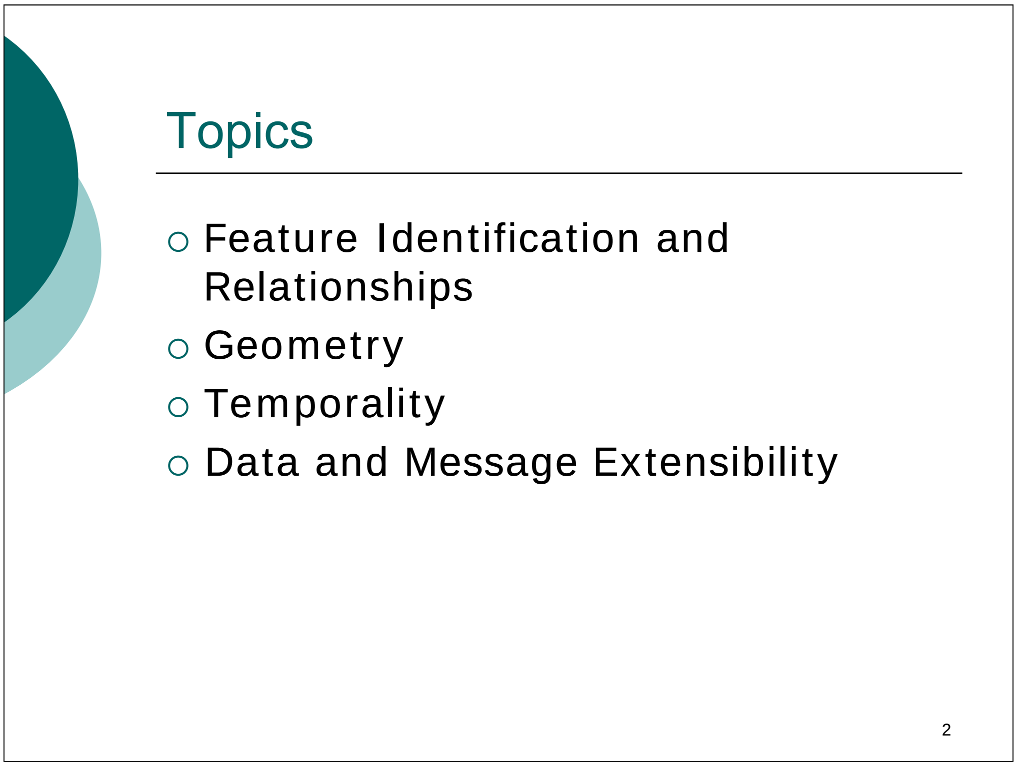
AICM/AIXM 5 Design Concepts

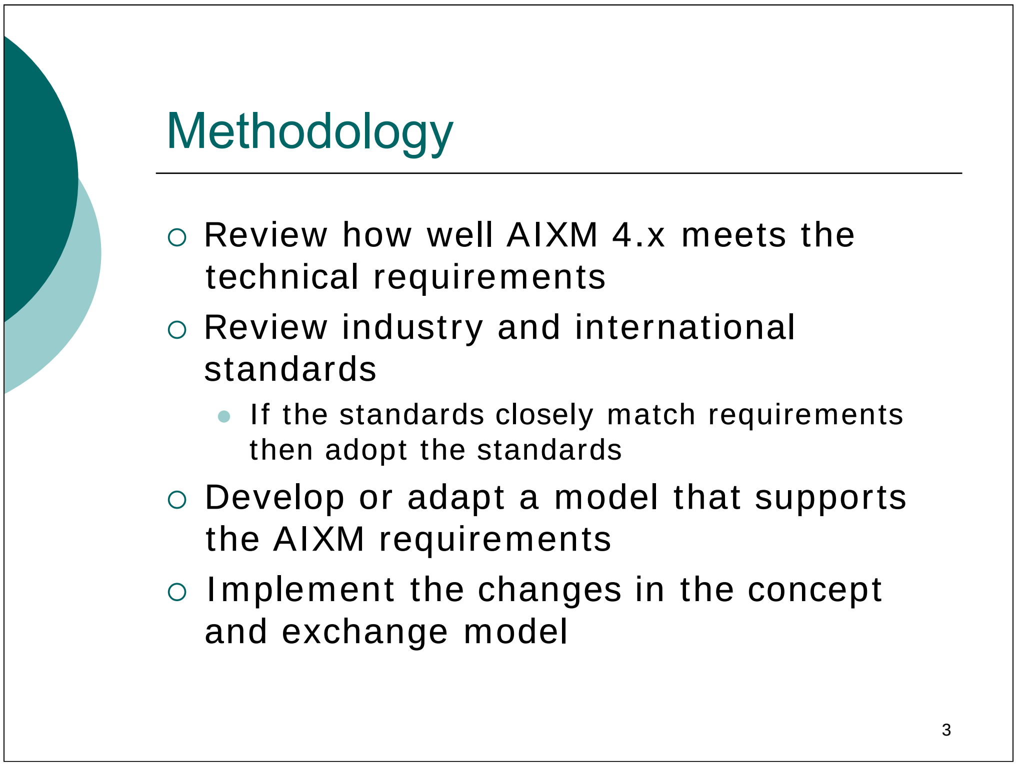
AIXM 5 Public Design Review
February 7-8, 2006
Washington DC

The purpose of this briefing is to present the major design concepts for AICM and AIXM 5.



We are going to cover 4 major topics:

1. Feature identification and relationships
2. Geometry
3. Temporality
4. Data and Message Extensibility.



During the AIXM 5 requirements analysis and design process we followed a methodology to evaluating changes and how these changes would be implemented in AIXM.

First we reviewed how well AIXM 4.x meets the technical requirements. Next we reviewed industry and international standards to see how these issues have been resolved by others. If the standards closely match our requirements then we tried to adopt the standard.

Next we develop or adopted the model to support our aeronautical information exchange requirements. Finally we implemented the changes in the conceptual and exchange models.

Topics

- Feature Identification and Relationships
- Geometry
- Temporality
- Data and Message Extensibility

Let’s begin by discussing feature identification and feature relationships.

Definitions

- Feature identification
  - How do we uniquely identify an important aeronautical entity
    - KJFK Aerodrome
    - Taxiway N
- Feature relationships
  - How do we associate one feature with another
    - Runway 25 is at Airport MXAB
    - AML Navaid is the starting Point on a Procedure Leg

5

Feature identification is how we uniquely identify an important aeronautical entity. For example KJFK airport or Taxiway N. Is this identified using a number or a set of feature properties?

Feature relationships is closely related. How do we associate one feature with another. For example Runway 25 at Airport MXAB. Or AML Navaid is the starting Point on a Procedure Leg.

Feature Identification and Relationships

A safety critical issue

- Safety critical applications
  - NOTAMS indicating a change in operating environment
  - Avionics updates – FMS, Electronic Flight Bag, Situational Awareness
- Must ensure positive feature identification

6

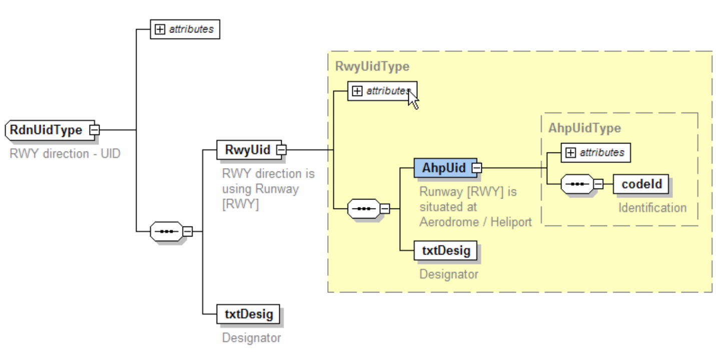
Feature identification and feature relationships are a safety critical issue. Suppose you send out information about Taxiway N but a receiving system has no way of knowing exactly what taxiway you are talking about. This can lead to problems and potential safety issues.

