XSD to UML

Washington D.C. | October 10 - 11, 2007

AIXM Class | 2007

AIXM 5 RC2

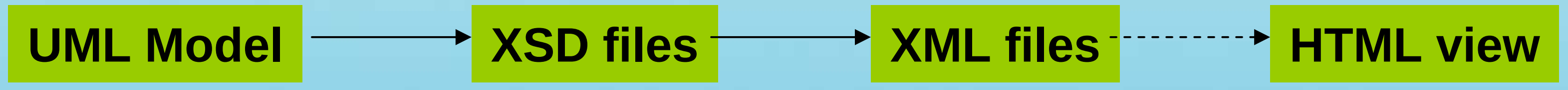
EUROCONTROL

FEDERAL AVIATION ADMINISTRATION

Topics

- Examples of AIXM xml data
- Overview of XML Schema Definition Language (XSDL)
  - Schema components
  - AIXM use of XSDL
- Examples from the AIXM 5 RC2 xsd files
  - Representative XSDL datatypes
  - VerticalStructure (a generalization on structures that includes obstacles)

General Scheme


UML Model

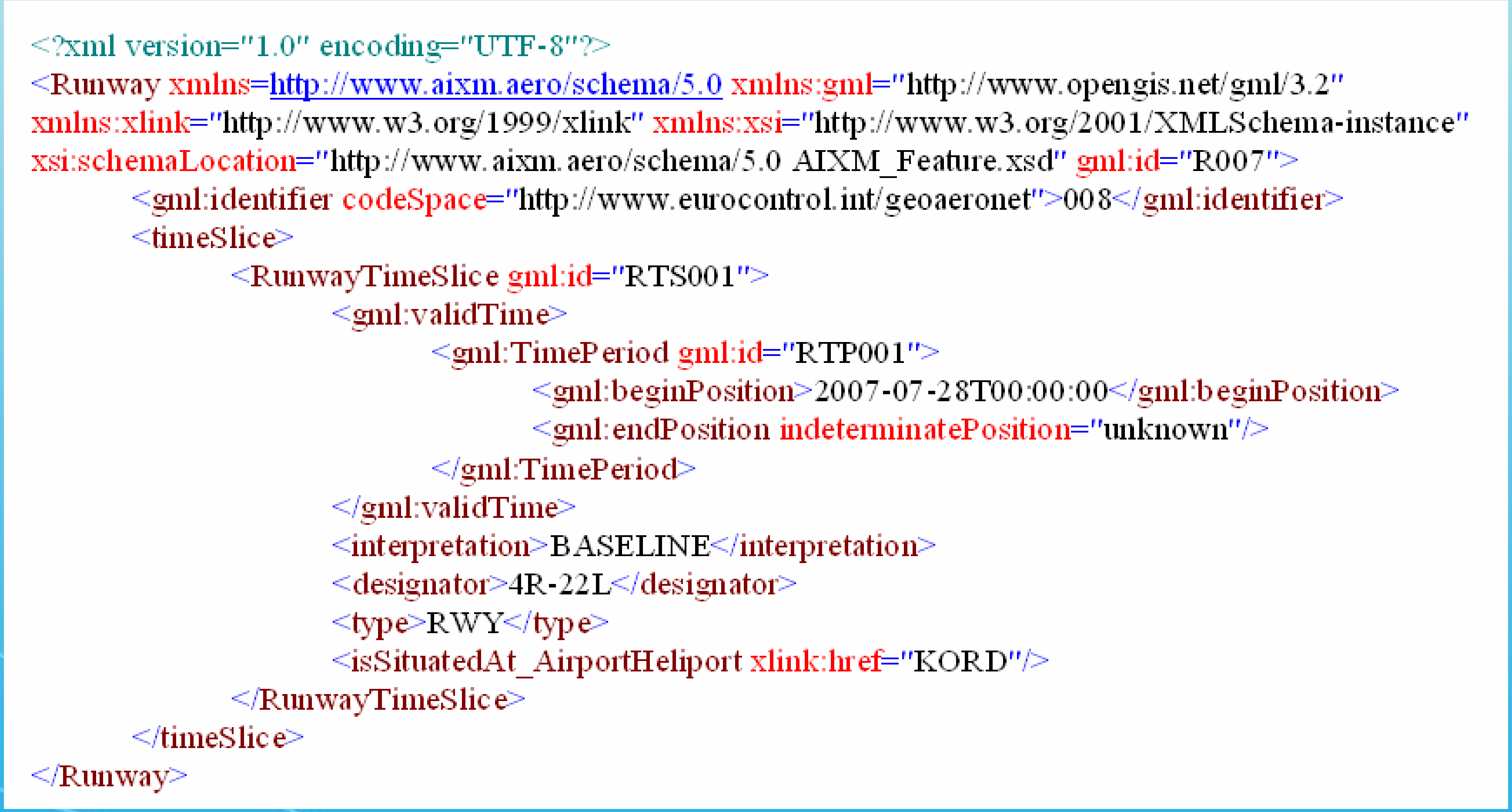
XSD files

XML files

HTML view

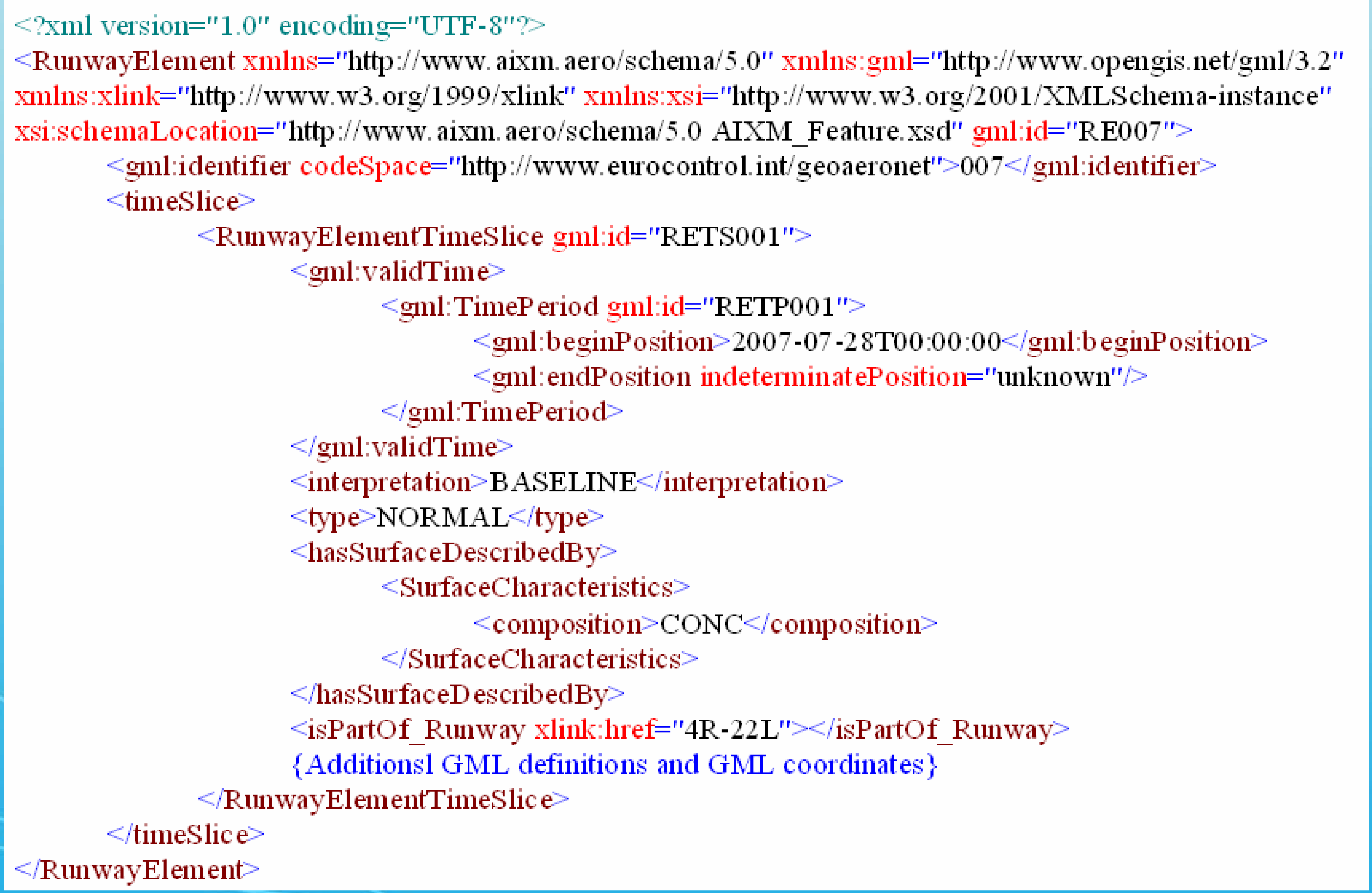
- AIXM is modelled using UML
- Scripts within the UML model are used to generate XSD files. These files act as schema definitions for the AIXM data
- The AIXM data is defined as XML files
- Software (Java, XSLT, etc) is used to transform the XML data to HTML to be viewed by a browser

Sample AIXM Data: Runway



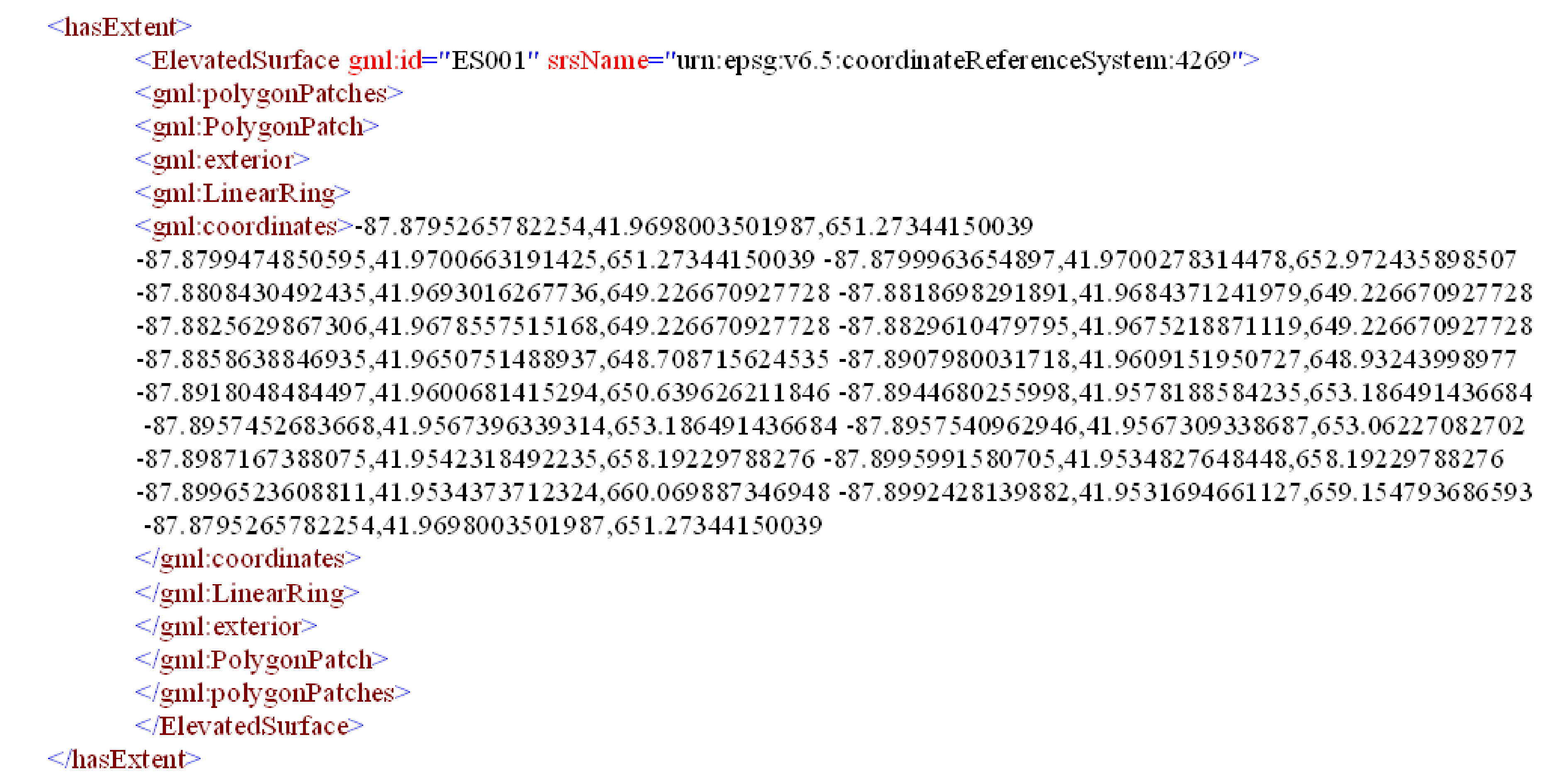
```
<?xml version="1.0" encoding="UTF-8"?>
<Runway xmlns=http://www.aixm.aero/schema/5.0 xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aixm.aero/schema/5.0 AIXM_Feature.xsd" gml:id="R007">
        <gml:identifier codeSpace="http://www.eurocontrol.int/geoaeronet">008</gml:identifier>
        <timeSlice>
                <RunwayTimeSlice gml:id="RTS001">
                        <gml:validTime>
                                <gml:TimePeriod gml:id="RTP001">
                                        <gml:beginPosition>2007-07-28T00:00:00</gml:beginPosition>
                                        <gml:endPosition indeterminatePosition="unknown"/>
                                </gml:TimePeriod>
                        </gml:validTime>
                        <interpretation>BASELINE</interpretation>
                        <designator>4R-22L</designator>
                        <type>RWY</type>
                        <isSituatedAt_AirportHeliport xlink:href="KORD"/>
                </RunwayTimeSlice>
        </timeSlice>
</Runway>
```

Sample AIXM Data: Runway Element



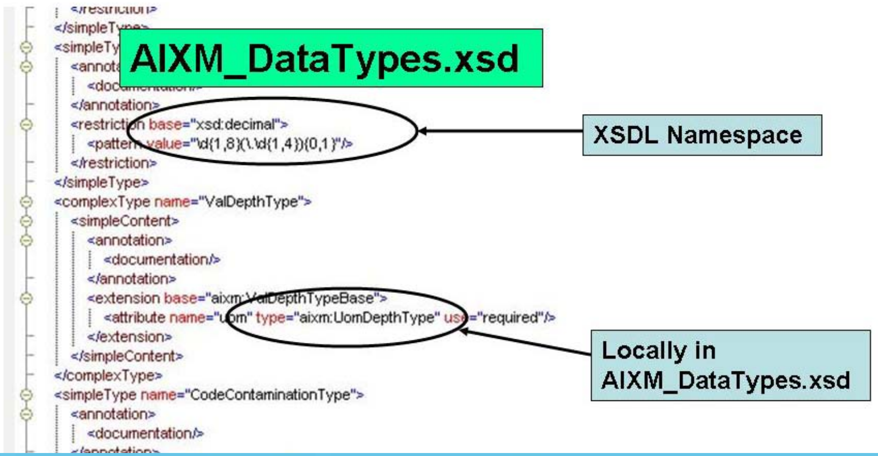
```
<?xml version="1.0" encoding="UTF-8"?>
<RunwayElement xmlns="http://www.aixm.aero/schema/5.0" xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.aixm.aero/schema/5.0 AIXM_Feature.xsd" gml:id="RE007">
        <gml:identifier codeSpace="http://www.eurocontrol.int/geoaeronet">007</gml:identifier>
        <timeSlice>
                <RunwayElementTimeSlice gml:id="RETS001">
                        <gml:validTime>
                                <gml:TimePeriod gml:id="RETP001">
                                        <gml:beginPosition>2007-07-28T00:00:00</gml:beginPosition>
                                        <gml:endPosition indeterminatePosition="unknown"/>
                                </gml:TimePeriod>
                        </gml:validTime>
                        <interpretation>BASELINE</interpretation>
                        <type>NORMAL</type>
                        <hasSurfaceDescribedBy>
                                <SurfaceCharacteristics>
                                        <composition>CONC</composition>
                                </SurfaceCharacteristics>
                        </hasSurfaceDescribedBy>
                        <isPartOf_Runway xlink:href="4R-22L"></isPartOf_Runway>
                        {Additionsl GML definitions and GML coordinates}
                </RunwayElementTimeSlice>
        </timeSlice>
</RunwayElement>
```

Sample AIXM Data: Runway Element



```
<hasExtent>
          <ElevatedSurface gml:id="ES001" srsName="urn:epsg:v6.5:coordinateReferenceSystem:4269">
          <gml:polygonPatches>
          <gml:PolygonPatch>
          <gml:exterior>
          <gml:LinearRing>
          <gml:coordinates>-87.8795265782254,41.9698003501987,651.27344150039
          -87.8799474850595,41.9700663191425,651.27344150039 -87.8799963654897,41.9700278314478,652.972435898507
          -87.8808430492435,41.969301626773 6,649.22667092772 8 -87.8818698291891,41.9684371241979,649.22667092772 8
          -87.8825629867306,41.9678557515168,649.22667092772 8 -87.8829610479795,41.9675218871119,649.22667092772 8
          -87.8858638846935,41.9650751488937,648.70871562453 5 -87.8907980031718,41.9609151950727,648.9324399897 7
          -87.8918048484497,41.9600681415294,650.639626211846 -87.8944680255998,41.9578188584235,653.186491436684
           -87.8957452683668,41.9567396339314,653.186491436684 -87.8957540962946,41.9567309338687,653.06227082702
          -87.8987167388075,41.9542318492235,658.19229788276 -87.8995991580705,41.9534827648448,658.19229788276
          -87.8996523608811,41.9534373712324,660.069887346948 -87.8992428139882,41.9531694661127,659.15479368659 3
           -87.8795265782254,41.9698003501987,651.27344150039
          </gml:coordinates>
          </gml:LinearRing>
          </gml:exterior>
          </gml:PolygonPatch>
          </gml:polygonPatches>
          </ElevatedSurface>
</hasExtent>
```

AIXM 5 Release Candidate 2 (RC2) xsd files

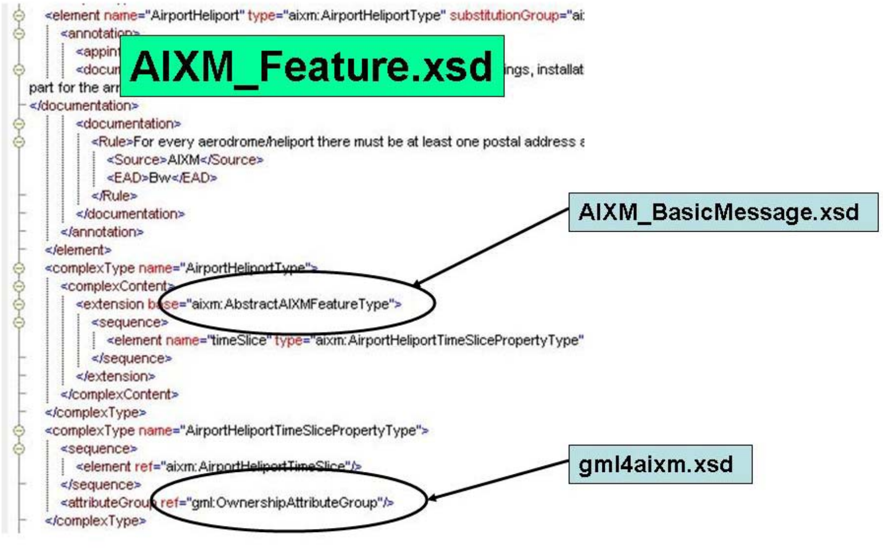


AIXM_DataTypes.xsd

```
      </restriction>
    </simpleType>
    <simpleTy
      <annota
        <doc
      </annotation>
      <restriction base="xsd:decimal">
        <pattern value="\d{1,8}(\.\d{1,4}){0,1}"/>
      </restriction>
    </simpleType>
    <complexType name="ValDepthType">
      <simpleContent>
        <annotation>
          <documentation/>
        </annotation>
        <extension base="aixm:ValDepthTypeBase">
          <attribute name="uom" type="aixm:UomDepthType" use="required"/>
        </extension>
      </simpleContent>
    </complexType>
    <simpleType name="CodeContaminationType">
      <annotation>
        <documentation/>
      </annotation>
```

XSDL Namespace

Locally in AIXM_DataTypes.xsd



AIXM_Feature.xsd

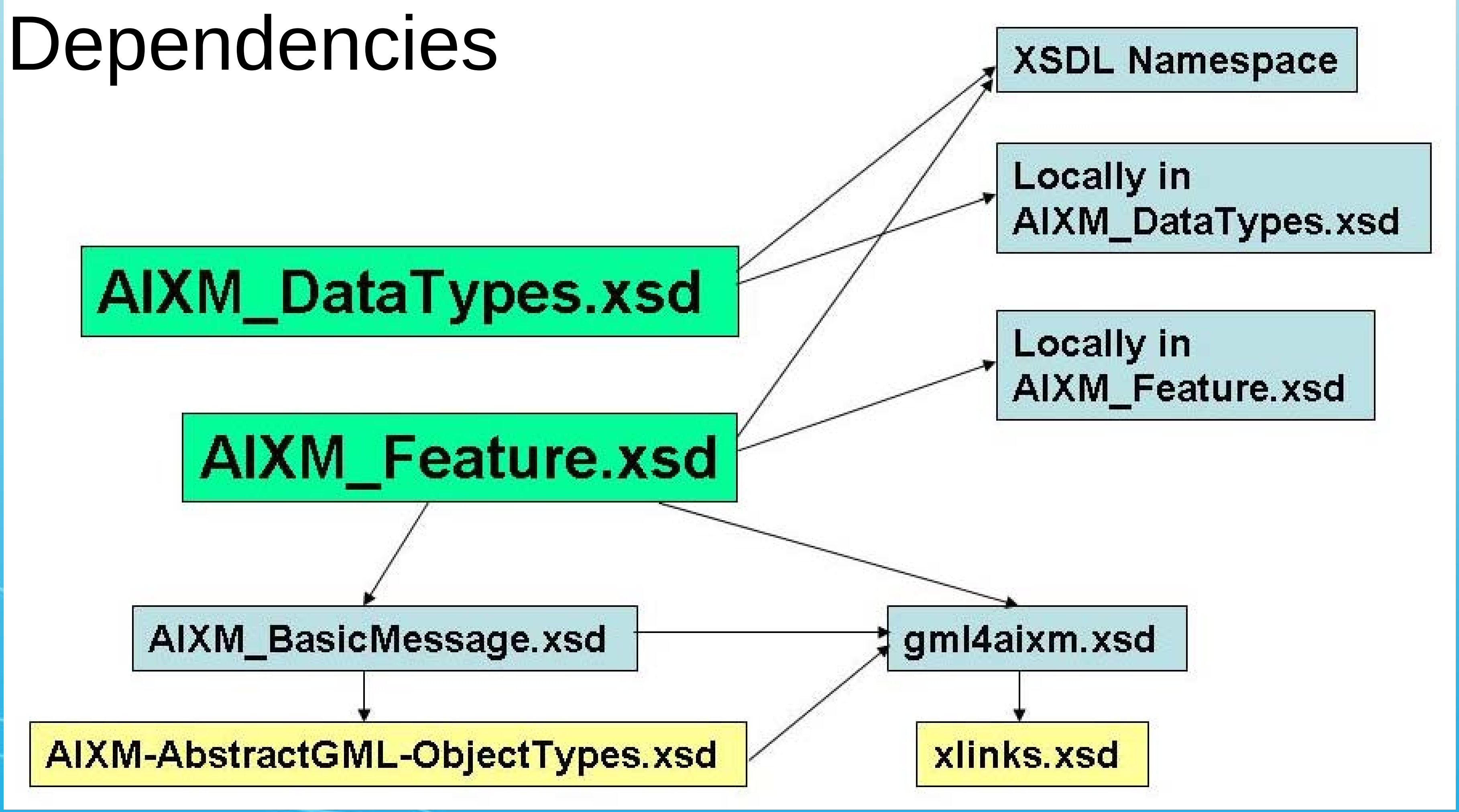
```
    <element name="AirportHeliport" type="aixm:AirportHeliportType" substitutionGroup="ai:
      <annotation>
        <appint
        <docum                                    ings, installat
part for the arr
</documentation>
        <documentation>
          <Rule>For every aerodrome/heliport there must be at least one postal address a
            <Source>AIXM</Source>
            <EAD>Bw</EAD>
          </Rule>
        </documentation>
      </annotation>
    </element>
    <complexType name="AirportHeliportType">
      <complexContent>
        <extension base="aixm:AbstractAIXMFeatureType">
          <sequence>
            <element name="timeSlice" type="aixm:AirportHeliportTimeSlicePropertyType"
          </sequence>
        </extension>
      </complexContent>
    </complexType>
    <complexType name="AirportHeliportTimeSlicePropertyType">
      <sequence>
        <element ref="aixm:AirportHeliportTimeSlice"/>
      </sequence>
      <attributeGroup ref="gml:OwnershipAttributeGroup"/>
    </complexType>
```

AIXM_BasicMessage.xsd

gml4aixm.xsd

- AIXM_DataTypes.xsd
- AIXM_Feature.xsd
- AIXM_BasicMessage.xsd
- AIXM-AbstractGML-ObjectTypes.xsd
- gml4aixm.xsd
- xlinks.xsd

AIXM 5 Release Candidate 2 (RC2) xsd files



XML Schema Definition Language (XSDL)

- Member of the XML family of standards
- XML schema is composed of pre-defined components:
  - Primary components:
    - Element declarations
    - Attribute declarations
    - Simple type definitions
    - Complex type definitions
  - Secondary components:
    - Attribute group definitions
    - Model group definitions
    - Identity-constraint definitions
    - Notation declarations
  - Dependent components:
    - Model groups
    - Annotations
    - Attribute uses
    - Particles and wildcards

Distinction between Global and Local components

- XSD components that are the children of the **schema** component are global components, and those lower in the schema hierarchy are local components:
  - <schema>
  - <global component>
  - <local component>
  - </local component>
  - </global component>
  - </schema>
- Global components can be reused many times within a schema by making references to these components, and local components can only be reused within the component for which they are defined.

Children and Character Data

- Some XML schema components can contain other components, referred to as children. The XML will read like:
  - <parent>
  - <child>
  - </child>
  - </parent>
- XML schema components may or may not contain character data, which are the definitions that are contained within a declaration or definition:
  - <component "character data">
  - </component>

XSDL Datatype Definitions

- Not to be confused with the <<Datatype>> class in UML
- Three types of XSDL datatypes in AIXM:
  - Datatype <<datatype>>
  - Datatype <<enumeration>>
  - Datatype <<codelist>>
- XSDL datatypes can be
  - Simple
  - Complex
- **A simple XSDL datatype can not have any XSDL attributes, and a complex XSDL datatype can.**
- **XSDL attributes are recommended for metadata (and as such are used in AIXM)**

What is an XSDL datatype?

- An XSDL datatype is defined as a three-tuple, and is composed of a value space, lexical space, and facets
  - Datatype = (value space, lexical space, facets)
- The *value space* of a datatype defines the set of values that are allowed for a datatype.
  - For example, the simple XSD datatype **string** has as value space the “set of finite length sequence of characters [XSD Spec Part 2, Section 3.2.1].”

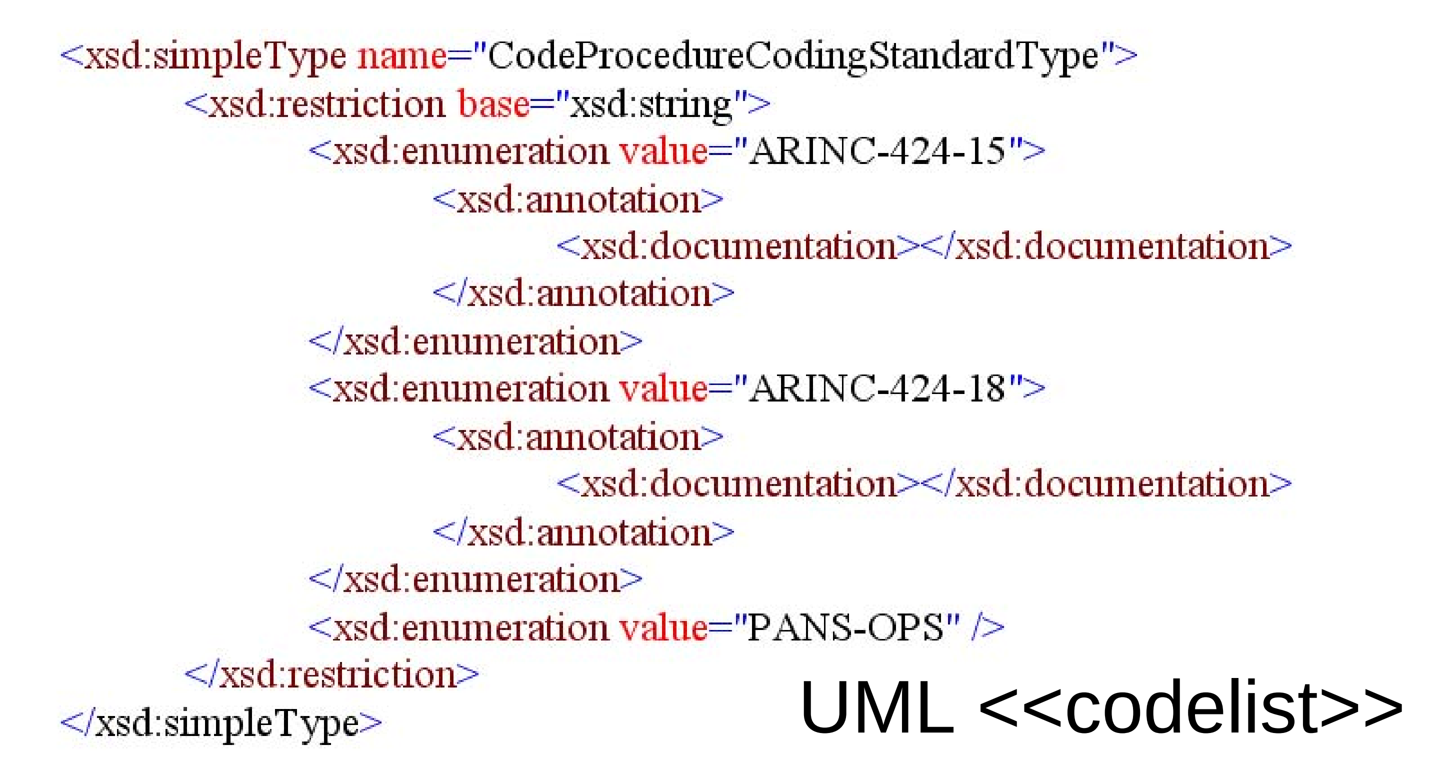
XSDL Datatype Lexical Space

- The *lexical space* of a datatype defines the “valid literals for a datatype [Part 2, Section 2.3].” A literal is a specific way in which a datatype is expressed.
  - For example, the simple datatype **time** may include an optional time zone indicator.
  - Time specified as 12:00 AM, or as 12:00 AM EST is said to be expressed as two types of literals
  - These literals are in the set of literals that define the lexical space of the simple datatype **time**.

XSDL Facets

- Datatypes *facets* are used to constrain the value space of an XSDL datatype. There are two types of facets:
  - Fundamental, such as applying the **numeric** facet to a datatype. Fundamental facets are not used in AIXM.
  - Constraining, such as applying the minimum and maximum integer value to the simple XSD datatype **byte** using the facets **minInclusive**

XSDL Simple Type Definitions : UML <<codelist>>



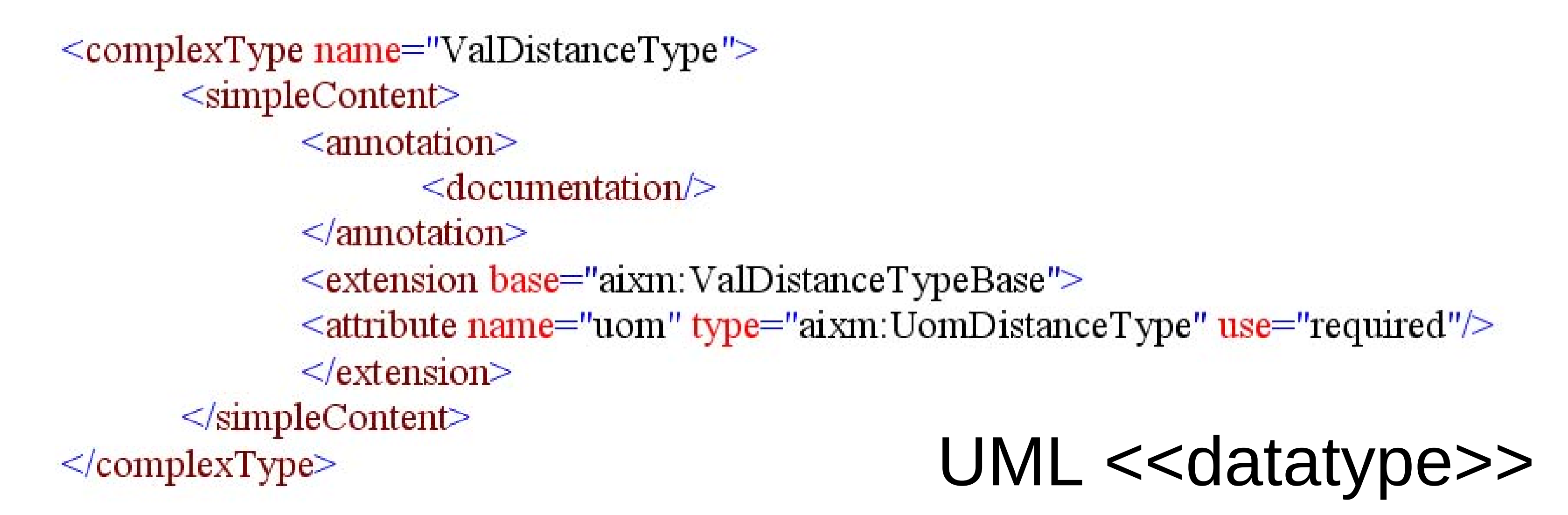
```
<xsd:simpleType name="CodeProcedureCodingStandardType">
        <xsd:restriction base="xsd:string">
                <xsd:enumeration value="ARINC-424-15">
                        <xsd:annotation>
                                <xsd:documentation></xsd:documentation>
                        </xsd:annotation>
                </xsd:enumeration>
                <xsd:enumeration value="ARINC-424-18">
                        <xsd:annotation>
                                <xsd:documentation></xsd:documentation>
                        </xsd:annotation>
                </xsd:enumeration>
                <xsd:enumeration value="PANS-OPS" />
        </xsd:restriction>
</xsd:simpleType>
```

UML <<codelist>>

- The example here from the AIXM_DataTypes.xsd file shows a simpleType XSDL datatype that has been restricted using an **enumeration** facet

XSDL Complex Type Definitions

- SimpleType and ComplexType XSDL datatypes can be extended in two ways:
  - By restricting the definition of the datatype, using the **restriction** tag.
  - By extending the definition of the datatype, using the **extension** tag.
- The resulting derived datatype from SimpleType and ComplexType XSDL datatypes is a ComplexType XSDL datatype if an attribute is added.



```
<complexType name="ValDistanceType">
        <simpleContent>
                <annotation>
                        <documentation/>
                </annotation>
                <extension base="aixm:ValDistanceTypeBase">
                <attribute name="uom" type="aixm:UomDistanceType" use="required"/>
                </extension>
        </simpleContent>
</complexType>
```

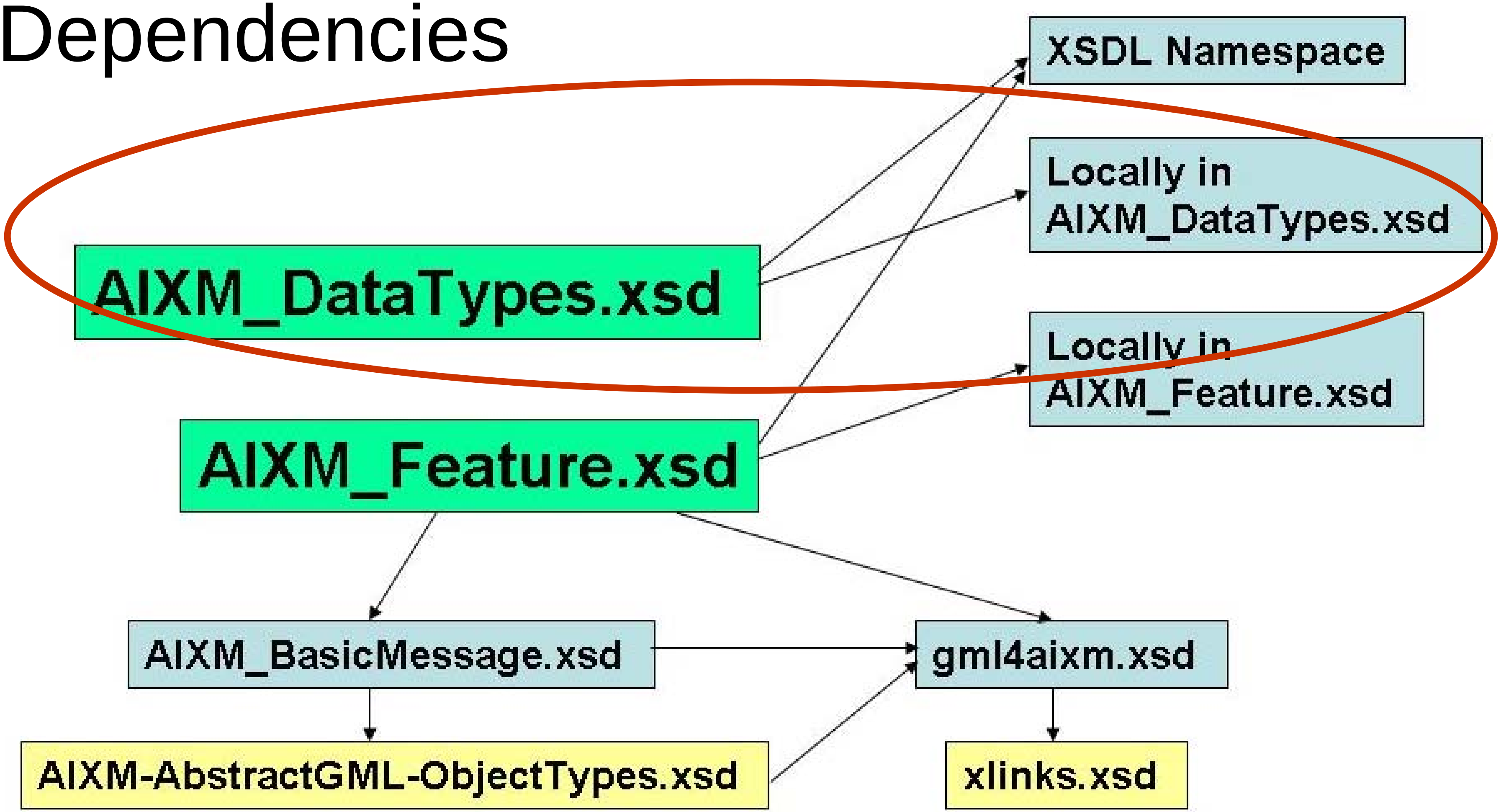
UML <<datatype>>

AIXM use of XSDL

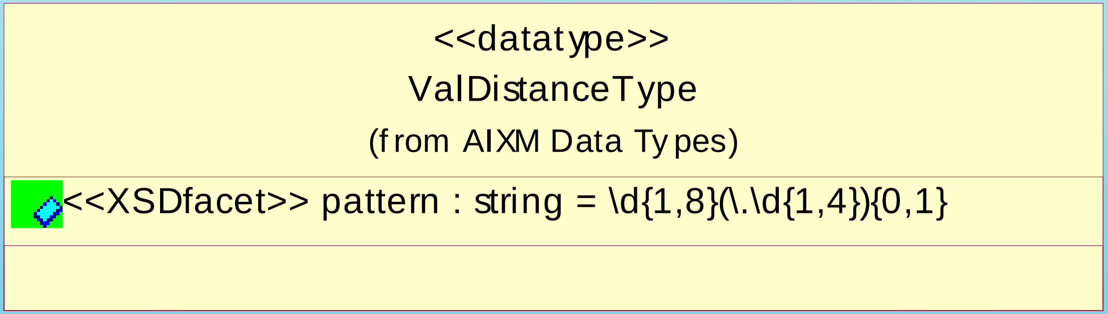
- Datatype definitions:
  - define AIXM features to encompass detailed aeronautical domain information accurately.
- Extension and restriction of datatypes:
  - define and reuse datatype definitions within the schema.
- Element and attribute declarations:
  - define AIXM features.
- Annotation and documentation:
  - to enable users to understand the contents of the AIXM XSD files.
- Additional schema components:
  - to incorporate the use of the Geographical Mark-up Language (GML), and to make use of additional concepts available in XSDL.

AIXM 5 Release Candidate 2 (RC2) xsd files

Dependencies



Associated Datatypes of Stereotype <<datatype>>



<<datatype>>
ValDistanceType
(from AIXM Data Types)

<<XSDfacet>> pattern : string = \d{1,8}(\.\d{1,4}){0,1}

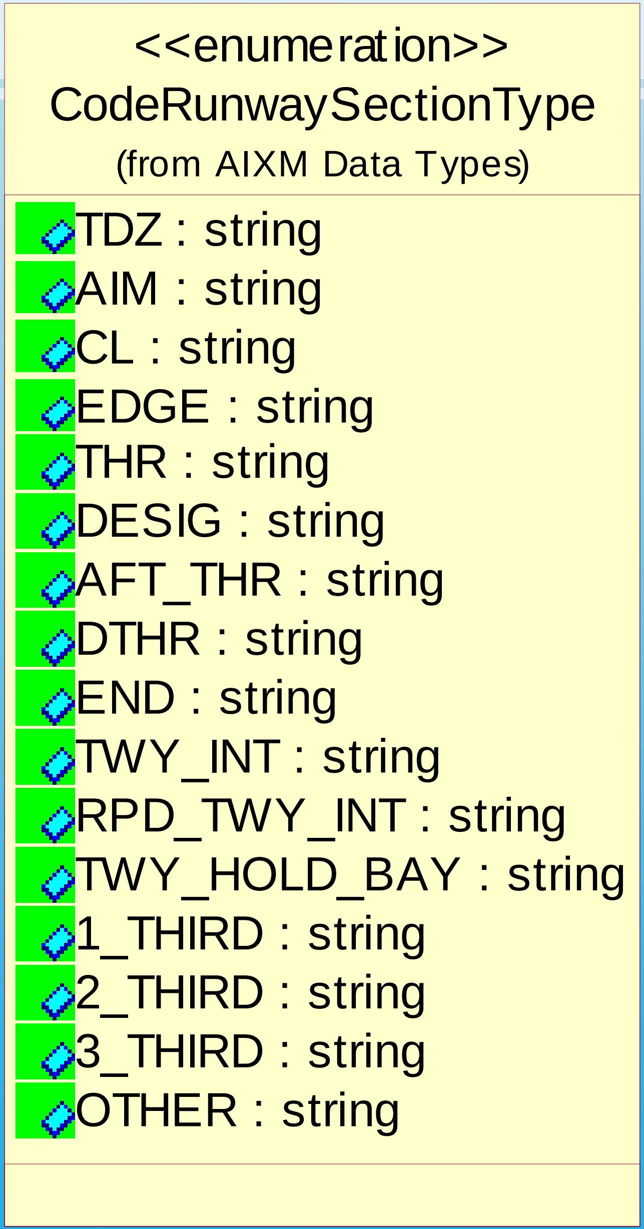
- Pattern
- Minimum Length
- Maximum Length
- ComplexType/SimpleContent
- Extends a SimpleType

ValDistanceType

```
<complexType name="ValDistanceType">
    <simpleContent>
          <annotation>
          <documentation/>
          </annotation>
          <extension base="aixm:ValDistanceTypeBase">
                    <attribute name="uom"
                    type="aixm:UomDistanceType"
                    use="required"/>
          </extension>
    </simpleContent>
</complexType>
```

Associated Datatypes of Stereotype <<enumeration>>

- This list cannot be expanded
- SimpleType
- Restricted from base type ‘string’
- Enumeration facet used to define UML attributes



<<enumeration>>
CodeRunwaySectionType
(from AIXM Data Types)

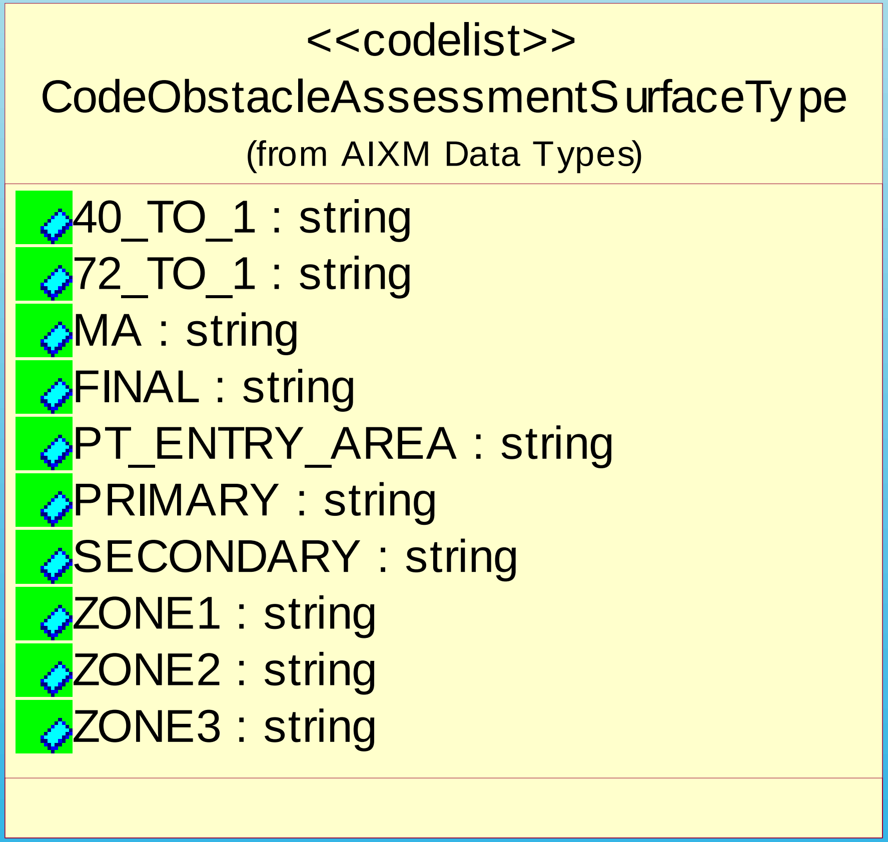
TDZ : string
AIM : string
CL : string
EDGE : string
THR : string
DESIG : string
AFT_THR : string
DTHR : string
END : string
TWY_INT : string
RPD_TWY_INT : string
TWY_HOLD_BAY : string
1_THIRD : string
2_THIRD : string
3_THIRD : string
OTHER : string

CodeRunwaySectionType

```
<simpleType name="CodeRunwaySectionType">
        <annotation>
        <documentation/>
        </annotation>
        <restriction base="xsd:string">
                <enumeration value="TDZ">
                        <annotation>
                        <documentation/>
                        </annotation>
                </enumeration>
                …additional UML attributes as enumeration
                facets
        </restriction>
</simpleType>
```

Associated Datatypes of Stereotype <<codelist>>

- The types of surfaces used for obstacle assessments form an open set of options
- The list can be expanded



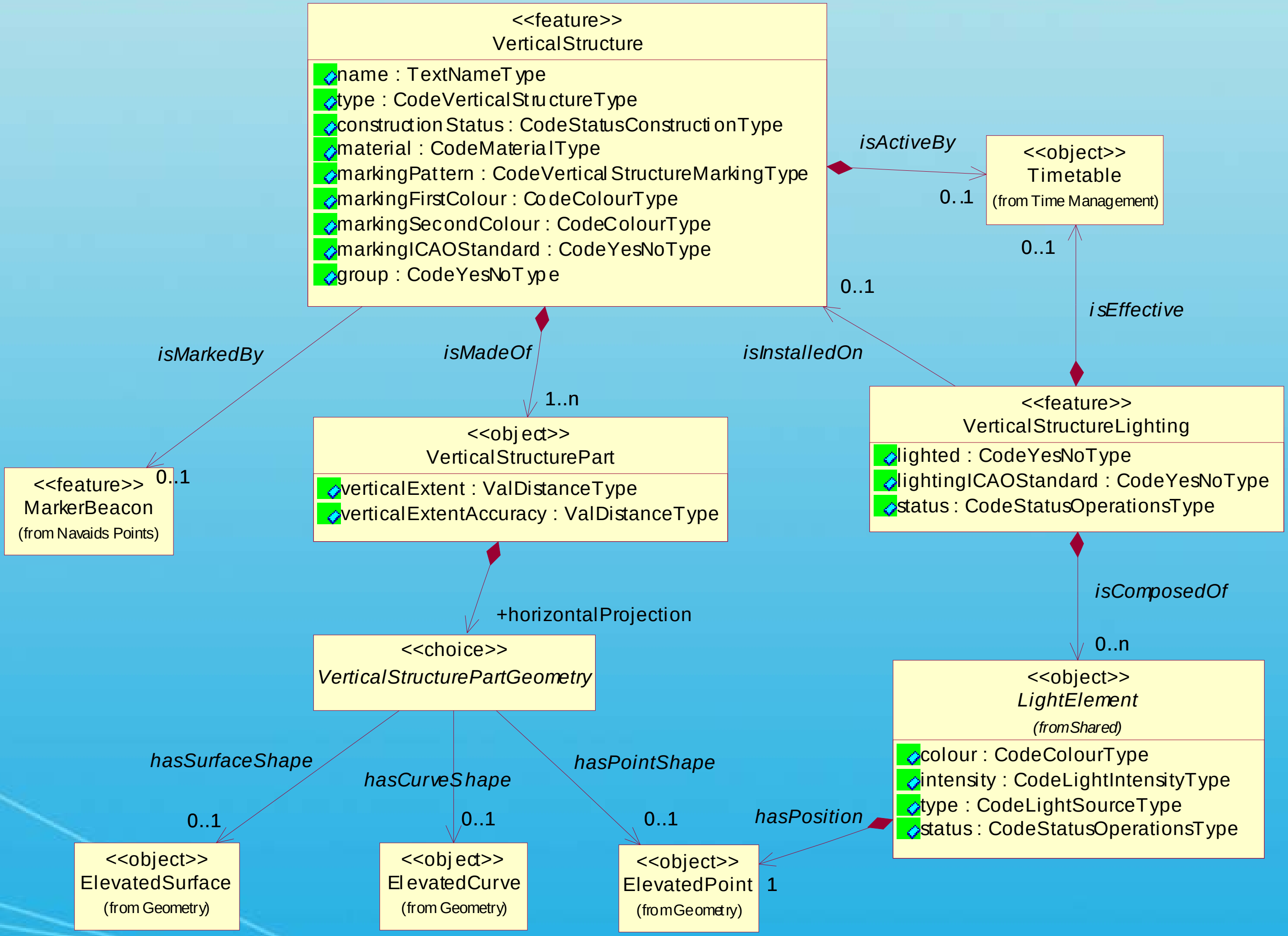
<<codelist>>
CodeObstacleAssessmentSurfaceType
(from AIXM Data Types)

40_TO_1 : string
72_TO_1 : string
MA : string
FINAL : string
PT_ENTRY_AREA : string
PRIMARY : string
SECONDARY : string
ZONE1 : string
ZONE2 : string
ZONE3 : string

CodeObstacleAssessmentSurfaceType

```
<simpleType name="CodeObstacleAssessmentSurfaceType">
   <union
   memberTypes="aixm:CodeObstacleAssessmentSurfaceType_base
   xsd:string"/>
</simpleType>
<simpleType name="CodeObstacleAssessmentSurfaceType_base">
      <annotation><documentation/></annotation>
      <restriction base="xsd:string">
            <enumeration value="40_TO_1">
            <annotation>
            <documentation/>
            </annotation>
            </enumeration>
            …additional UML attributes as enumeration
            facets
      </restriction>
</simpleType>
```

VerticalStructure includes Obstacles



<<feature>>
VerticalStructure
name : TextNameType
type : CodeVerticalStructureType
constructionStatus : CodeStatusConstructionType
material : CodeMaterialType
markingPattern : CodeVerticalStructureMarkingType
markingFirstColour : CodeColourType
markingSecondColour : CodeColourType
markingICAOStandard : CodeYesNoType
group : CodeYesNoType

isActiveBy
0..1
<<object>>
Timetable
(from Time Management)

0..1
isEffective

isMarkedBy
0..1
<<feature>>
MarkerBeacon
(from Navaids Points)

isMadeOf
1..n
<<object>>
VerticalStructurePart
verticalExtent : ValDistanceType
verticalExtentAccuracy : ValDistanceType

isInstalledOn
0..1
<<feature>>
VerticalStructureLighting
lighted : CodeYesNoType
lightingICAOStandard : CodeYesNoType
status : CodeStatusOperationsType

isComposedOf
0..n
<<object>>
LightElement
(from Shared)
colour : CodeColourType
intensity : CodeLightIntensityType
type : CodeLightSourceType
status : CodeStatusOperationsType

+horizontalProjection
<<choice>>
VerticalStructurePartGeometry

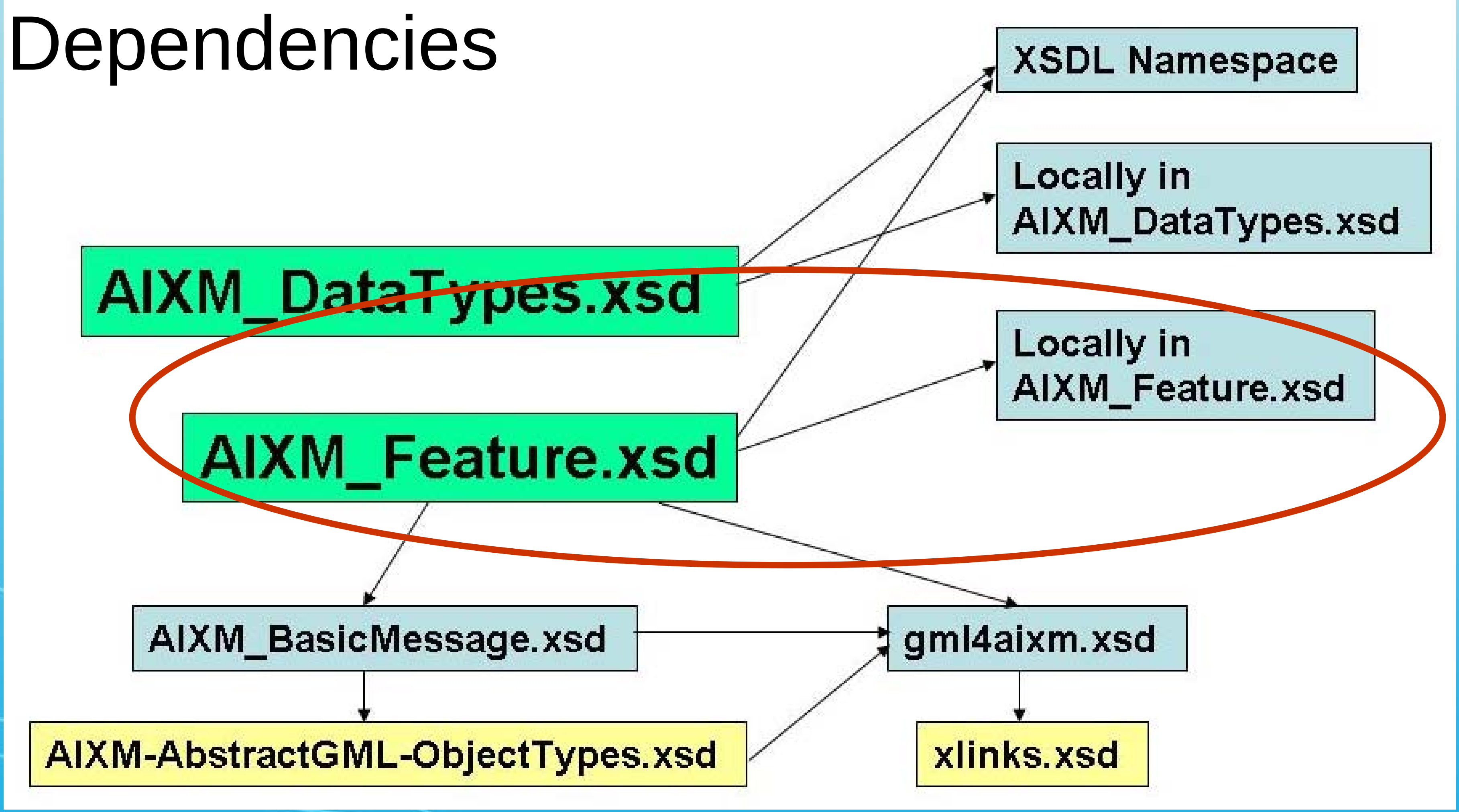
hasSurfaceShape
0..1
<<object>>
ElevatedSurface
(from Geometry)

hasCurveShape
0..1
<<object>>
ElevatedCurve
(from Geometry)

hasPointShape
0..1
<<object>>
ElevatedPoint
(from Geometry)

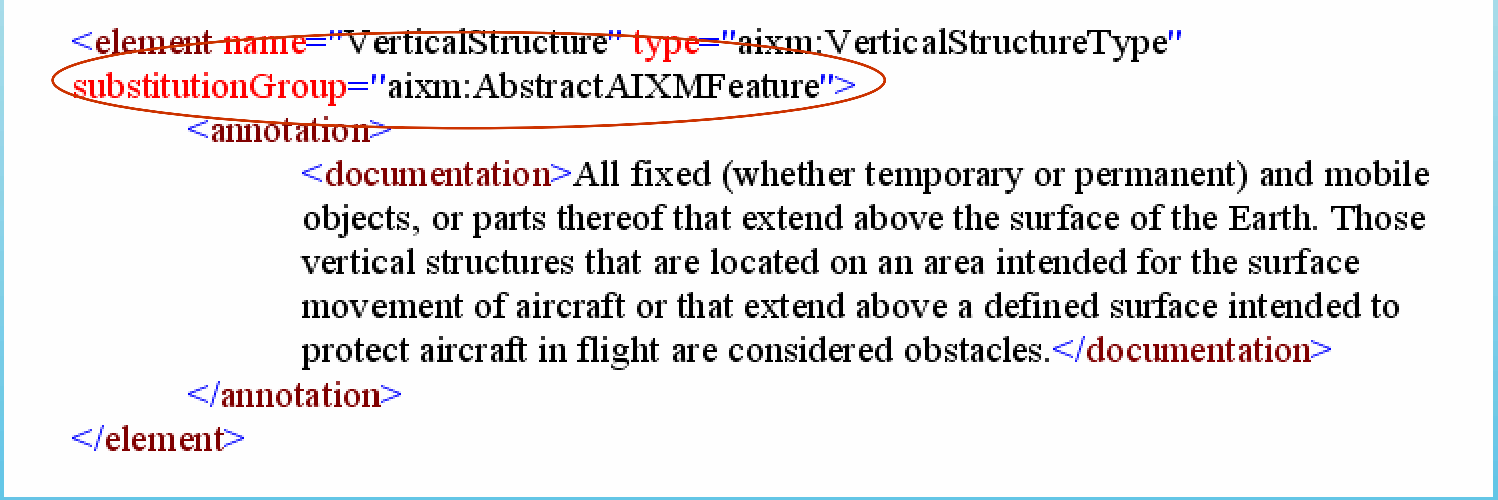
hasPosition
1

AIXM 5 Release Candidate 2 (RC2) xsd files



Dependencies

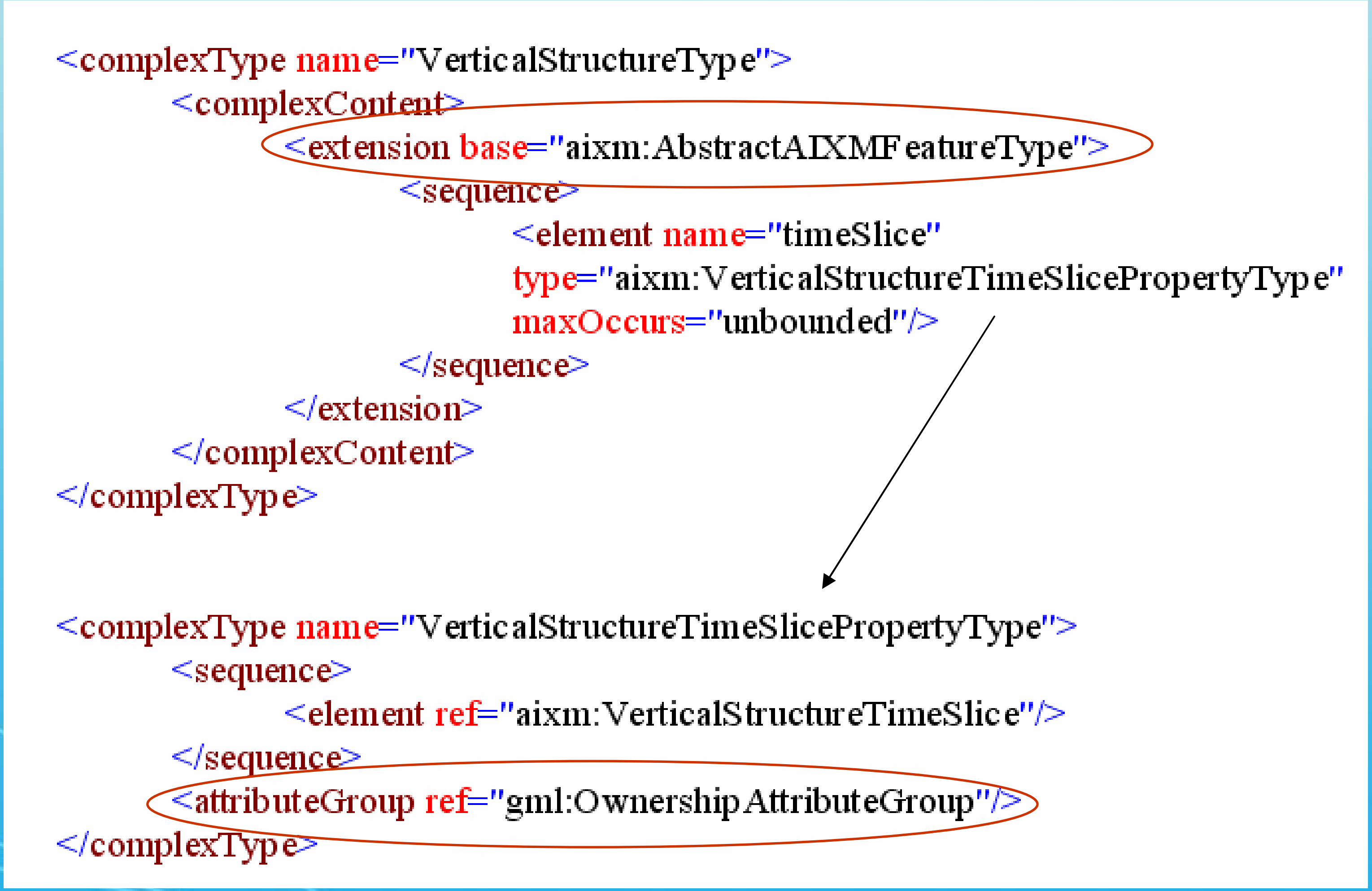
VerticalStructure element



```
<element name="VerticalStructure" type="aixm:VerticalStructureType"
substitutionGroup="aixm:AbstractAIXMFeature">
        <annotation>
                <documentation>All fixed (whether temporary or permanent) and mobile
                objects, or parts thereof that extend above the surface of the Earth. Those
                vertical structures that are located on an area intended for the surface
                movement of aircraft or that extend above a defined surface intended to
                protect aircraft in flight are considered obstacles.</documentation>
        </annotation>
</element>
```

- Parent is <schema>
  - A Global element
- Of type VerticalStructureType
- Inherits from the AbstractAIXMFeature
  - Defined in XML as a substitutionGroup

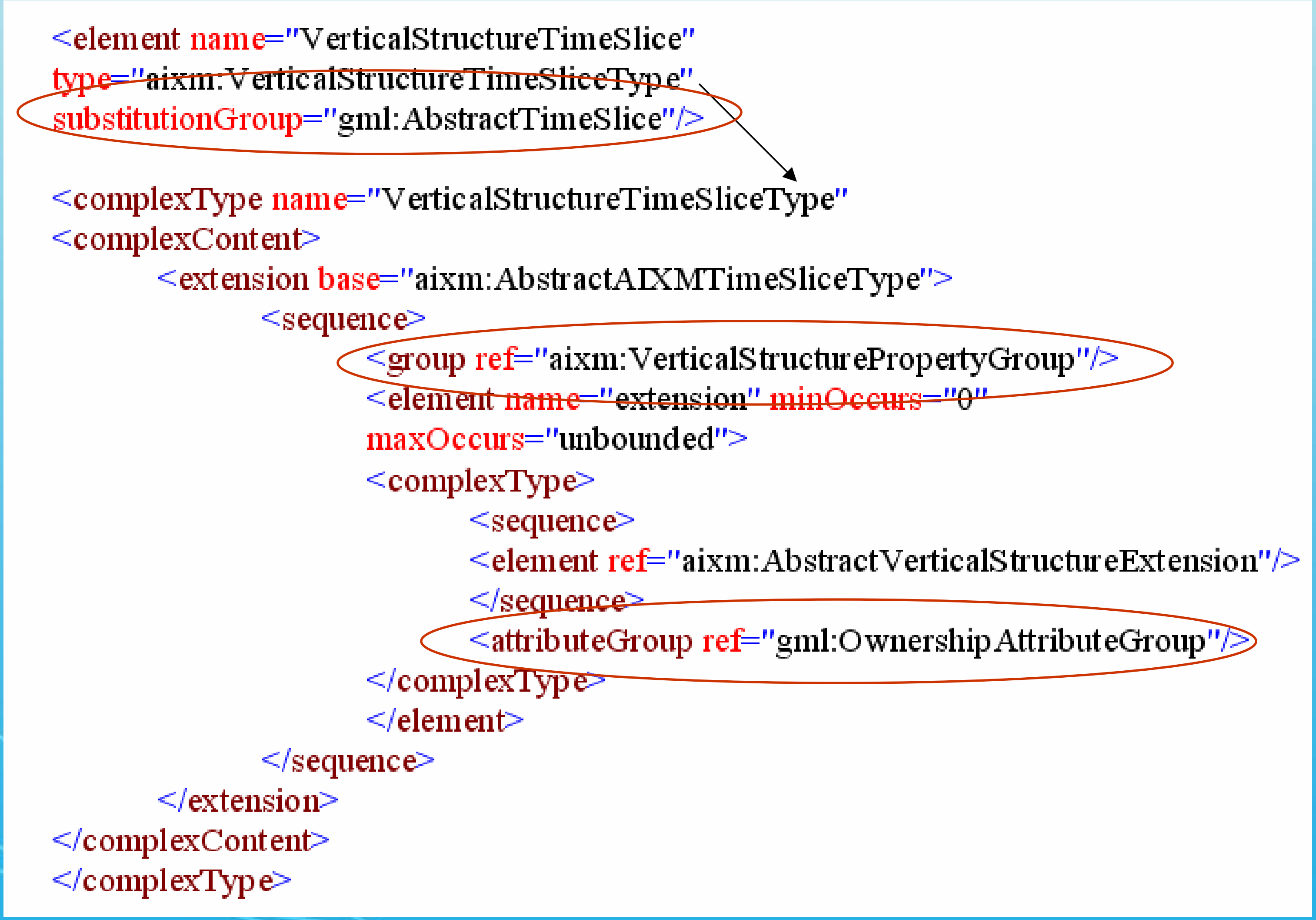
VerticalStructureType and reference to the TimeSlicePropertyType



```
<complexType name="VerticalStructureType">
        <complexContent>
                <extension base="aixm:AbstractAIXMFeatureType">
                        <sequence>
                                <element name="timeSlice"
                                type="aixm:VerticalStructureTimeSlicePropertyType"
                                maxOccurs="unbounded"/>
                        </sequence>
                </extension>
        </complexContent>
</complexType>


<complexType name="VerticalStructureTimeSlicePropertyType">
        <sequence>
                <element ref="aixm:VerticalStructureTimeSlice"/>
        </sequence>
        <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
```

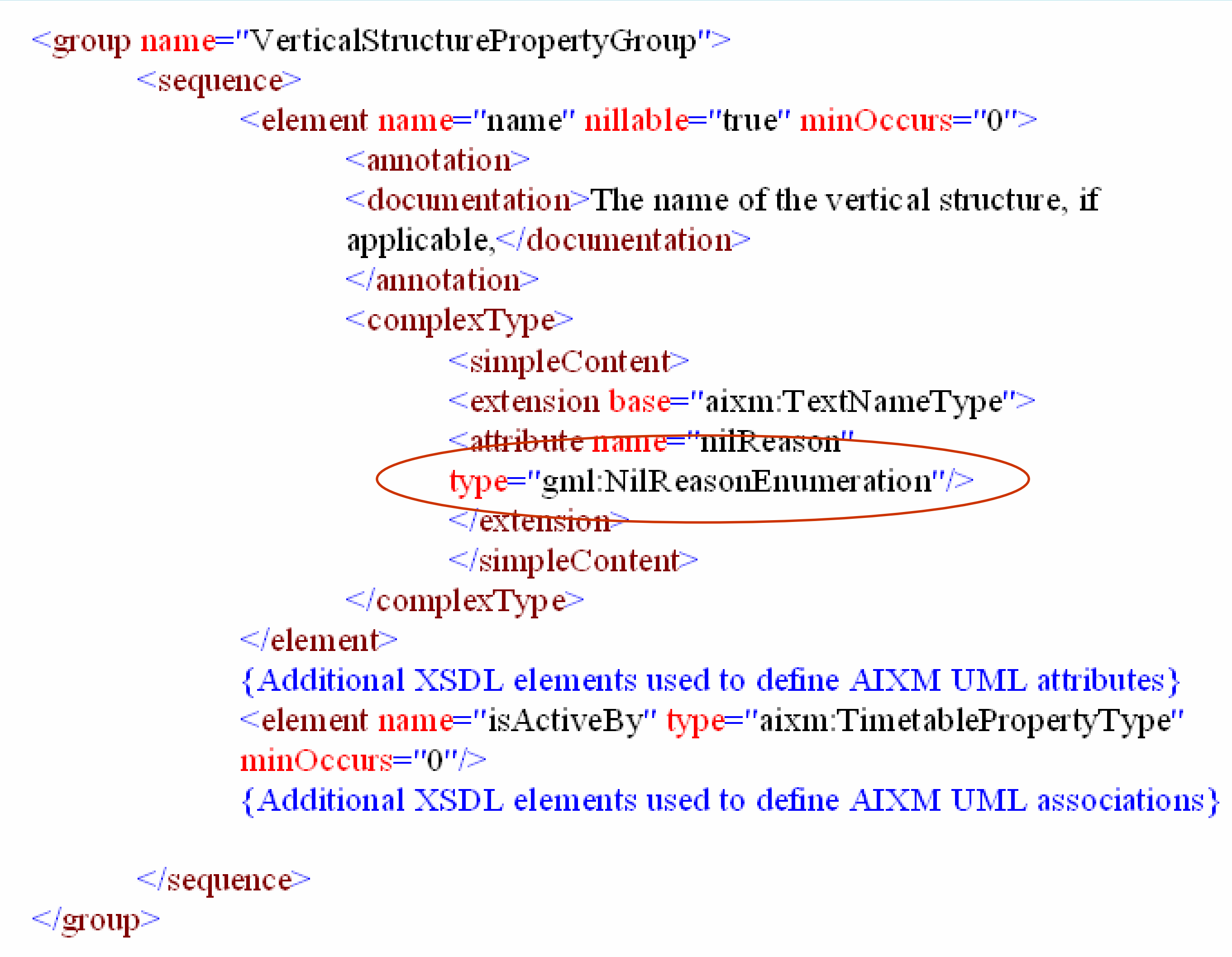
complexType definition for VerticalStructureTimeSliceType



```
<element name="VerticalStructureTimeSlice"
type="aixm:VerticalStructureTimeSliceType"
substitutionGroup="gml:AbstractTimeSlice"/>

<complexType name="VerticalStructureTimeSliceType"
<complexContent>
        <extension base="aixm:AbstractAIXMTimeSliceType">
                <sequence>
                        <group ref="aixm:VerticalStructurePropertyGroup"/>
                        <element name="extension" minOccurs="0"
                        maxOccurs="unbounded">
                        <complexType>
                                <sequence>
                                <element ref="aixm:AbstractVerticalStructureExtension"/>
                                </sequence>
                                <attributeGroup ref="gml:OwnershipAttributeGroup"/>
                        </complexType>
                        </element>
                </sequence>
        </extension>
</complexContent>
</complexType>
```

How XSDL is used to encode AIXM UML attributes VerticalStructurePropertyGroup

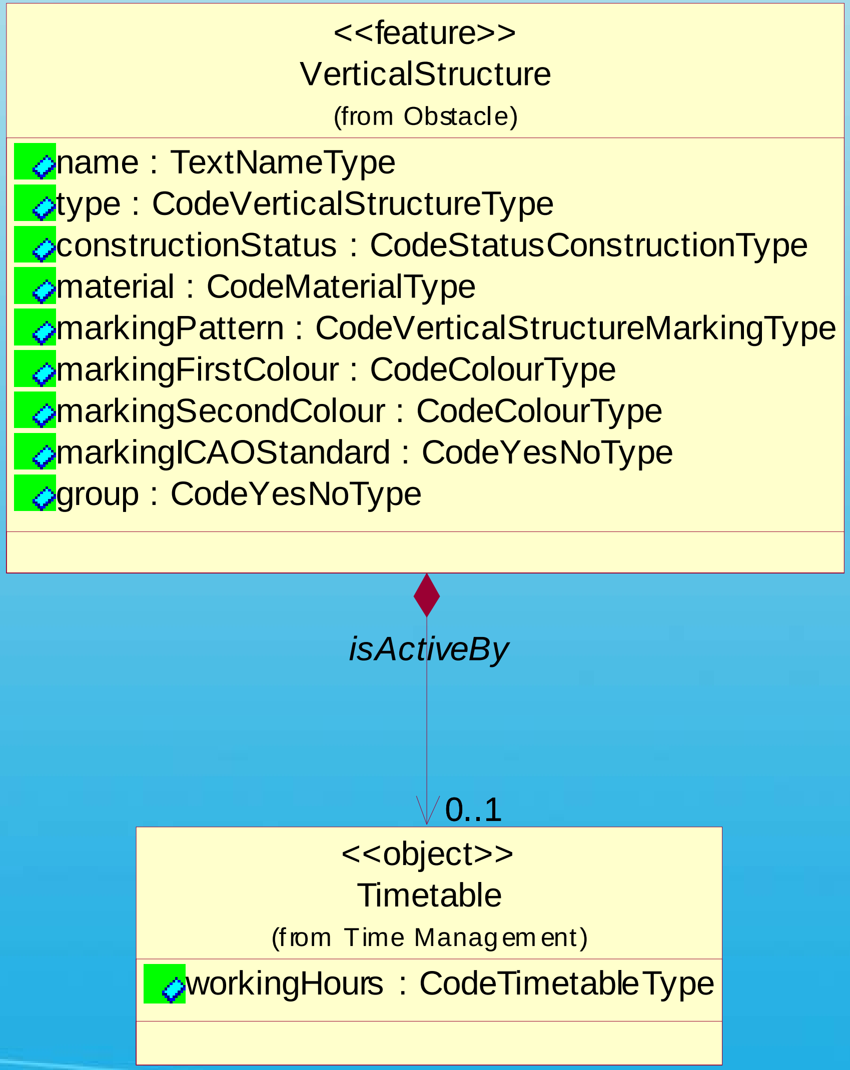


```
<group name="VerticalStructurePropertyGroup">
      <sequence>
            <element name="name" nillable="true" minOccurs="0">
                  <annotation>
                  <documentation>The name of the vertical structure, if
                  applicable,</documentation>
                  </annotation>
                  <complexType>
                        <simpleContent>
                        <extension base="aixm:TextNameType">
                        <attribute name="nilReason"
                        type="gml:NilReasonEnumeration"/>
                        </extension>
                        </simpleContent>
                  </complexType>
            </element>
            {Additional XSDL elements used to define AIXM UML attributes}
            <element name="isActiveBy" type="aixm:TimetablePropertyType"
            minOccurs="0"/>
            {Additional XSDL elements used to define AIXM UML associations}

      </sequence>
</group>
```

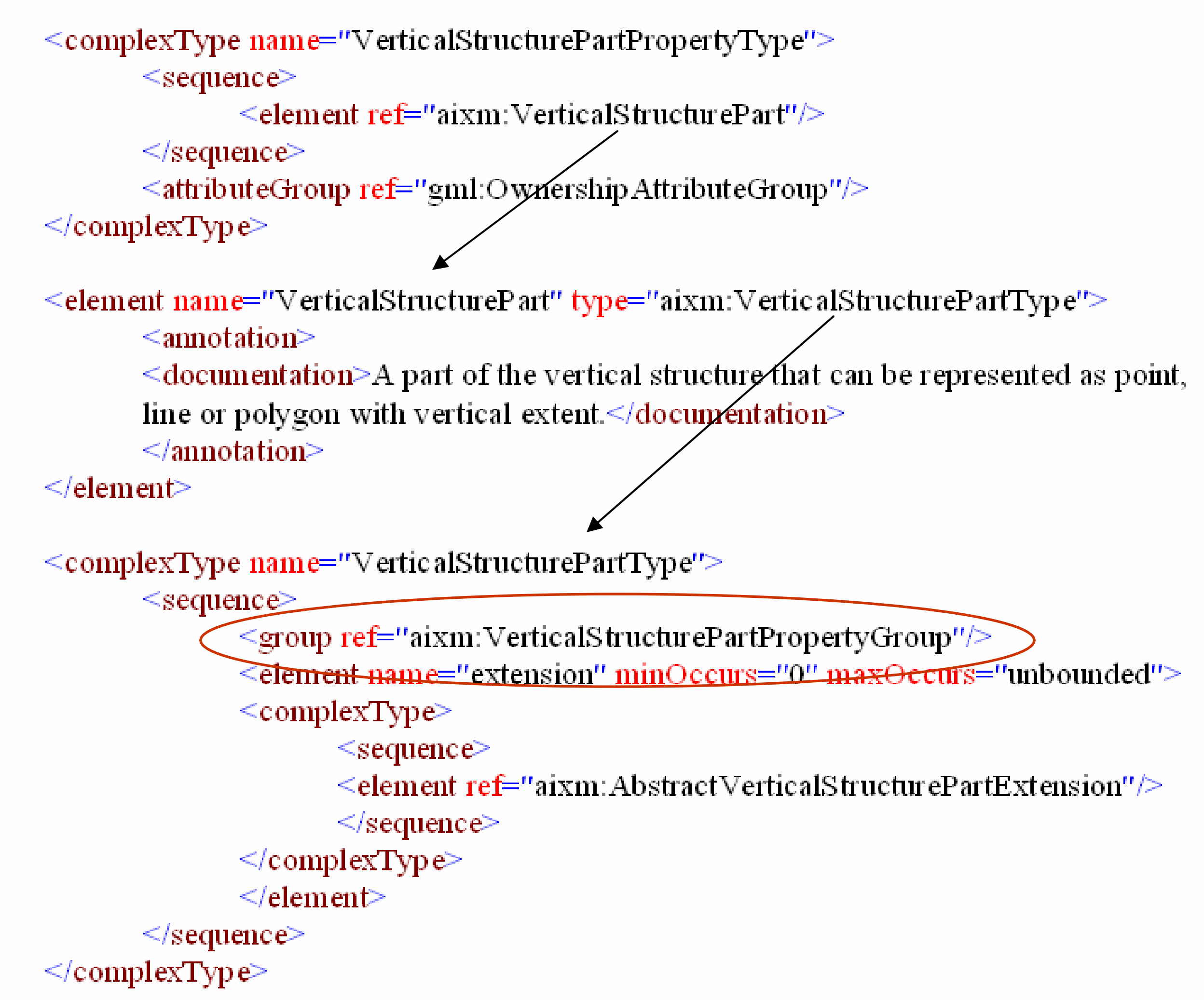
VerticalStructure UML feature

- The VerticalStructure feature is an XSDL element
- Its UML attributes are elements in a named model group
- The UML attribute UML datatypes are XSDL simpleType and complexType datatypes
- The UML associations are XSDL elements



<<feature>>
VerticalStructure
(from Obstacle)
name : TextNameType
type : CodeVerticalStructureType
constructionStatus : CodeStatusConstructionType
material : CodeMaterialType
markingPattern : CodeVerticalStructureMarkingType
markingFirstColour : CodeColourType
markingSecondColour : CodeColourType
markingICAOStandard : CodeYesNoType
group : CodeYesNoType
isActiveBy
0..1
<<object>>
Timetable
(from Time Management)
workingHours : CodeTimetableType

How XSDL is used to encode AIXM UML choice classes VerticalStructurePartPropertyGroup

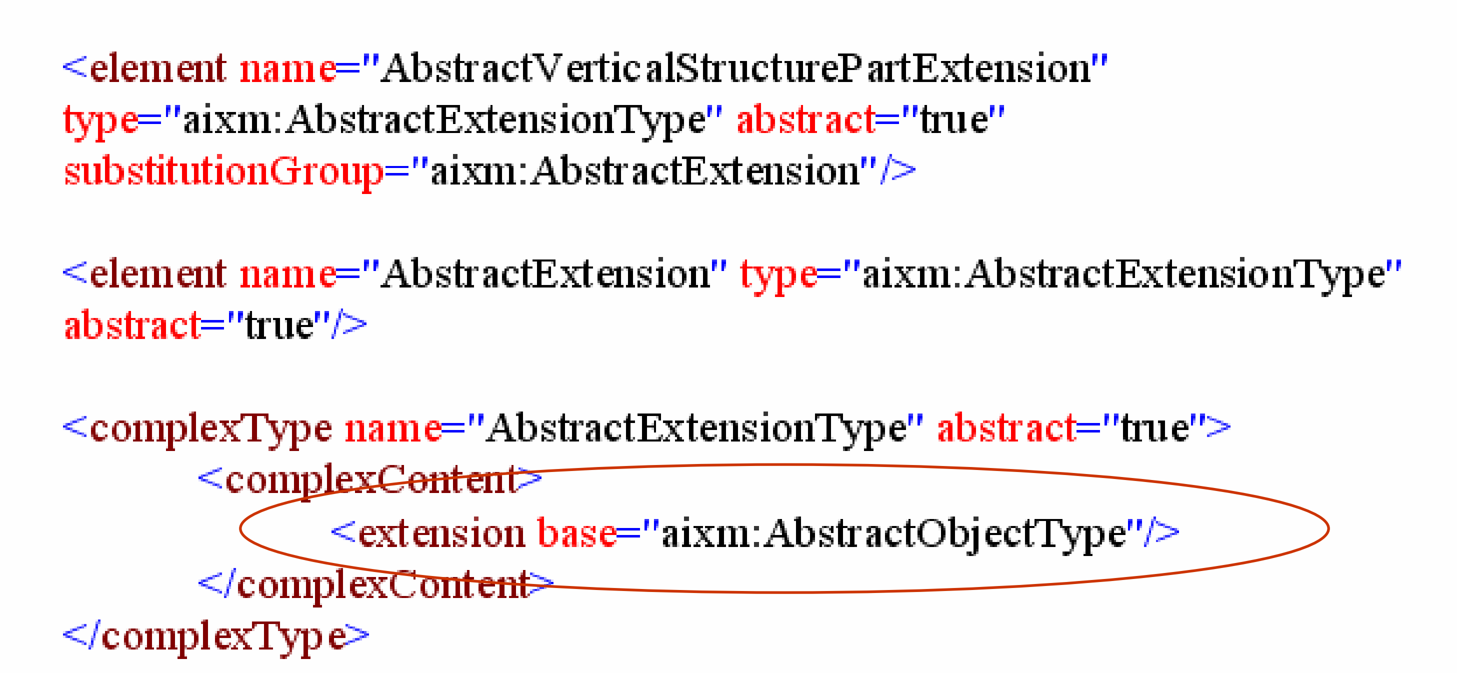


```
<complexType name="VerticalStructurePartPropertyType">
        <sequence>
                <element ref="aixm:VerticalStructurePart"/>
        </sequence>
        <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="VerticalStructurePart" type="aixm:VerticalStructurePartType">
        <annotation>
        <documentation>A part of the vertical structure that can be represented as point,
        line or polygon with vertical extent.</documentation>
        </annotation>
</element>

<complexType name="VerticalStructurePartType">
        <sequence>
                <group ref="aixm:VerticalStructurePartPropertyGroup"/>
                <element name="extension" minOccurs="0" maxOccurs="unbounded">
                <complexType>
                        <sequence>
                        <element ref="aixm:AbstractVerticalStructurePartExtension"/>
                        </sequence>
                </complexType>
                </element>
        </sequence>
</complexType>
```

How XSDL is used to encode AIXM UML choice classes VerticalStructurePartPropertyGroup

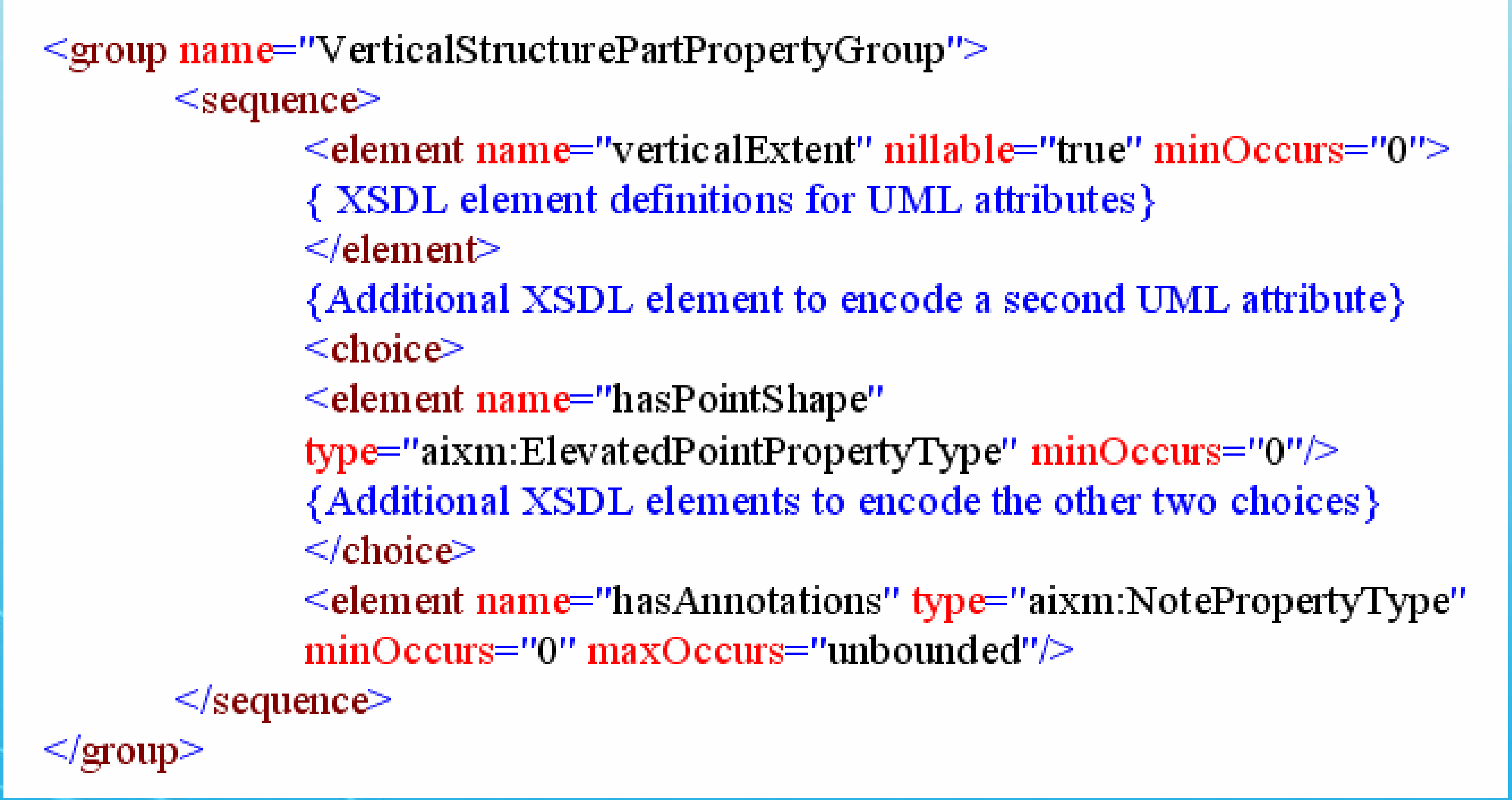


```
<element name="AbstractVerticalStructurePartExtension"
type="aixm:AbstractExtensionType" abstract="true"
substitutionGroup="aixm:AbstractExtension"/>

<element name="AbstractExtension" type="aixm:AbstractExtensionType"
abstract="true"/>

<complexType name="AbstractExtensionType" abstract="true">
        <complexContent>
                <extension base="aixm:AbstractObjectType"/>
        </complexContent>
</complexType>
```

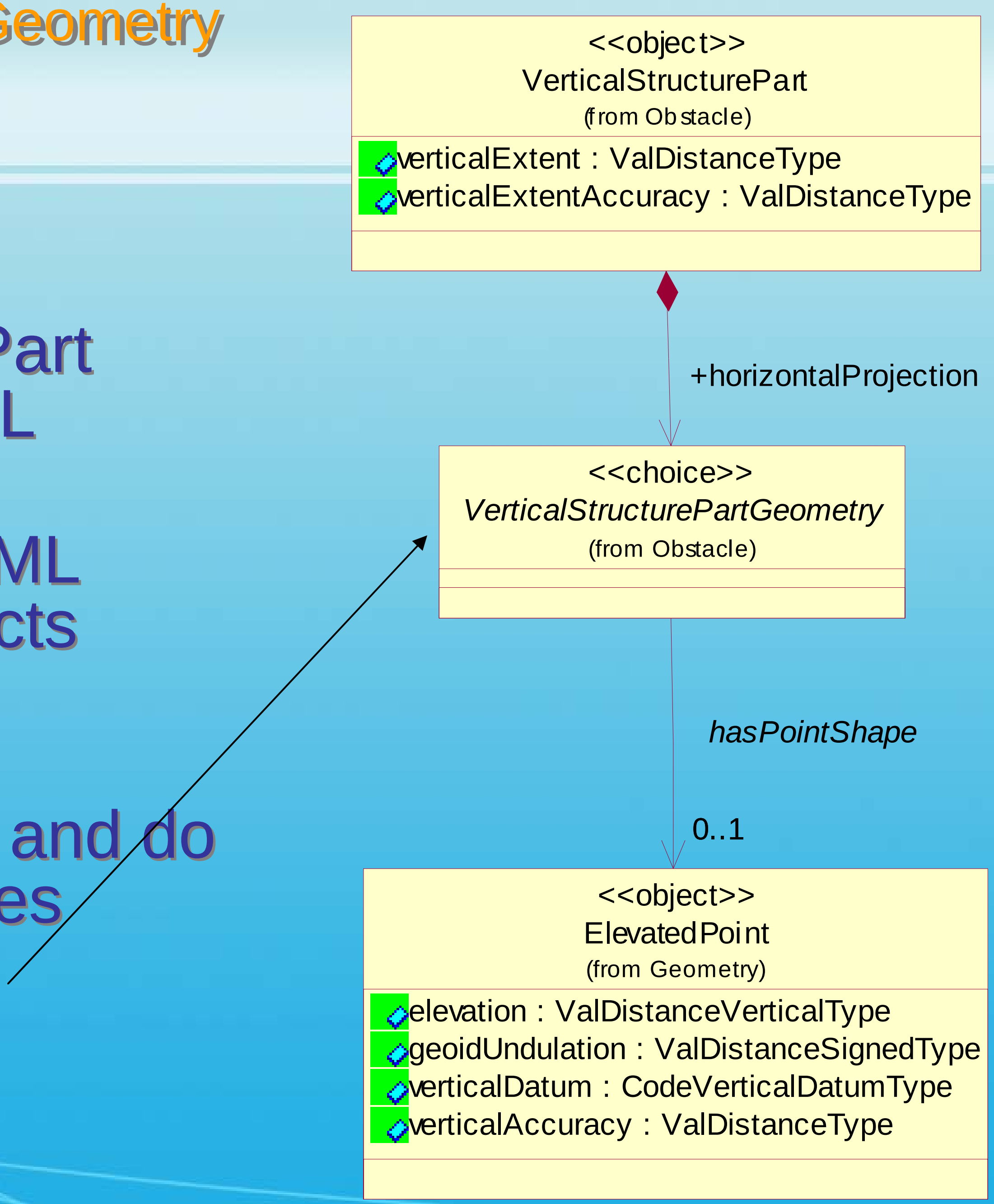
How XSDL is used to encode AIXM UML choice classes VerticalStructurePart



```
<group name="VerticalStructurePartPropertyGroup">
        <sequence>
                <element name="verticalExtent" nillable="true" minOccurs="0">
                { XSDL element definitions for UML attributes}
                </element>
                {Additional XSDL element to encode a second UML attribute}
                <choice>
                <element name="hasPointShape"
                type="aixm:ElevatedPointPropertyType" minOccurs="0"/>
                {Additional XSDL elements to encode the other two choices}
                </choice>
                <element name="hasAnnotations" type="aixm:NotePropertyType"
                minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
</group>
```

VerticalStructurePartGeometry
UML choice

- The VerticalStructurePart feature is an XSDL element
- XSDL encodes UML features and objects using the same components
- Objects are static and do not have time slices
- UML choice is an abstract class



<<object>>
VerticalStructurePart
(from Obstacle)
verticalExtent : ValDistanceType
verticalExtentAccuracy : ValDistanceType
+horizontalProjection
<<choice>>
VerticalStructurePartGeometry
(from Obstacle)
hasPointShape
0..1
<<object>>
ElevatedPoint
(from Geometry)
elevation : ValDistanceVerticalType
geoidUndulation : ValDistanceSignedType
verticalDatum : CodeVerticalDatumType
verticalAccuracy : ValDistanceType

AIXM
Aeronautical Information Exchange Model

Summary

- Described XSDL for
  - Representative AIXM UML datatypes
  - Representative AIXM UML classes
    - Feature (dynamic, use time slices)
    - Choice (UML abstract class)
    - Object (static, do not use time slices)