xsd:annotation>
    <xsd:restriction>
      <xsd:simpleType>
        <xsd:list>
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="bold"/>
              <xsd:enumeration value="italics"/>
              <xsd:enumeration value="smallcaps"/>
              <xsd:enumeration value="strikethrough"/>
              <xsd:enumeration value="subscript"/>
              <xsd:enumeration value="superscript"/>
              <xsd:enumeration value="underline"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:list>
      </xsd:simpleType>
      <xsd:minLength value="1"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:complexType name="ComposedBlockType">
    <xsd:annotation>
      <xsd:documentation>A block that consists of other blocks</xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="BlockType">
        <xsd:sequence minOccurs="0" maxOccurs="unbounded">
          <xsd:group ref="BlockGroup"/>
        </xsd:sequence>
        <xsd:attribute name="TYPE" type="xsd:string" use="optional">
          <xsd:annotation>
            <xsd:documentation>A user defined string to identify the type of composed block (e.g. table, advertisement, ...)</xsd:documentation>
          </xsd:annotation>
        </xsd:attribute>
        <xsd:attribute name="FILEID" type="xsd:string" use="optional">
          <xsd:annotation>
            <xsd:documentation>An ID to link to an image which contains only the composed block. The ID and the file link is defined in the related METS file.</xsd:documentation>
          </xsd:annotation>
        </xsd:attribute>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:complexType name="IllustrationType">
    <xsd:annotation>
      <xsd:documentation>A picture or image.</xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="BlockType">
        <xsd:attribute name="TYPE" type="xsd:string" use="optional">
          <xsd:annotation>
            <xsd:documentation>A user defined string to identify the type of illustration like photo, map, drawing, chart, ...</xsd:documentation>
          </xsd:annotation>
        </xsd:attribute>
        <xsd:attribute name="FILEID" type="xsd:string" use="optional">
          <xsd:annotation>
            <xsd:documentation>A link to an image which contains only the illustration.</xsd:documentation>
          </xsd:annotation>
        </xsd:attribute>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:complexType name="GraphicalElementType">
    <xsd:annotation>
      <xsd:documentation>A graphic used to separate blocks. Usually a line or rectangle. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="BlockType"/>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:complexType name="TextBlockType">
    <xsd:annotation>
      <xsd:documentation>A block of text.</xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="BlockType">
        <xsd:sequence minOccurs="0">
          <xsd:element name="TextLine" maxOccurs="unbounded">
            <xsd:annotation>
```

```
            <xsd:documentation>A single line of text.</xsd:documentation>
          </xsd:annotation>
          <xsd:complexType>
            <xsd:sequence>
              <xsd:sequence>
                <xsd:element name="Shape" type="ShapeType" minOccurs="0" maxOccurs="1"/>
              </xsd:sequence>
              <xsd:sequence maxOccurs="unbounded">
                <xsd:element name="String" type="StringType"/>
                <xsd:element name="SP" type="SPType" minOccurs="0"/>
              </xsd:sequence>
              <xsd:element name="HYP" minOccurs="0">
                <xsd:annotation>
                  <xsd:documentation>A hyphenation char. Can appear only at the end of a line.</xsd:documentation>
                </xsd:annotation>
                <xsd:complexType>
                  <xsd:attribute name="HEIGHT" type="xsd:float" use="optional"/>
                  <xsd:attribute name="WIDTH" type="xsd:float" use="optional"/>
                  <xsd:attribute name="HPOS" type="xsd:float" use="optional"/>
                  <xsd:attribute name="VPOS" type="xsd:float" use="optional"/>
                  <xsd:attribute name="CONTENT" type="xsd:string" use="required"/>
                </xsd:complexType>
              </xsd:element>
            </xsd:sequence>
            <xsd:attribute name="ID" type="TextLineID"/>
            <xsd:attribute name="STYLEREFS" type="xsd:IDREFS" use="optional"/>
            <xsd:attribute name="TAGREFS" type="xsd:IDREFS" use="optional"/>
            <xsd:attribute name="PROCESSINGREFS" type="xsd:IDREFS" use="optional"/>
            <xsd:attribute name="HEIGHT" type="xsd:float" use="optional"/>
            <xsd:attribute name="WIDTH" type="xsd:float" use="optional"/>