Imagine if we had problems with feature identification for NOTAMS, FMS, flight bag or other situational awareness systems. It is therefore, important that the AIXM data standard include mechanisms to ensure positive feature identification.

Feature Identification and Relationships

Review of AIXM 4.x

- Natural keys
  - Groups of feature properties and relationships used to identify features



7

AIXM 4 handles feature identification using natural keys. Natural keys are groups of feature properties and relationships to identify features. In this example taken from XML Spy, A runway direction can be uniquely identified from the Runway Direction txtDesig (designator) and its relationship to a Runway. Note that a Runway is uniquely identified by the Runway Designator and its relationship to the Aerodrome/Heliport. Finally the Aerodrome/Heliport is identified by its identification code. Normally this code is the Aerodrome/Heliport ICAO code.

Feature Identification and Relationships

Review of AIXM 4.x

```
<Gsd>
  <GsdUid>
    <ApnUid>
      <AhpUid>
        <codeId>KBPT</codeId>
      </AhpUid>
      <txtName>TRANSIENT PARKING</txtName>
    </ApnUid>
    <txtDesig>PARKING1</txtDesig>
  </GsdUid>
  <codeType>UKN</codeType>
  <geoLat>295725N</geoLat>
  <geoLong>0940115W</geoLong>
  <codeDatum>NAW</codeDatum>
</Gsd>
```


Here is another example, but in XML. You can see that the Gatestand is uniquely identified by a Designator and its relationship to an Apron. The Apron is identified by its designator and a relationship to the Aerodrome/Heliport.

Feature Identification and Relationships
Issues with Natural Keys

- Unnamed aeronautical features
  - Small domestic aerodromes

Louis Bennett Field Airport
Weston, West Virginia, USA

Small airfield
Where is it?
No standard ID

- Features without natural keys
  - Runway markings, Procedure Legs

9

Natural keys can be difficult to uniformly apply in a global setting. Some problems with natural keys are shown on this slide.

1. Unnamed aeronautical features. Many times smaller facilities don't have globally understood natural identifies. For example, small domestic aerodrome may not have ICAO or IATA codes, thus providing a natural key that is consistent across data suppliers is problematic.
2. Features without natural keys. Even for large facilities, sometimes natural keys cannot be identified for features. For example, Runway Markings are a feature that we need to be able to exchange if we are going to support aerodrome mapping applications. Runway Markers are features, but there is not good natural key.

Feature Identification and Relationships
Issues with Natural Keys

- Geographic mismatch
  - Two NAVAIDS with same codeID and slightly different locations
  - Differences in precision, datum, representation

<geoLat>332000N</geoLat>

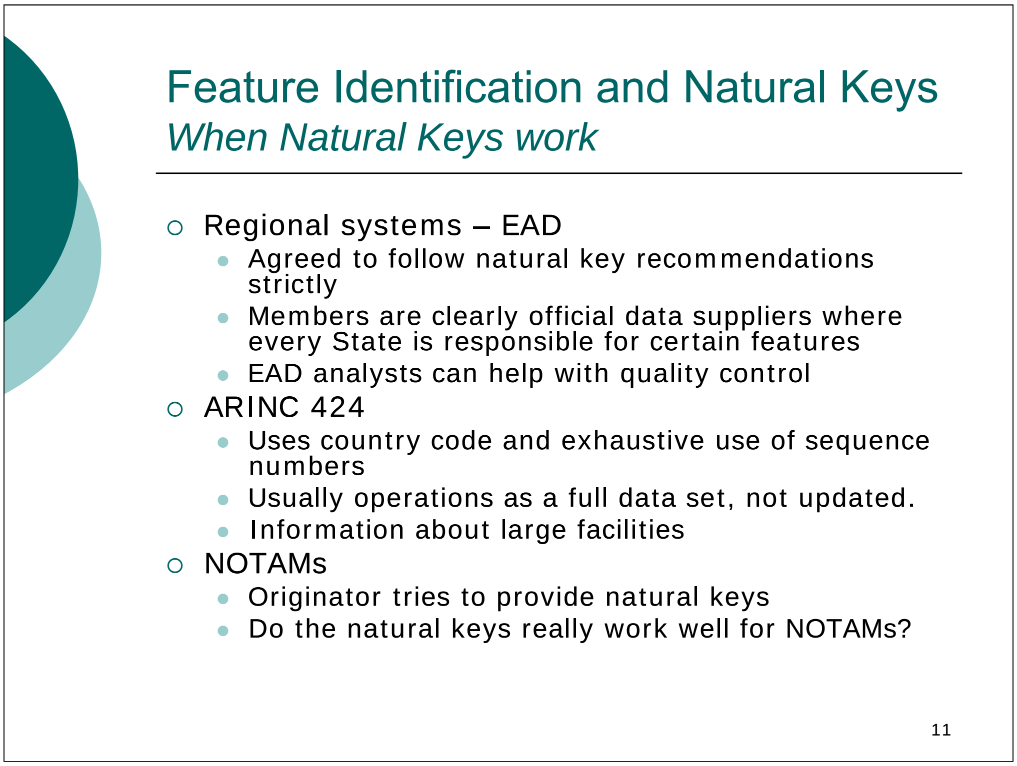
<geoLat>33.33333N</geoLat>

- Overloaded purpose
  - Natural keys act as feature identification and feature properties
  - Issues when properties in natural keys change

10

Natural keys can be difficult to uniformly apply in a global setting. Some problems with natural keys are shown on this slide.

1. Geographic Mismatch. Using latitude and longitude in natural keys can lead to trouble. Two data suppliers may have different natural keys because of precision differences, datum changes or whether the information is encoded as decimal degrees or degrees minutes seconds.
2. Overloaded purpose. From a data modeling and system interfacing perspective natural keys also pose a problem. Natural keys overload the purpose of feature properties and make interpretation more complex. Some properties of the feature serve two purposes – feature identification and feature characteristic. If one of the properties changes this can lead to a change in the natural key. This change can be difficult to synchronize with everyone receiving or transmitting the aeronautical information.



Despite these technical issues, we should acknowledge that natural keys do work in some situations.

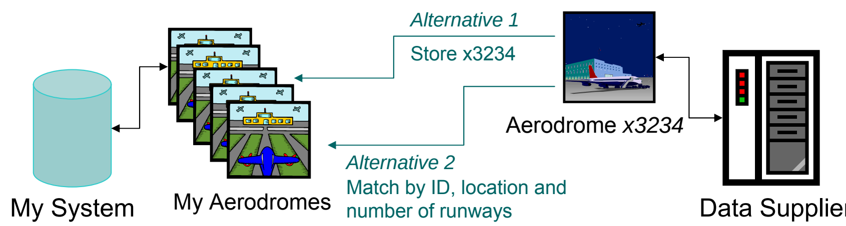
The EAD has been successfully using AIXM 4.x with natural keys in their implementation. However, EAD is a regional system that is established to harmonize and centralize aeronautical information within Europe. EAD member states are official data providers so they can provide the defining natural key for each feature. In addition the States have agreed to strictly apply ICAO SARPs and AIXM natural key recommendations.

ARINC 424 also uses carefully designed natural keys. Arinc typically uses the country code to prefix all features and makes extensive use of sequence numbers. In addition, ARINC 424 databases are usually uploaded in full and do not undergo updates. Finally ARINC typically contains information for large facilities.

Finally NOTAMS also use natural keys to identify features and operational status updates. Do natural keys really work well for NOTAMS?

Feature Identification and Relationships
Alternatives

- Artificial identifiers
  - Keys provided by the data supplier
  - Feature identification
    - User may need to use business rules to reconcile data with their system
    - Or store data supplier keys in their database.

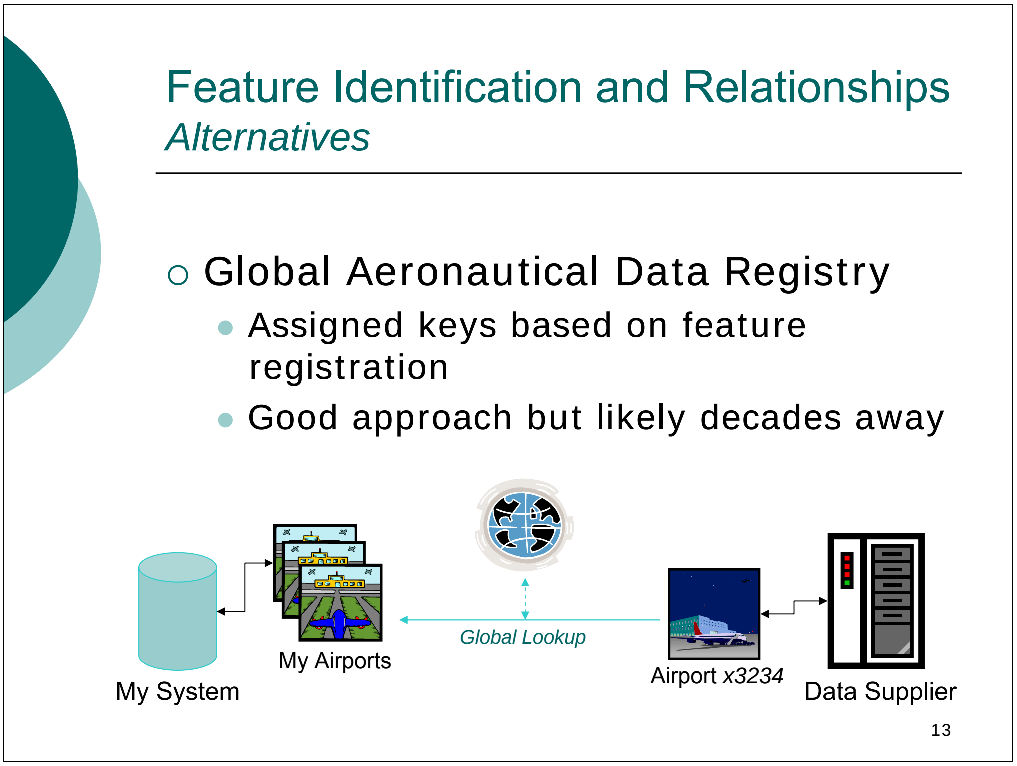


12

An alternative to consider is artificial identifiers. This might be keys provided by the data supplier. For example I may have a database system containing aerodromes and I have just received a data packet from my data supplier. This packet might contain the data suppliers key of x3234 for an Aerodrome.

How could I incorporate this into my system? One alternative would be store the artificial key of the data supplier into my system. That way I know exactly who gave me the record and what they called it. Another alternative is to use business rules to reconcile the incoming data with the my existing database. Maybe I first try matching the incoming aerodrome by the identification code and then I check the aerodrome reference point and number of reported runways.

So with this approach maintaining feature identification becomes a data receiver burden. The data supplier maintains the supplied keys and the data receiver has to use rules to incorporate the received data into the receiver’s system. Notice also that the receiver has the flexibility to set the reconciliation business rules as appropriate.



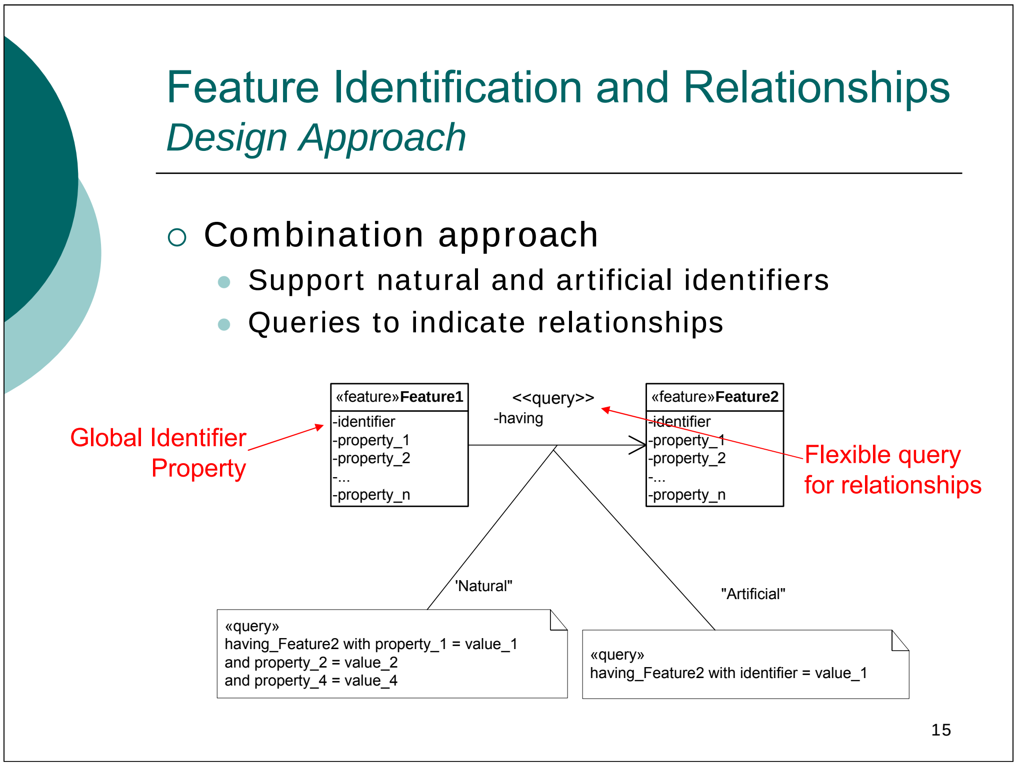
Perhaps an future alternative might be a global aeronautical registry where feature identification is registered at an official site. Registered features would receive keys assigned by the global registry and information exchange would occur using these global keys. This would be a good approach, but unfortunately a global aeronautical data registry does not exist today and it is likely decades away.

Feature Identification and Relationships
Recommendation Approach

- Flexible use of artificial or natural identification
- Support global registry if it becomes a reality
- Eliminate problem of property overloading
- Allow data supplier to provide natural key encoding rules

14

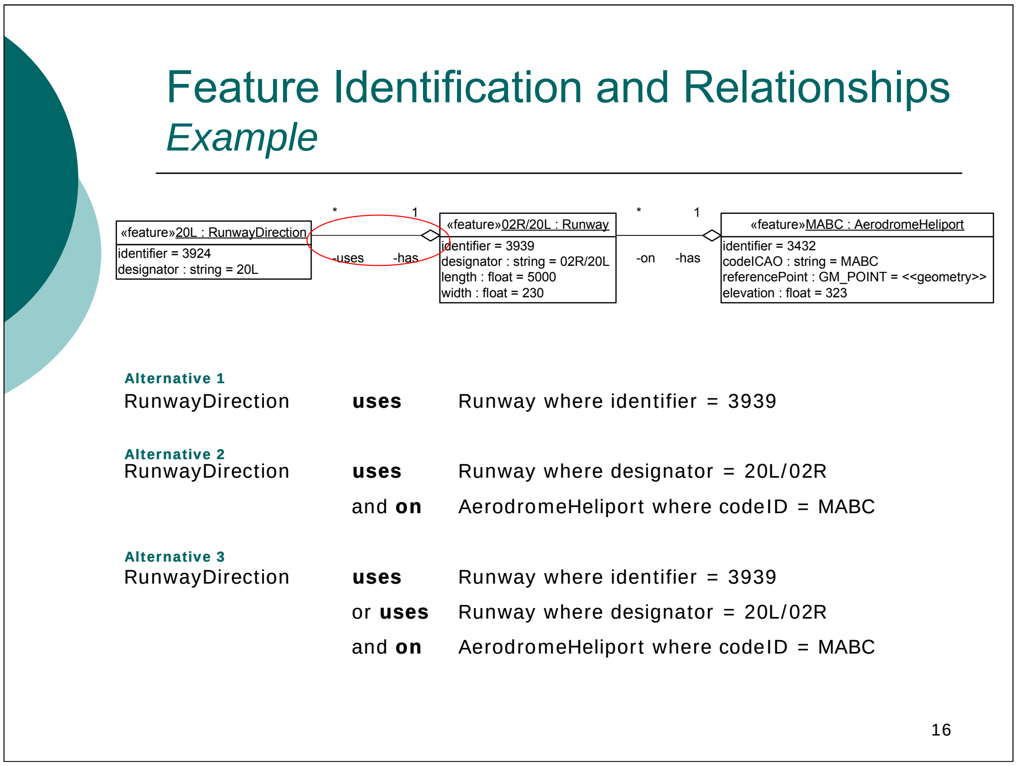
Our recommendation is to use a flexible approach that can accommodate any type of feature identification. We recognize that feature identification is complicated and what works for one community of interest, may not work for another community. We propose the flexible use of natural and/ or artificial identification. Should a global registry be created, our approach will work with the registry as well. We propose to eliminate the problem of property overloading and allow the data supplier to specify the properties that should be included in a natural key or a relationship.



Our combined approach support artificial identification and the use of natural properties in feature relationships. All features now include an identifier property meant to hold the global key for the artificial key of the data originator.

Relationships are encoded using an abstract <<query>> concept similar to a database SQL query. In this approach the data supplier indicates relationships by encoding a query against one or more properties of the related feature.

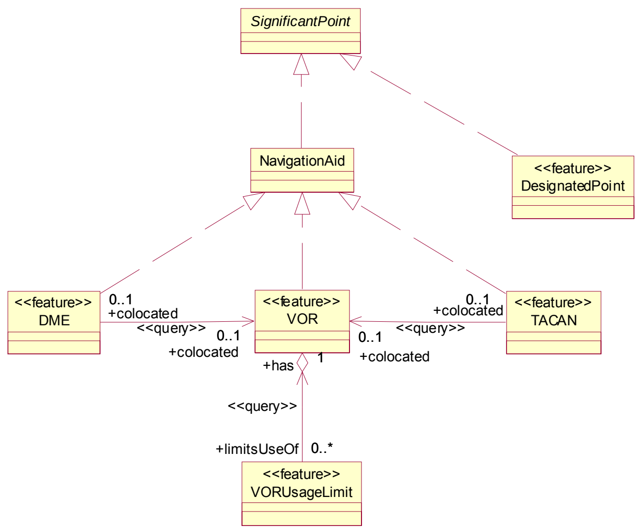
This slide shows an example of a Feature1 having a relationship to Feature2. Using the query approach we can encode the relationship in an unlimited number of ways. A “Natural” encoding might say, Feature1 has a relationship to a Feature2 where Feature 2 property_1 = value_1, property_2 = value_2 and property_4 = value_4. As an alternative an “Artificial” relationship might indicate Feature1 is related to a Feature2 where Feature2 identifier = value_1. Obviously any number of properties can be encoded into this flexible <<query>> approach so that the data supplier can indicate how to find related features.



This slide show an example of three alternative encodings for the relationship between a Runway Direction and a Runway. A Runway Direction uses a Runway. How would we encode the “uses” relationship in AIXM?

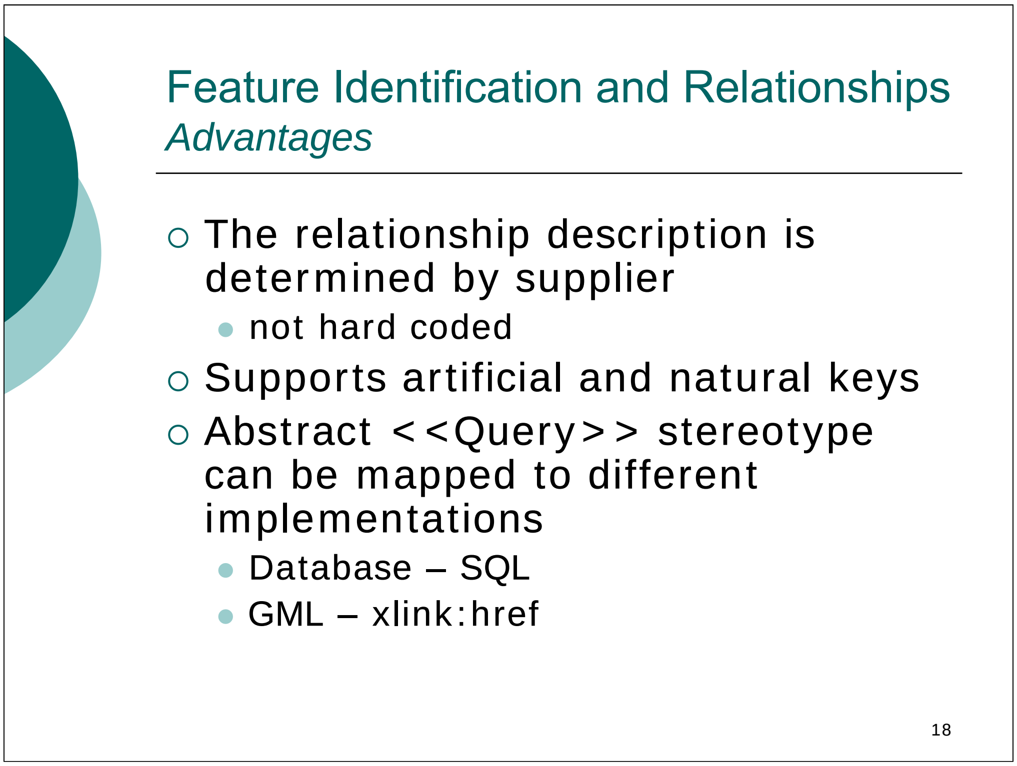
Alternative 1 shows a solution using artificial keys. Alternative 2 is a solution using natural property identification. Alternative 3 combines the two approaches and indicates a way to specify the relationship using the artificial identifier or the natural identifier.

UML Example
<<query>> stereotype



17

This UML diagram shows a portion of the navigation aid model. You can see that VOR is colocated to DME and TACAN through the <<query>> relationship and that the VOR has VORUsageLimits through another <<query> relationship.



The abstract <<query>> approach has several advantages:

1. The relationship is determined by the data supplier and it is not hard coded. Depending on the feature, the data supplier has the flexibility to use the best natural key.
2. The approach supports, artificial identification, natural identification or a combination.
3. The abstract <<query>. Concept can be mapped to different implementations. In a database the <<query>> can be mapped to SQL and in GML it can be implemented as xlink:href.

Feature Identification and Relationships

Modeling Relationships in UML and GML

- Navigability
  - A runway is **situatedAt** an aerodrome
  - An aerodrome **has** one or more runways


```
<AerodromeHeliport>
   <has>
      <Runway>…</Runway>
   </has>
</AerodromeHeliport>
```

Relationship in Aerodrome

```
<Runway>
   <situatedAt
       xlink:href="AerodromeHeliport"/>
</Runway>
```

Relationship in Runway

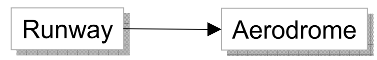
19

Not only do we need to specify how to represent a relationship, but we also need to decide relationship directionality. Navigability refers a relationship is traversed in an implementation. In the Runway and Aerodrome picture, the navigability is not provided. Does the runway have a link to the Aerodrome, does the Aerodrome have a link to the Runway or are both links available?

Feature Identification and Relationships

Modeling Relationships in UML and GML

- Navigability



  - Use an arrow to indicate preferred direction for relationship
  - A runway **situatedAt** an Aerodrome

```
<Runway>
   <situatedAt
      xlink:href="AerodromeHeliport"/>
</Runway>
```

Relationship in Runway

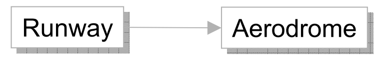
20

Our convention is to specific relationship navigability using an arrow to indicate the preferred direction. So, in this example, the Runway feature contains a relationship to the Aerodrome.

Feature Identification and Relationships

Modeling Relationships in UML and GML

- Encoding



  - Inline encoding with target explicit
  - Remote reference using xlink:href

```
<AerodromeHeliport>
   <has_Runway>
      <Runway>…</Runway>
   </has_Runway>
</AerodromeHeliport>
```

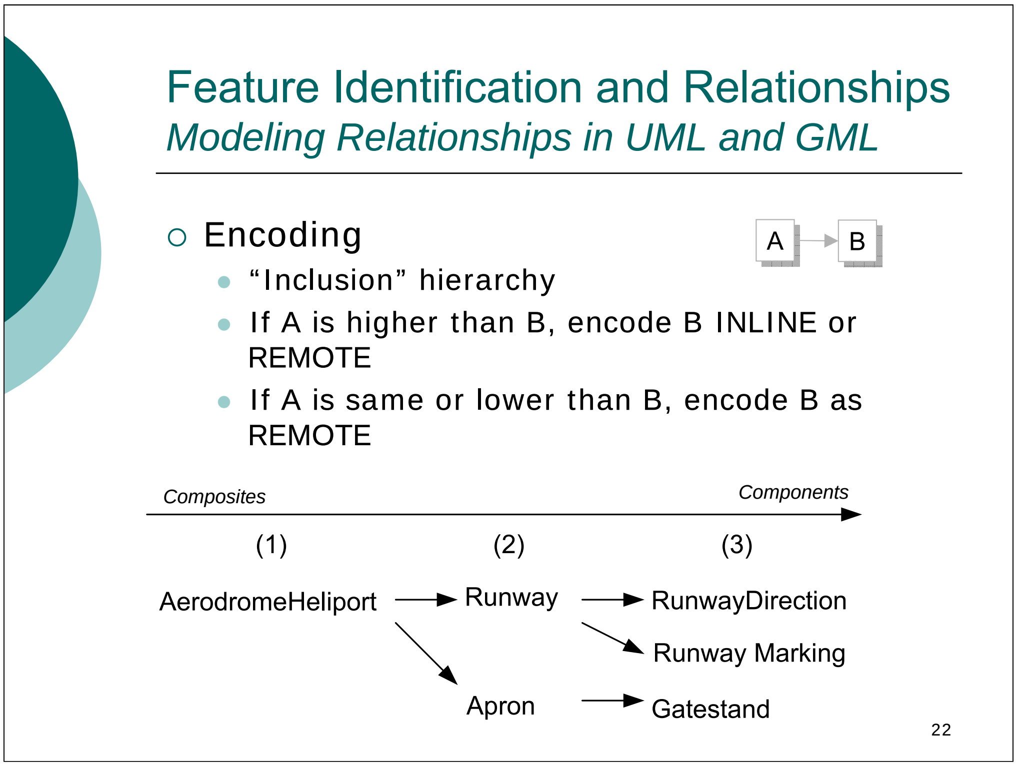
Inline

```
<AerodromeHeliport>
   <has_Runway
      xlink:href="Runway"/>
</AerodromeHeliport>
```

Remove Reference in Runway

21

How should relationships be encoded: inline or by reference?

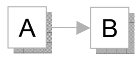


We use an inclusion hierarchy to determine whether to encode the relationship inline or by reference. If I have a relationship A is related to B. If A is higher in the hierarchy than B then I can encoded B inline or by reference. If B is equal to or higher than A then B is encoded by reference.

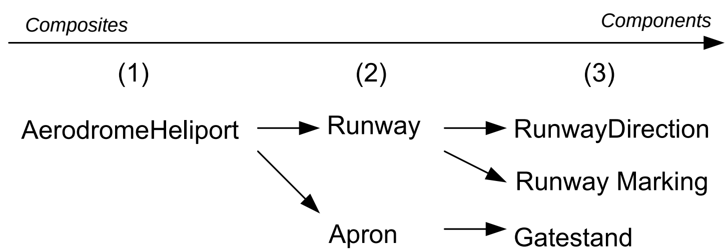
A partial hierarchy is shown for the AerodromeHeliport and several of its components.

Feature Identification and Relationships

Modeling Relationships in UML and GML



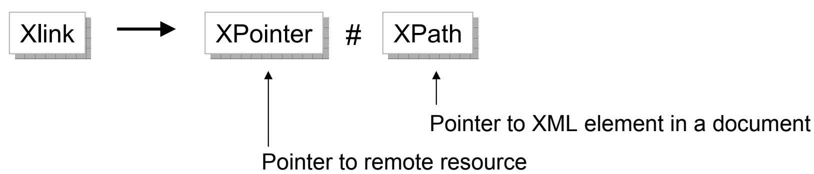
Feature A	Feature B	A->B is Encoded
AerodromeHeliport	Runway	Inline or Remote
Gatestand	AerodromeHeliport	Remote
RunwayMarking	Runway	Remote
Runway	Apron	Remote
Runway	Runway Marking	Inline or Remote



23

Feature Identification and Relationships
Encoding Relationships in GML

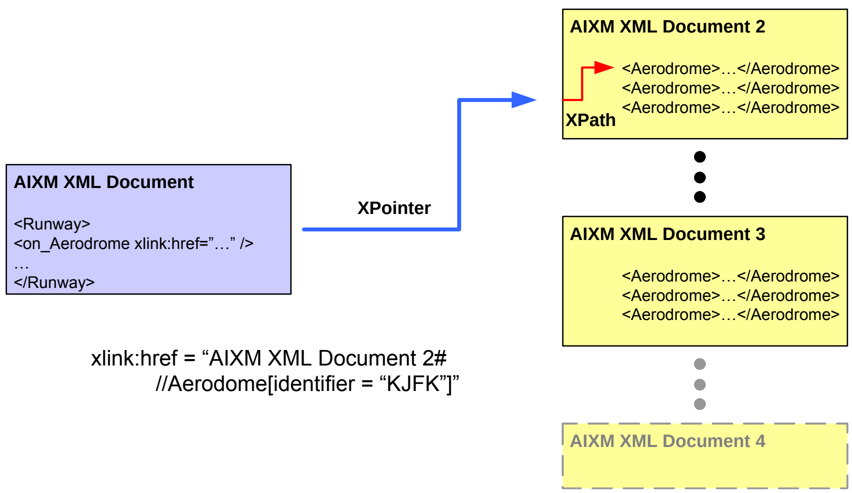
- GML relationships
  - Inline or Remote
- Remote GML relationships
  - Xlink:href
    - See http://www.brics.dk/~amoeller/XML/linking/



24

Feature Identification and Relationships

Encoding Relationships in GML



25

Feature Identification and Relationships
Encoding Relationships in GML

Aerodrome with a specified ID and location

```
xlink:href=”http://www.eurocontrol.int/ead#
  //aixm:AerodromeHeliport[
    aixm:icaoCode=”EDDF” and
    aixm:position/gml:Point/gml:pos=’50.0333 8.5704’]”
```

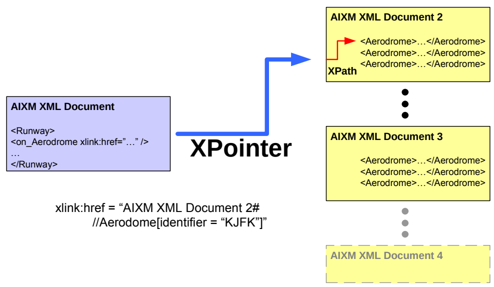
Runway with a specified designator and aerodrome

```
xlink:href='http://www.faa.gov/nasr#
  //aixm:Runway[
    aixm:designator="09/27" and
    aixm:on_AerodromeHeliport/AerodromeHeliport[
      aixm:icaoCode="EDDF" and
      aixm:position/gml:Point/gml:pos="50.0333 8.5704"
    ]
  ]
’
```

26

Xlink implementation issues

- XPointer – Pointer to a remote resource
  - An XML document
  - A web service
  - <u>No requirement that this is resolvable</u>



Xlink implementation issues

- Xlink resolution is outside the scope of AIXM
  - System to system implementation issue
  - Handled via Service Level Agreements
  - In the future XLinks may be resolvable
- Advantages of XLink
  - Standard way to point to resource and element inside resource
  - Tools exist to interpret and transform XLink references
  - Supports artificial and natural identification

28

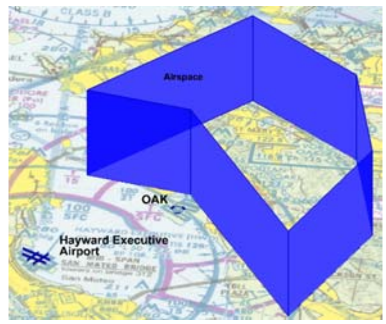
Topics

- Feature Identification and Relationships
- Geometry
- Temporality
- Data and Message Extensibility

Geometry

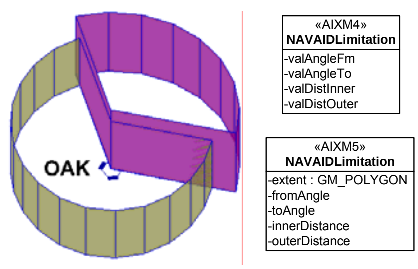
In AIXM 4.x

- Based on aeronautical domain
- Custom construction
- 2 ½ D
  - Horizontal boundary
  - Properties for upper and lower limits.



Geometry
Recommendations

- Standardize geometries using ISO19107 Spatial Schema
  - GM_POLYGON
  - GM_LINE
  - Consistent with GML (ISO19136)
- Augment with aeronautical properties where needed



31

Geometry
GML Profile for AIXM

- Profile defined
  - Limits AIXM to a subset of GML features
  - Simplifies implementation of AIXM
- GML Profile for AIXM
  - Remove pre-GML 3.1.1 deprecated elements
  - Restrict choice
  - Limited set of geometry and temporal properties

Geometry

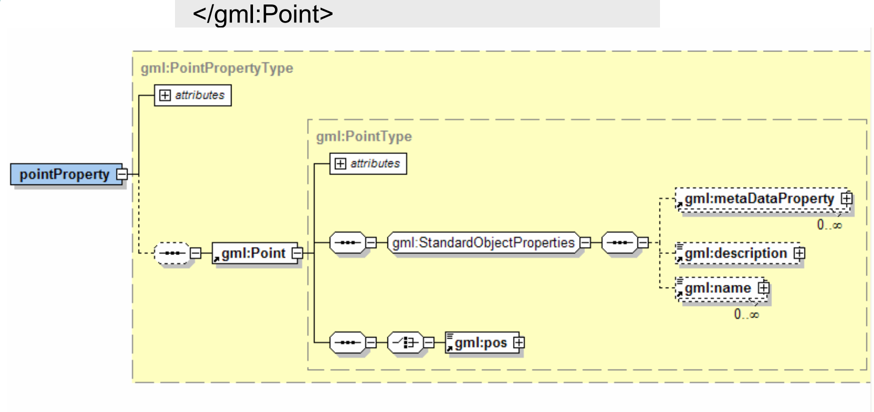
GML Profile for AIXM

- Arc
- ArcByCenterPoint
- CircleByCenterPoint
- CompositeSurface
- Curve
- Geodescic
- LineString
- MultiPoint
- Point
- Polygon
- Ring
- Surface

Example GML geometries

Point

```
<gml:Point>
  <gml:pos>-101.3929 25.323</gml:pos>
</gml:Point>
```



Example GML geometries

Polygon – Circle by Center Point

```
<gml:Polygon>
    <gml:exterior>
        <gml:Ring>
            <gml:curveMember>
                <gml:Curve>
                    <gml:segments>
                        <gml:CircleByCenterPoint numArc="1">
                            <gml:pos>-101.3929 25.323</gml:pos>
                            <gml:radius uom="nm">200</gml:radius>
                            <gml:startAngle uom="degree">0</gml:startAngle>
                            <gml:endAngle uom="degree">360</gml:endAngle>
                        </gml:CircleByCenterPoint>
                    </gml:segments>
                </gml:Curve>
            </gml:curveMember>
        </gml:Ring>
    </gml:exterior>
</gml:Polygon>
```

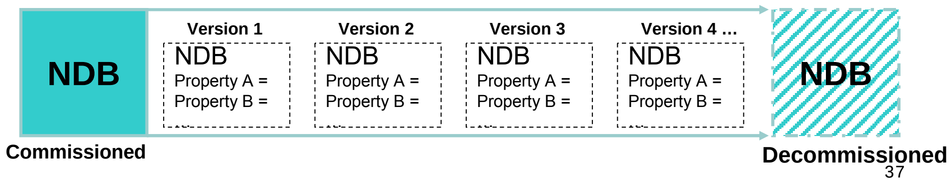
Topics

- Feature Identification and Relationships
- Geometry
- Temporality
- Data and Message Extensibility

36

Temporality
Requirements

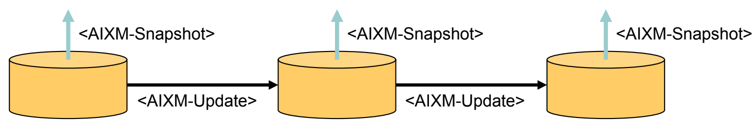
- Aeronautical Features have
  - Start of Life and End of Life
  - Feature properties can change with time
- Classify changes
  - Temporary – like NOTAMs
  - Permanent – like AIRAC Cycle



37

Temporality
In AIXM 4.x

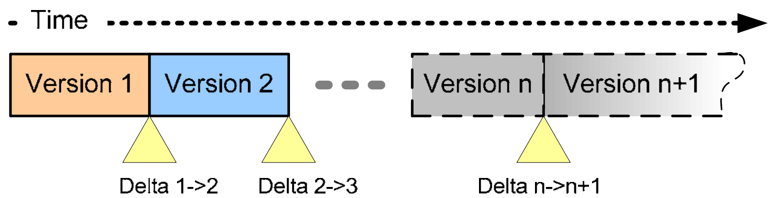
- Temporality in the Messages
  - <AIXM-Update>
  - <AIXM-Snapshot>
- Disadvantages
  - New messages will need a new temporal model (e.g., WFS, xNOTAM, etc)
  - Do not support temporary changes



38

Temporality

Conceptual Model

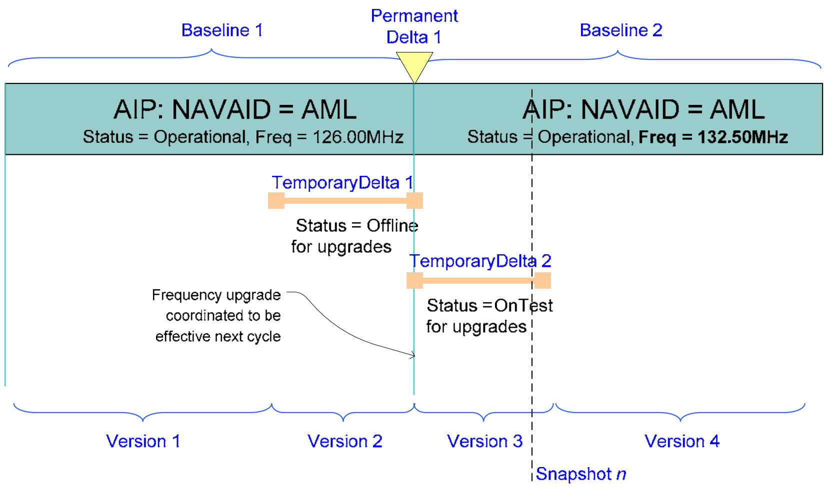


- **Version** – State of a feature and all its properties over a time period
- **Delta** – Change in one or more properties
  - Permanent
  - Temporary

39

Temporality

Conceptual Model



40

Temporality
Application Issues

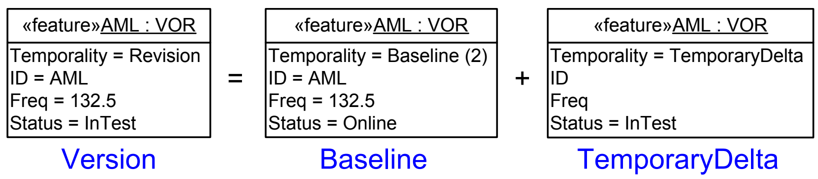
- The AIXM temporality model is complex and complete
  - Systems implementing AIXM do not need to use all temporal concepts
- Interpreting and storing temporality is a system issue
  - Some systems may only accept baselines – charting office
  - Some systems may only generate temporary changes – NOTAM office
  - Some systems may want a new version after everything change – flight plan simulation
  - Some systems may handle all temporal representations

41

Temporality

Alternative expressions

- The model supports more than one way to express a temporal concept



Other Examples

$\text{Baseline}_1 + \text{Baseline}_2 = \text{Baseline}_1 + \text{PermanentDelta}_{1\rightarrow 2}$

Temporality
Communicating Deltas

Baseline1 + Delta

Feature

Timeslice
interpretation = baseline
period of validity
property 1
property 2
property 3
property 4
property 5

Timeslice
interpretation = delta
start of validity
property 1'
property 2'

Baseline1 + Baseline2

Feature

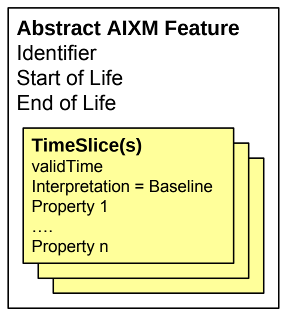
Timeslice
-*interpretation = baseline*
-period of validity
-property 1
-property 2
-property 3
-property 4
-property 5

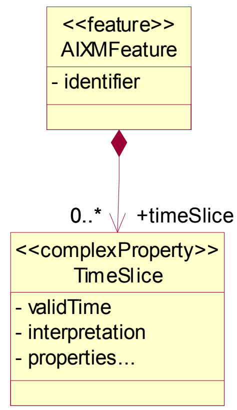
Timeslice
-*interpretation = baseline*
-start of validity
-property 1'
-property 2'
-property 3
-property 4
-property 5

43

Temporality

UML Model



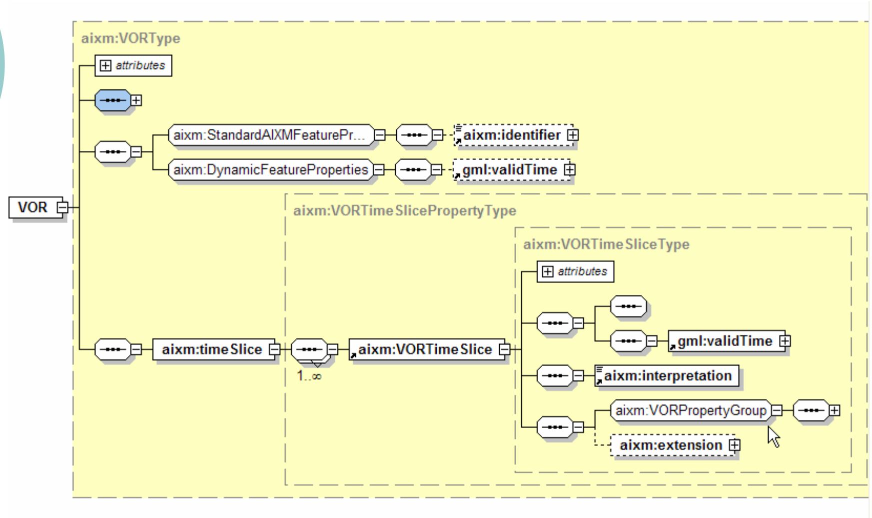


Interpretation = (Baseline, TemporaryDelta, PermanentDelta, Version, Snapshot)

44

Temporality

VOR Example



45

Temporality

VOR Example

```
<aixm:VOR gml:id="_002">
  <aixm:identifier codeSpace="www.faa.gov/ato/aim/nasr">939</aixm:identifier>
  <aixm:timeSlice>
      <aixm:VORTimeSlice>
          <gml:validTime>
              <gml:TimePeriod>
                  <gml:beginPosition>2005-10-27T00:00:00</gml:beginPosition>
                  <gml:endPosition>2005-11-23T00:00:00</gml:endPosition>
              </gml:TimePeriod>
          </gml:validTime>
          <aixm:interpretation>BASELINE</aixm:interpretation>
          <aixm:codeID>AML</aixm:codeID>
      …
      </aixm:VORTimeSlice>
  </aixm:timeSlice>
</aixm:VOR>
```

46

Temporality

Integration with Feature Relationships

- Relationships using natural identifiers need to reference
  - Identifying feature properties
  - TimeSlice interpretation
  - TimeSlice validTime

Alternative 1

RunwayDirection	**uses**	Runway where identifier = 3939

Alternative 2

RunwayDirection	**uses**	where interpretation = Baseline
	and	startPosition >= Nov 21, 2005
	and	endPosition <= Dec 21, 2005
	and	designator = 20L/02R
	and on	Aerodrome where
		interpretation = Baseline
		and startPosition >= Nov 21, 2005
		and endPosition <= Dec 21, 2005
		and codeID = MABC

47

Topics

- Feature Identification and Relationships
- Geometry
- Temporality
- Data and Message Extensibility

48

The last topic is data and message extensibility.

Data Model Extensibility

Requirements

- Feature properties
  - Add a new **power** for a VOR navaid
- Feature relationships
  - Add a new **hasEmergencyAerodrome** relationship to Airspace
- New features
  - Create a new **Aerial Refueling Track** feature

Data Model Extensibility

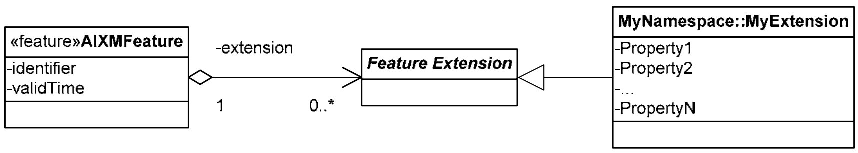
Advantages

- Increased adoption of AIXM by allowing AIXM to support more applications
  - Charting with style properties
  - Design tools with design criteria properties
- Decreased pressure on AIXM configuration control board (CCB)
  - Easy to add custom properties
  - Use CCB to globalize useful extensions

Data Model Extensibility

Model

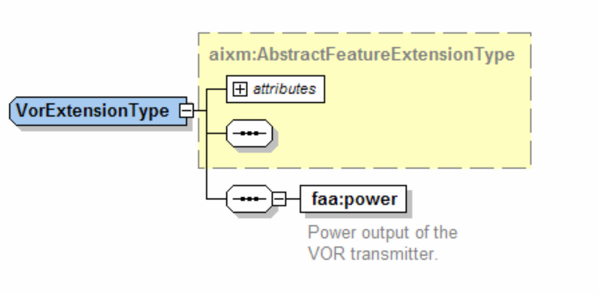
- Specialization of AbstractFeatureExtension
- Namespaced for identification
  - www.faa.gov:VORExtension
- Properties and relationships following AIXM patterns



51

Data Model Extensibility

Schema example - VOR



52

Data Model Extensibility

XML implementation

```
<aixm:VOR gml:id="B">
    <aixm:identifier>2929383</aixm:identifier>
    <gml:validTime/>
    <aixm:timeSlice>
      <aixm:VORTimeSlice>
        <gml:validTime/>
        <aixm:interpretation>BASELINE</aixm:interpretation>
        <aixm:codeID>AML</aixm:codeID>
        …
        <aixm:extension>
          <faa:VorExtension>
            <faa:power uom="kW">50</faa:Power>
          <ec:VorExtension>
        </aixm:extension>
      </aixm:VORTimeSlice>
    </aixm:timeSlice>
  </aixm:VOR>
```

Message Extensibility

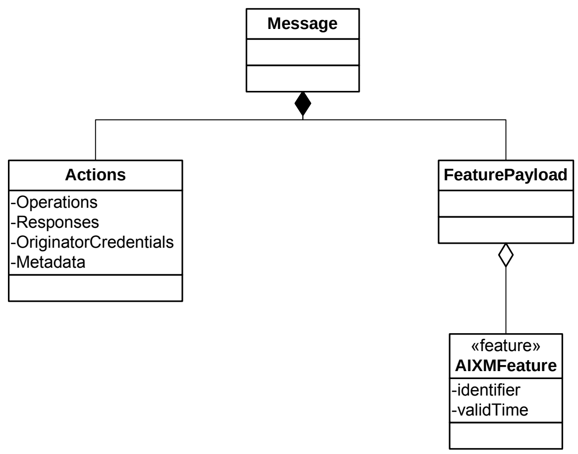
Issues

- AIXM 4.x
  - <AIXM-Snapshot> and <AIXM-Update> insufficient for global exchange
  - Based on EAD requirements for database updates
- Other messages
  - WFS (Web Feature Service)
  - xNOTAM
  - US NOTAMS
  - Procedure Design packets
  - Automated charting data packets
  - Airport Mapping

54

Message Extensibility

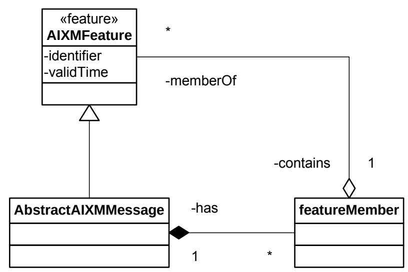
Characteristics of a Message



Message Extensibility

Characteristics of a Message

- Message requirements
  - Application specific
  - Can contain arbitrary properties
  - Contains one or more containers for aeronautical data
    - featureMember
- Exactly the GML Feature Collection



56

Message Extensibility

A Potential AIXM Message

AIXM Message

AIXM Feature Data


```
<faa:AIXMCollection gml:id="us_faa_DAFIF_1">
    <faa:cycleDate>2005-10-12</faa:cycleDate>
    <faa:coverage>ALL</faa:coverage>
    <faa:seriesNumber>102939</faa:seriesNumber>
    <aixm:featureMembers>
        <aixm:VOR>
            <aixm:timeSlice>
                <VORTimeSlice>
                    <validTime>
                        <gml:beginPosition>2005-10-12T00:00:00</gml:beginPosition>
                        <gml:endPosition>2005-12-01T00:00:00</gml:endPosition>
                    </validTime>
                    <interpretation>BASELINE</interpretation>
                    <aixm:codeID>AML<aixm:codeID>
                    <aixm:responsibilityOf hlink="us_faa_DAFIF_1#FAA"/>
                    <aixm:txtName>Armel Vortac</aixm:txtName>
                    …
                </VORTimeSlice>
            </aixm:timeSlice>
        </aixm:VOR>
        <aixm:AerodromeHeliport>
            …
        </aixm:AerodromeHeliport>
        …
    <aixm:featureMembers>
</faa:AIXMCollection>
```


57

Summary of AIXM 5 Design Recommendations

- Feature Identification and Relationships
  - Include artificial identifier
  - Use <<query>> stereotype
  - Implement with GML's xlink:href
- Geometry
  - 2 ½ D
  - Profile of GML 3.2
- Temporality
  - Versions and Deltas
  - Implement with GML's TimeSlice
- