AIXM 5 Technologies

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

AIXM Class | 2007

AIXM 5 RC2

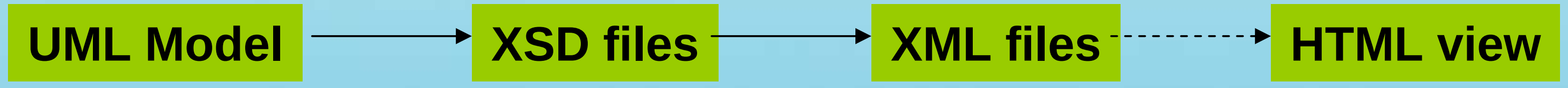
EUROCONTROL

FEDERAL AVIATION ADMINISTRATION

Topics

- UML view of AIXM
  - Class diagrams
    - Datatype stereotypes
    - Attributes
    - Datatype definitions
- XSD view of AIXM
  - XSD specification
  - UML to XSD
- GML
  - Object-property model

General Scheme


UML Model

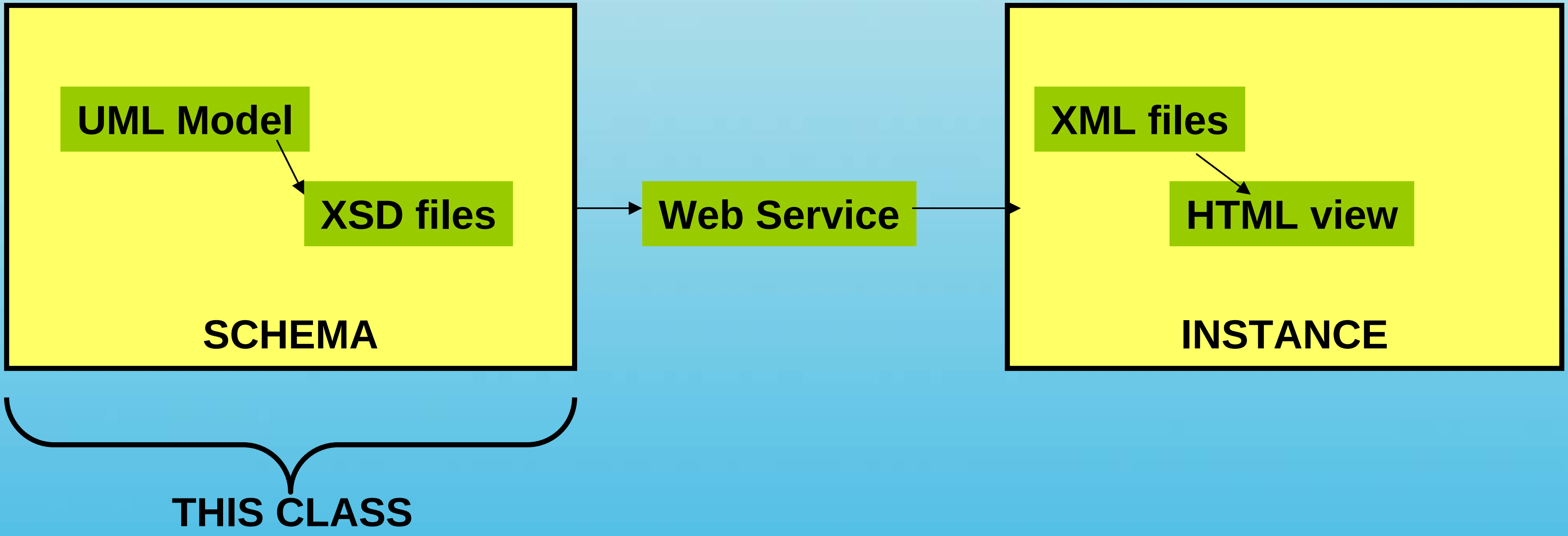
XSD files

XML files

HTML view

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

Scope of this class



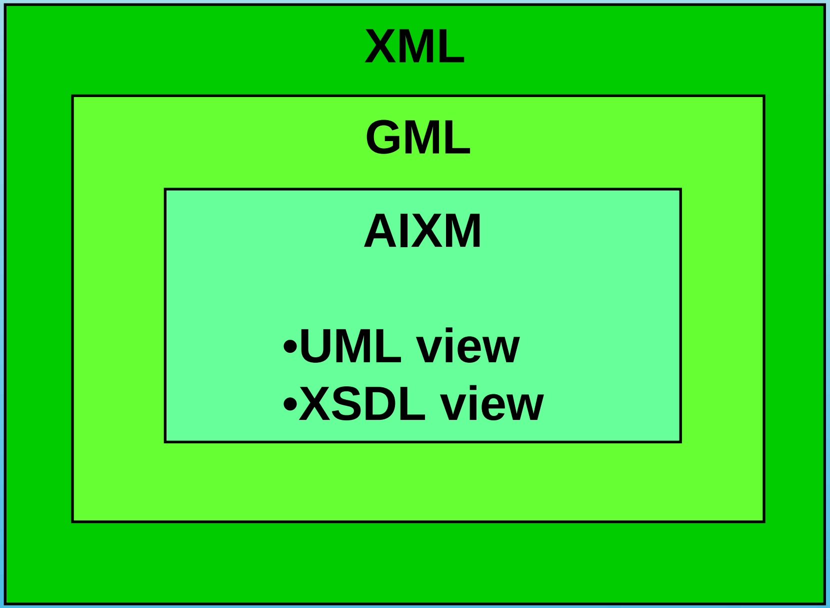
- AIXM, as defined using XSD and GML is a schema definition
- Java, XSLT, and other technologies can be used to transform the content as defined in this schema into data instances

Schema and schema instance

- AIXM defines a schema for aeronautical information using the XML Schema definition language, XSDL
- GML is also defined using XSDL
- AIXM XSD is a subset of GML, and we can say AIXM is GML
- A schema instance is an XML file that is generated using the schema
- An XML file using features from the AIXM/GML schema is an instance of the schema

AIXM technologies

- UML view used for conceptual design of AIXM
- GML is the core markup language used to define AIXM
- XSDL from UML generated for implementations of AIXM



XML

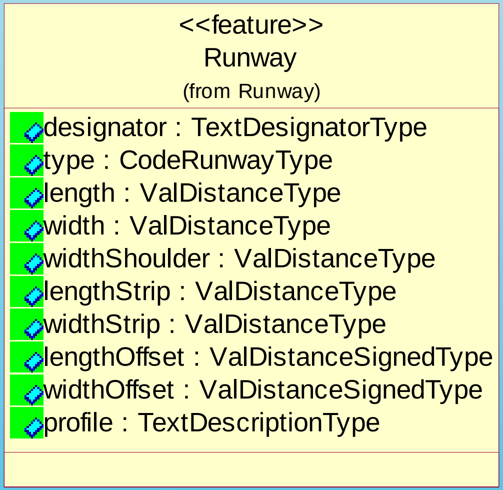
GML

AIXM

•UML view
•XSDL view

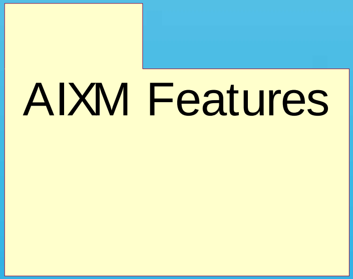
Diagrams

- Model uses:
  - Class diagrams
    - features, properties, relationships and inheritance between features
  - Package diagrams
    - split the model into modules and identify dependencies among sets of classes.



<<feature>>
Runway
(from Runway)

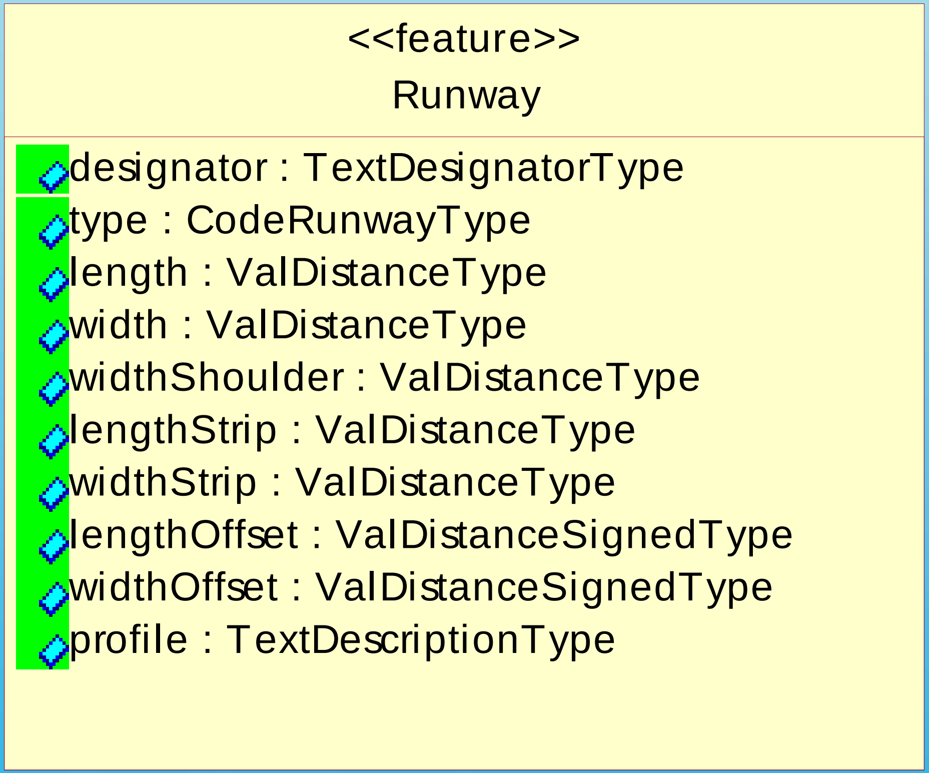
designator : TextDesignatorType
type : CodeRunwayType
length : ValDistanceType
width : ValDistanceType
widthShoulder : ValDistanceType
lengthStrip : ValDistanceType
widthStrip : ValDistanceType
lengthOffset : ValDistanceSignedType
widthOffset : ValDistanceSignedType
profile : TextDescriptionType



AIXM Features

UML for AIXM

- Class diagrams
  - Three sections
  - Top
    - datatype stereotype
    - name of class (Runway)
    - Reference to external class package, if needed
  - Middle
    - Attributes
  - Bottom
    - Methods (left empty in AIXM to allow implementers freedom of using methods within their applications)



<<feature>>
Runway

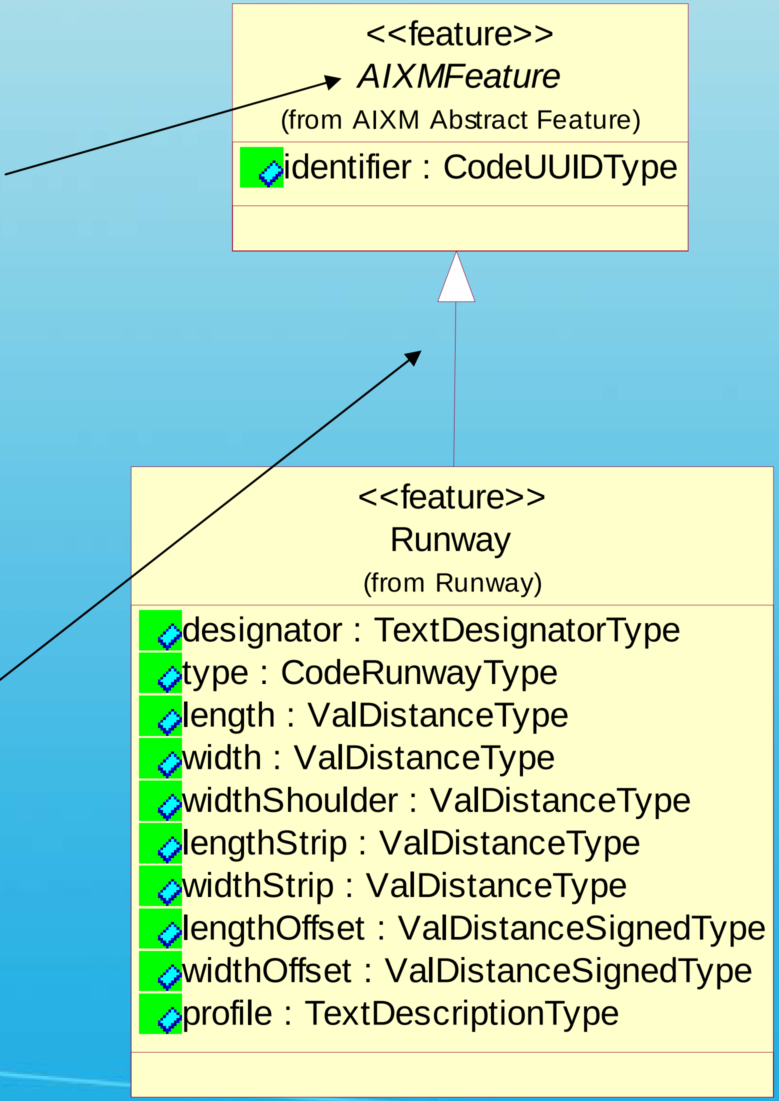
designator : TextDesignatorType
type : CodeRunwayType
length : ValDistanceType
width : ValDistanceType
widthShoulder : ValDistanceType
lengthStrip : ValDistanceType
widthStrip : ValDistanceType
lengthOffset : ValDistanceSignedType
widthOffset : ValDistanceSignedType
profile : TextDescriptionType

What is abstraction, a.k.a inheritance?

- Describes “is type of” relationships
  - A car is a type of vehicle
  - A bicycle is a type of vehicle
- A VOR is a type of navigation equipment
- A DME is a type of navigation equipment
- Characteristics common to both can be abstracted into a navigation equipment feature from which VOR and DME inherit their common characteristics
- Inheritance - The specialized feature inherits all of the properties of the more general feature

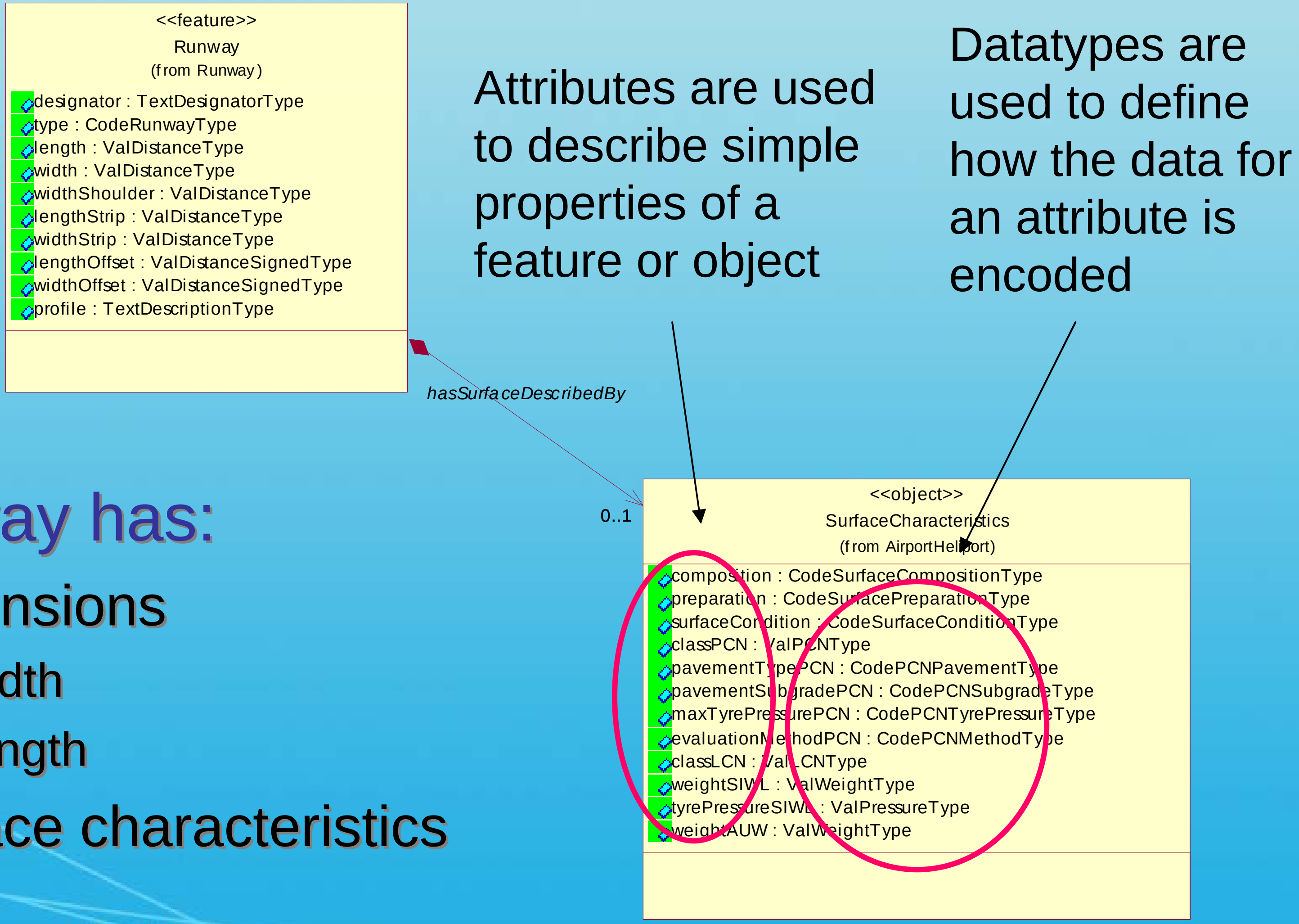
Abstract Classes

- Abstract classes are designated by putting the class name in *italics*.
- An abstract class cannot be realised in an implementation such as an XML document
- Abstract classes are used as base classes in an inheritance hierarchy.



<<feature>>
AIXMFeature
(from AIXM Abstract Feature)
identifier : CodeUUIDType
<<feature>>
Runway
(from Runway)
designator : TextDesignatorType
type : CodeRunwayType
length : ValDistanceType
width : ValDistanceType
widthShoulder : ValDistanceType
lengthStrip : ValDistanceType
widthStrip : ValDistanceType
lengthOffset : ValDistanceSignedType
widthOffset : ValDistanceSignedType
profile : TextDescriptionType

Runway



<<feature>>
Runway
(from Runway)

- designator : TextDesignatorType
- type : CodeRunwayType
- length : ValDistanceType
- width : ValDistanceType
- widthShoulder : ValDistanceType
- lengthStrip : ValDistanceType
- widthStrip : ValDistanceType
- lengthOffset : ValDistanceSignedType
- widthOffset : ValDistanceSignedType
- profile : TextDescriptionType

hasSurfaceDescribedBy

0..1

<<object>>
SurfaceCharacteristics
(from AirportHeliport)

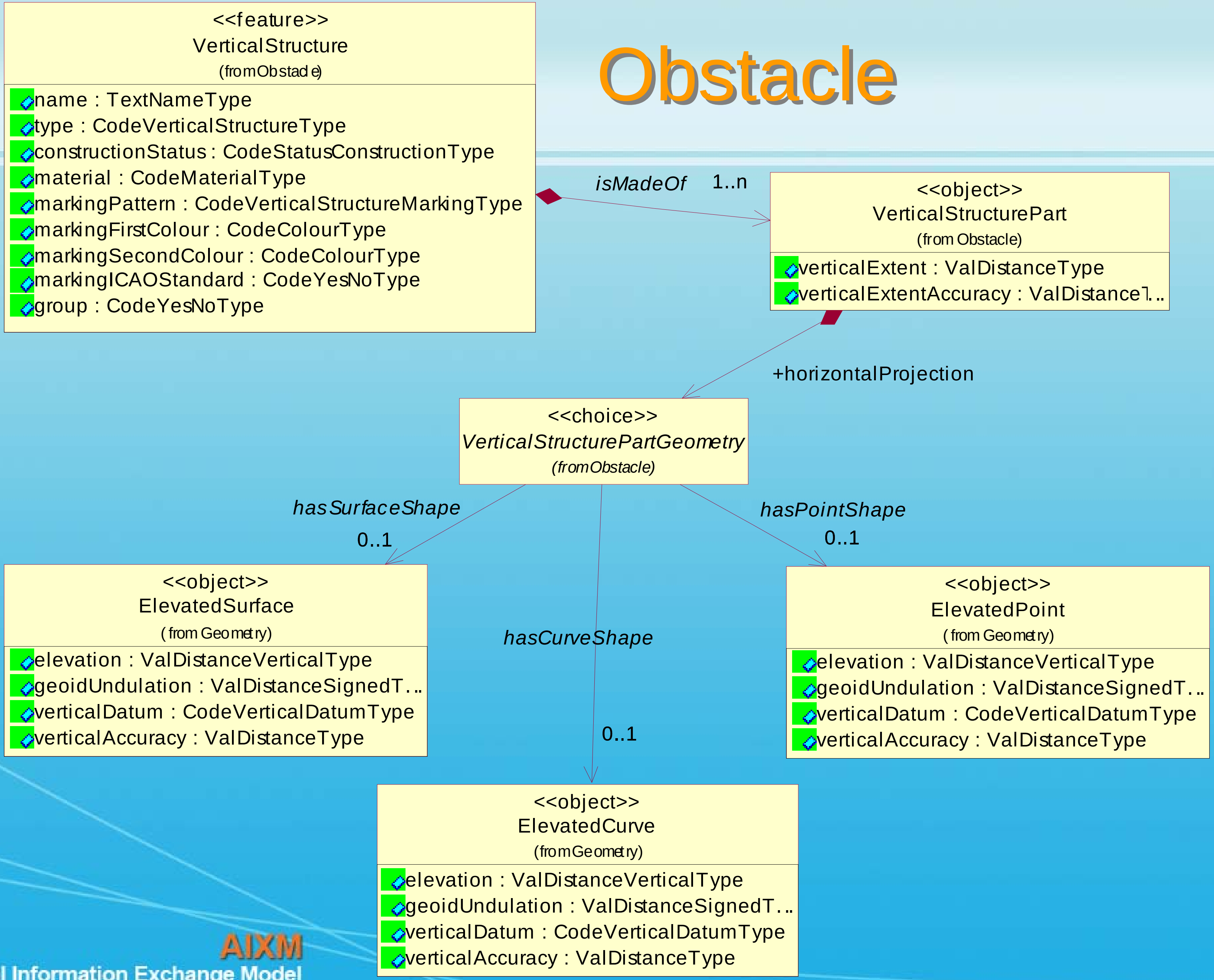
- composition : CodeSurfaceCompositionType
- preparation : CodeSurfacePreparationType
- surfaceCondition : CodeSurfaceConditionType
- classPCN : ValPCNType
- pavementTypePCN : CodePCNPavementType
- pavementSubgradePCN : CodePCNSubgradeType
- maxTyrePressurePCN : CodePCNTyrePressureType
- evaluationMethodPCN : CodePCNMethodType
- classLCN : ValLCNType
- weightSIWL : ValWeightType
- tyrePressureSIWL : ValPressureType
- weightAUW : ValWeightType

Attributes are used to describe simple properties of a feature or object

Datatypes are used to define how the data for an attribute is encoded

- A runway has:
  - Dimensions
    - Width
    - Length
  - Surface characteristics

Obstacle



<<feature>>
VerticalStructure
(from Obstacle)

- name : TextNameType
- type : CodeVerticalStructureType
- constructionStatus : CodeStatusConstructionType
- material : CodeMaterialType
- markingPattern : CodeVerticalStructureMarkingType
- markingFirstColour : CodeColourType
- markingSecondColour : CodeColourType
- markingICAOStandard : CodeYesNoType
- group : CodeYesNoType

isMadeOf 1..n

<<object>>
VerticalStructurePart
(from Obstacle)

- verticalExtent : ValDistanceType
- verticalExtentAccuracy : ValDistanceT...

+horizontalProjection

<<choice>>
VerticalStructurePartGeometry
(from Obstacle)

hasSurfaceShape 0..1

hasCurveShape 0..1

hasPointShape 0..1

<<object>>
ElevatedSurface
(from Geometry)

- elevation : ValDistanceVerticalType
- geoidUndulation : ValDistanceSignedT...
- verticalDatum : CodeVerticalDatumType
- verticalAccuracy : ValDistanceType

<<object>>
ElevatedPoint
(from Geometry)

- elevation : ValDistanceVerticalType
- geoidUndulation : ValDistanceSignedT...
- verticalDatum : CodeVerticalDatumType
- verticalAccuracy : ValDistanceType

<<object>>
ElevatedCurve
(from Geometry)

- elevation : ValDistanceVerticalType
- geoidUndulation : ValDistanceSignedT...
- verticalDatum : CodeVerticalDatumType
- verticalAccuracy : ValDistanceType

AIXM UML class diagrams can be categorized by their Stereotypes

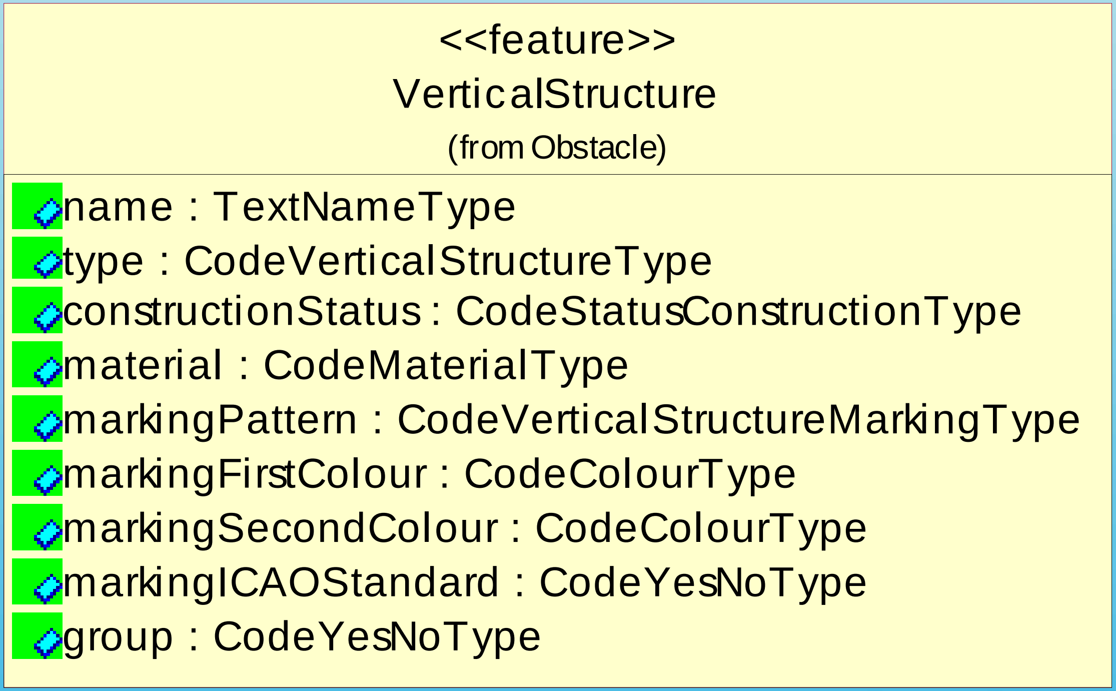
- Stereotypes are used to further define and extend standard UML concepts.
- Stereotypes enable us to have more than one type of UML class diagram in the model
- Aeronautical Features
  - <<feature>>
  - <<object>>
  - <<choice>>
- Their associated datatypes:
  - <<datatype>>
  - <<enumeration>>
  - <<codelist>>

Aeronautical Features

- <<feature>> Models real world features that change with time
- <<object>> Used to represented geometry
- <<choice>> Used to codify the choice of one (feature,object) amongst several (features,objects)

Aeronautical Features of Stereotype <<feature>>

- Real World
  - Airport
  - Runway
  - Taxiway
  - Airspace
  - Obstacle (generalized as a vertical structure)
- Change in time
  - Runway closed
  - Airspace redefined

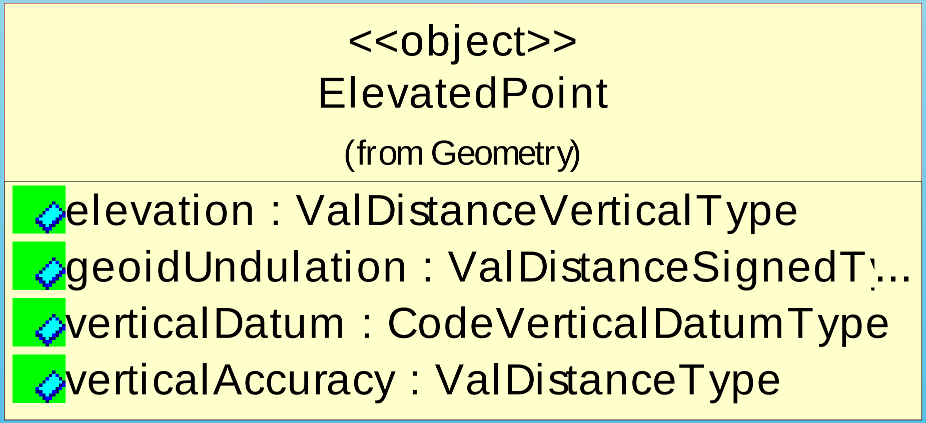


<<feature>>
VerticalStructure
(from Obstacle)

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

Aeronautical Features of Stereotype <<object>>

- The object has its own attributes that are reused throughout the model
- A property has a multiplicity greater than one

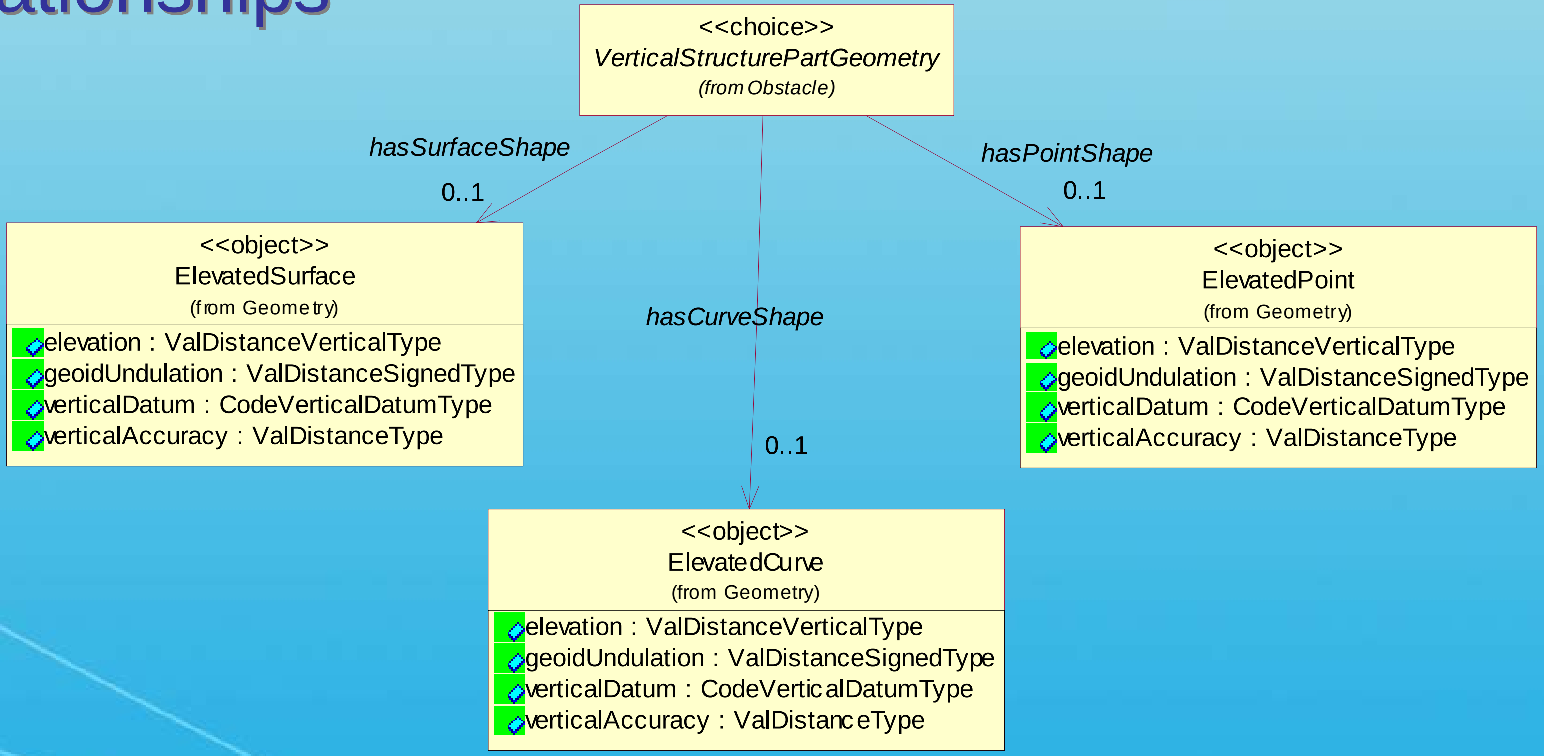


<<object>>
ElevatedPoint
(from Geometry)

- elevation : ValDistanceVerticalType
- geoidUndulation : ValDistanceSignedTy...
- verticalDatum : CodeVerticalDatumType
- verticalAccuracy : ValDistanceType

Aeronautical Features of Stereotype <<choice>>

- Used to model XOR relationships



<<choice>>
VerticalStructurePartGeometry
(from Obstacle)

hasSurfaceShape
0..1

hasCurveShape
0..1

hasPointShape
0..1

<<object>>
ElevatedSurface
(from Geometry)

- elevation : ValDistanceVerticalType
- geoidUndulation : ValDistanceSignedType
- verticalDatum : CodeVerticalDatumType
- verticalAccuracy : ValDistanceType

<<object>>
ElevatedPoint
(from Geometry)

- elevation : ValDistanceVerticalType
- geoidUndulation : ValDistanceSignedType
- verticalDatum : CodeVerticalDatumType
- verticalAccuracy : ValDistanceType

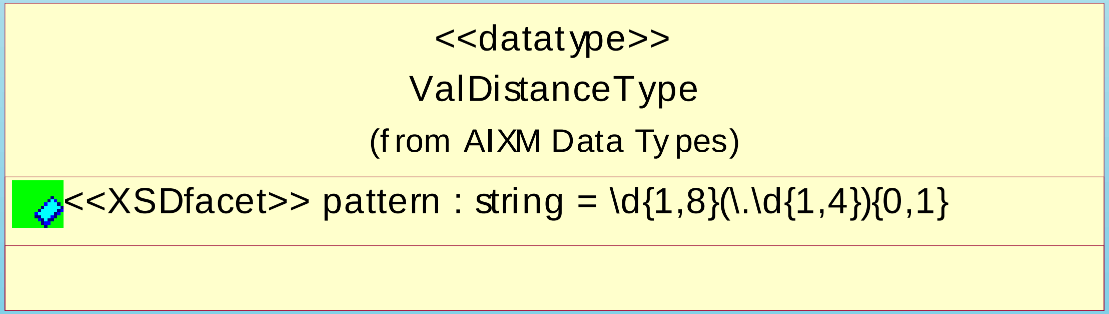
<<object>>
ElevatedCurve
(from Geometry)

- elevation : ValDistanceVerticalType
- geoidUndulation : ValDistanceSignedType
- verticalDatum : CodeVerticalDatumType
- verticalAccuracy : ValDistanceType

Datatypes for Aeronautical Features

- <<datatype>> – basic data type that specifies a pattern to use; usually used for labels and identifiers
- <<enumeration>> – a fixed list of values, such as the organizations associated with NOTAM regulations
- <<codelist>> – similar to an enumeration but the list can be expanded, such as the surface characteristics of runways

Associated Datatypes of Stereotype <<datatype>>



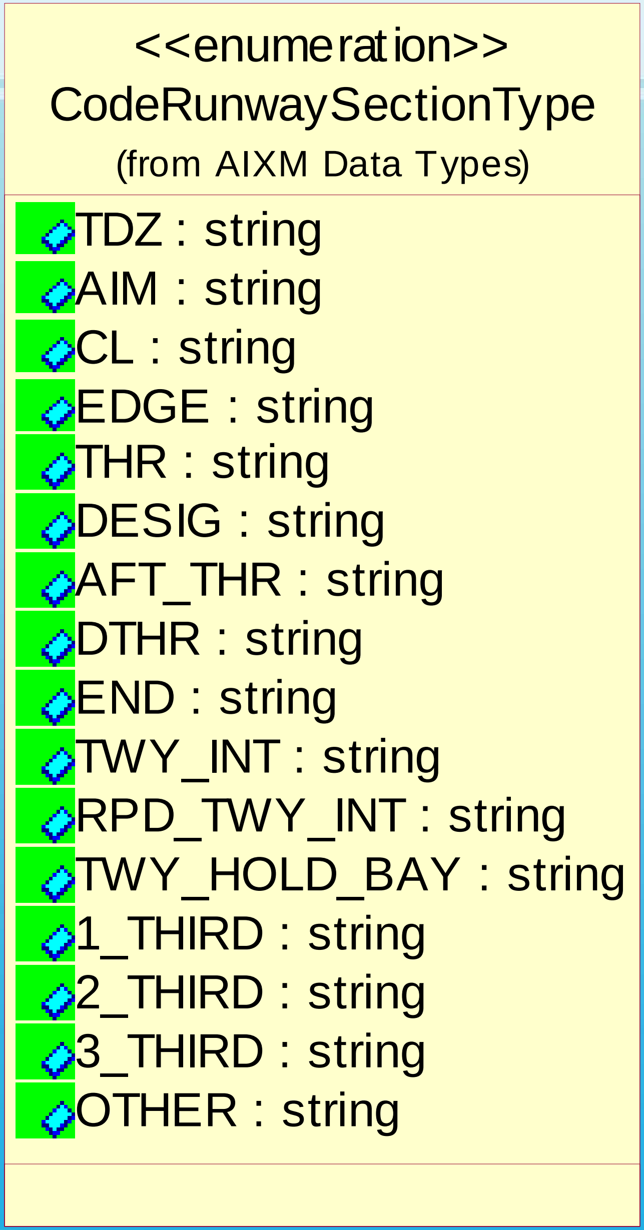
<<datatype>>
ValDistanceType
(from AIXM Data Types)

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

- Pattern
- Minimum Length
- Maximum Length

Associated Datatypes of Stereotype <<enumeration>>

- This list cannot be expanded

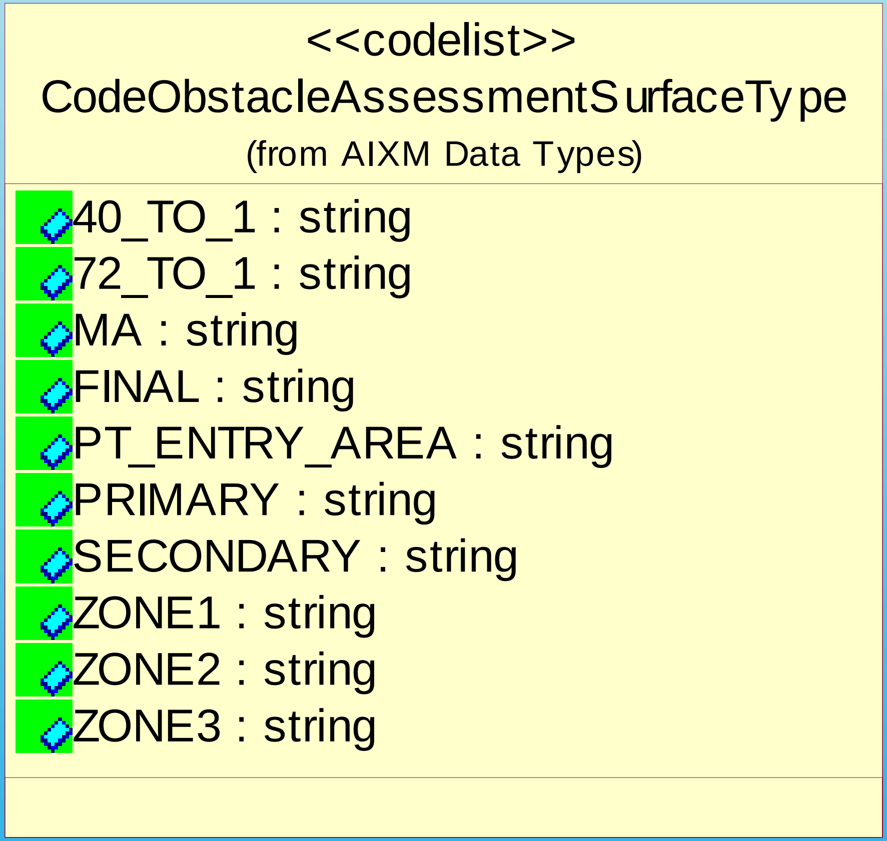


<<enumeration>>
CodeRunwaySectionType
(from AIXM Data Types)

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

Associated Datatypes of Stereotype <<codelist>>

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



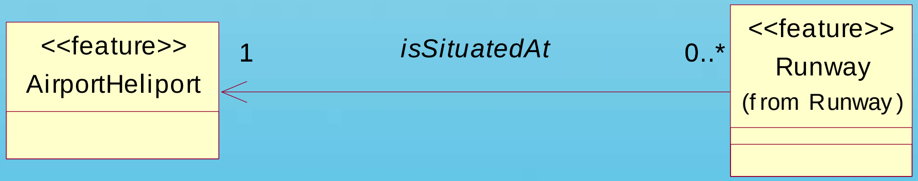
<<codelist>>
CodeObstacleAssessmentSurfaceType
(from AIXM Data Types)

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

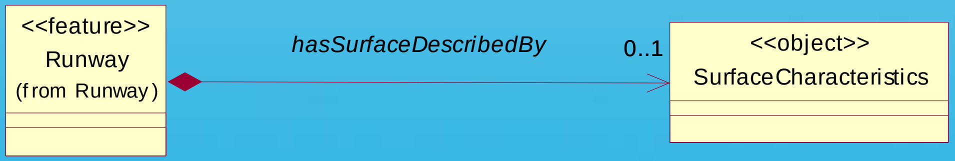
Properties - Relationships

- Relationships are used to describe associations to features or objects
  - All relationships have uni-directional navigability

- Association used when the relationship refers to a feature

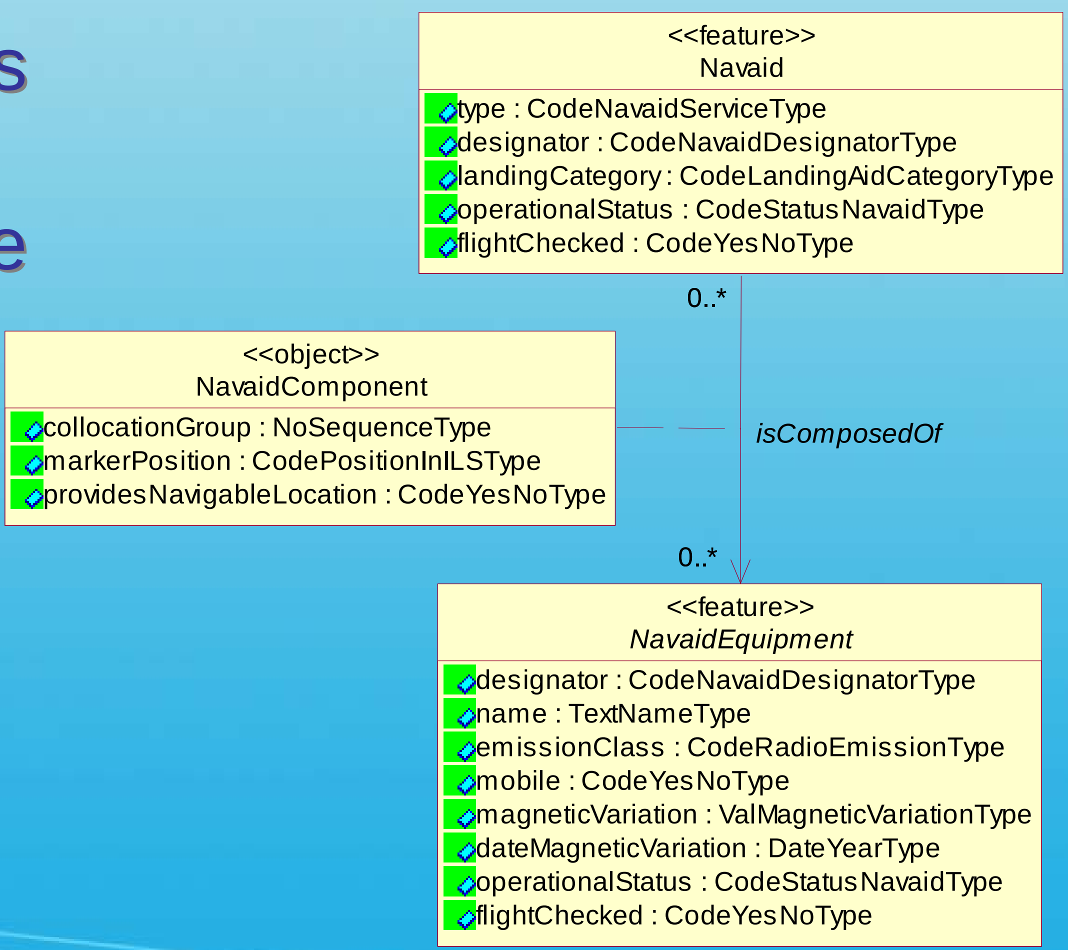


- Relationships to objects are depicted by the standard UML composition
  - The object does not exist by itself



Properties - Relationships

- Standard UML association class is used to describe attributes about the relationship



<<feature>>
Navaid
type : CodeNavaidServiceType
designator : CodeNavaidDesignatorType
landingCategory : CodeLandingAidCategoryType
operationalStatus : CodeStatusNavaidType
flightChecked : CodeYesNoType
0..*
<<object>>
NavaidComponent
collocationGroup : NoSequenceType
markerPosition : CodePositionInILSType
providesNavigableLocation : CodeYesNoType
isComposedOf
0..*
<<feature>>
NavaidEquipment
designator : CodeNavaidDesignatorType
name : TextNameType
emissionClass : CodeRadioEmissionType
mobile : CodeYesNoType
magneticVariation : ValMagneticVariationType
dateMagneticVariation : DateYearType
operationalStatus : CodeStatusNavaidType
flightChecked : CodeYesNoType

Naming Conventions

- Feature, Object and Choice names are written in UpperCamelCase e.g. NavaidEquipment
- Simple property names (i.e. attributes) are written in lowerCamelCase e.g. widthShoulder
- Relationship names are written in lowerCamelCase but as present tense verbs e.g. isSituatedAt
- Datatype names are written in UpperCamelCase and end with ‘Type’ e.g. CodeAircraftType

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

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 is GML

- AIXM is an XML exchange standard based on a subset of GML. Essentially:
  - AIXM Features are GML features
  - AIXM Objects are GML objects
  - AIXM follows the GML object-property concept
- To understand how AIXM is GML, we first need to describe the AIXM model using the XML Schema Definition Language (XSDL)

GML Basics

- Consists of 28 core XSD schemas
- AIXM uses:
  - xlinks.xsd (as is)
  - A compilation of GML definitions from the other GML core schemas in two files:
    - AIXM-AbstractGML-ObjectTypes.xsd
    - gml4aixm.xsd