            <xsd:attribute name="HPOS" type="xsd:float" use="optional"/>
            <xsd:attribute name="VPOS" type="xsd:float" use="optional"/>
            <xsd:attribute name="BASELINE" type="PointsType" use="optional">
              <xsd:annotation>
                <xsd:documentation>Pixel coordinates based on the left-hand top corner of an image which define a polyline on which a line of text rests.</xsd:documentation>
              </xsd:annotation>
            </xsd:attribute>
            <xsd:attribute name="LANG" type="xsd:language" use="optional">
              <xsd:annotation>
                <xsd:documentation>Attribute to record language of the textline.</xsd:documentation>
              </xsd:annotation>
            </xsd:attribute>
            <xsd:attribute name="CS" type="xsd:boolean" use="optional">
              <xsd:annotation>
                <xsd:documentation>Correction Status. Indicates whether manual correction has been done or not. The correction status should be recorded at the highest level possible (Block, TextLine, String).</xsd:documentation>
              </xsd:annotation>
            </xsd:attribute>
            <xsd:attribute name="BASEDIRECTION" type="InlineDirType" use="optional">
              <xsd:annotation>
                <xsd:documentation>Indicates the inline base direction of this TextLine. Overrides the value on elements higher in the hierarchy.</xsd:documentation>
              </xsd:annotation>
            </xsd:attribute>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
      <xsd:attribute name="language" type="xsd:language" use="optional">
        <xsd:annotation>
          <xsd:documentation>Attribute deprecated. LANG should be used instead.</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="LANG" type="xsd:language" use="optional">
        <xsd:annotation>
          <xsd:documentation>Attribute to record language of the textblock.</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="BASEDIRECTION" type="InlineDirType" use="optional">
        <xsd:annotation>
          <xsd:documentation>Indicates the inline base direction of the TextBlock.</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="TagType">
  <xsd:sequence>
    <xsd:element name="XmlData" minOccurs="0">
      <xsd:annotation>
        <xsd:documentation xml:lang="en"> The xml data wrapper element XmlData is used to contain XML encoded metadata. The content of an XmlData element can be in any namespace or in no namespace. As permitted by the XML Schema Standard, the processContents attribute value for the metadata in an XmlData is set to “lax”. Therefore, if
        the source schema and its location are identified by means of an XML schemaLocation attribute, then an XML processor will validate the elements for which it can find declarations. If a source schema is not identified, or cannot be found at the specified schemaLocation, then an XML validator will check for well-formedness, but
        otherwise skip over the elements appearing in the XmlData element. </xsd:documentation>
      </xsd:annotation>
      <xsd:complexType>
        <xsd:sequence>
          <xsd:any namespace="##any" processContents="lax" maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="ID" type="xsd:ID" use="required"/>
  <xsd:attribute name="TYPE" type="xsd:string" use="optional">
    <xsd:annotation>
      <xsd:documentation>Type can be used to classify and group the information within each tag element type.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="LABEL" type="xsd:string" use="required">
    <xsd:annotation>
      <xsd:documentation>Content / information value of the tag.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="DESCRIPTION" type="xsd:string" use="optional">
    <xsd:annotation>
      <xsd:documentation>Description text for tag information for clarification.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="URI" type="xsd:anyURI" use="optional">
    <xsd:annotation>
      <xsd:documentation>Any URI for authority or description relevant information.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>
<xsd:complexType name="GlyphType" mixed="false">
  <xsd:annotation>
    <xsd:documentation> Modern OCR software stores information on glyph level. A glyph is essentially a character or ligature. Accordingly the value for the glyph element will be defined as follows: Pre-composed representation = base + combining character(s) (decomposed representation) See
    http://www.fileformat.info/info/unicode/char/0101/index.htm "U+0101" = (U+0061) + (U+0304) "combining characters" ("base characters" in combination with non-spacing marks or characters which are combined to one) are represented as one "glyph", e.g. ā≈ā. Each glyph has its own coordinate information and must be separately addressable
    as a distinct object. Correction and verification processes can be carried out for individual characters. Post-OCR analysis of the text as well as adaptive OCR algorithm must be able to record information on glyph level. In order to reproduce the decision of the OCR software, optional characters must be recorded. These are called
    variants. The OCR software evaluates each variant and picks the one with the highest confidence score as the glyph. The confidence score expresses how confident the OCR software is that a single glyph had been recognized correctly. The glyph elements are in order of the word. Each glyph need to be recorded to built up the whole word
    sequence. The glyph’s CONTENT attribute is no replacement for the string’s CONTENT attribute. Due to post-processing steps such as correction the values of both attributes may be inconsistent. </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence minOccurs="0">
    <xsd:element name="Shape" type="ShapeType" minOccurs="0"/>
    <xsd:element name="Variant" type="VariantType" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="ID" type="xsd:ID" use="optional"/>
```

```
      <xsd:attribute name="CONTENT" use="required">
        <xsd:annotation>
          <xsd:documentation> CONTENT contains the precomposed representation (combining character) of the character from the parent String element. The sequence position of the Glyph element matches the position of the character in the String. </xsd:documentation>
        </xsd:annotation>
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:length fixed="true" value="1"/>
            <xsd:whiteSpace value="preserve"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
      <xsd:attribute name="GC" use="optional">
        <xsd:annotation>
          <xsd:documentation> This GC attribute records a float value between 0.0 and 1.0 that expresses the level of confidence for the glyph where 1 is certain. This attribute is optional. If it is not available, the default value for the glyph is “0”. The GC attribute semantic is the same as the WC attribute on the String element and VC
          on Variant element. </xsd:documentation>
        </xsd:annotation>
        <xsd:simpleType>
          <xsd:restriction base="xsd:float">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="1"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
      <xsd:attribute name="HEIGHT" type="xsd:float" use="optional"/>
      <xsd:attribute name="WIDTH" type="xsd:float" use="optional"/>
      <xsd:attribute name="HPOS" type="xsd:float" use="optional"/>
      <xsd:attribute name="VPOS" type="xsd:float" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="VariantType" mixed="false">
      <xsd:annotation>
        <xsd:documentation> Alternative (combined) character for the glyph, outlined by OCR engine or similar recognition processes. In case the variant are two (combining) characters, two characters are outlined in one Variant element. E.g. a Glyph element with CONTENT="m" can have a Variant element with the content "rn". Details for
        different use-cases see on the samples on GitHub. </xsd:documentation>
      </xsd:annotation>
      <xsd:attribute name="CONTENT" use="optional">
        <xsd:annotation>
          <xsd:documentation> Each Variant represents an option for the glyph that the OCR software detected as possible alternatives. In case the variant are two (combining) characters, two characters are outlined in one Variant element. E.g. a Glyph element with CONTENT="m" can have a Variant element with the content "rn". Details for
          different use-cases see on the samples on GitHub. </xsd:documentation>
        </xsd:annotation>
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:maxLength value="3"/>
            <xsd:whiteSpace value="preserve"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
      <xsd:attribute name="VC" use="optional">
        <xsd:annotation>
          <xsd:documentation> This VC attribute records a float value between 0.0 and 1.0 that expresses the level of confidence for the variant where is 1 is certain. This attribute is optional. If it is not available, the default value for the variant is “0”. The VC attribute semantic is the same as the GC attribute on the Glyph element.
          </xsd:documentation>
        </xsd:annotation>
        <xsd:simpleType>
          <xsd:restriction base="xsd:float">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="1"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
    </xsd:complexType>
  </xsd:schema>
```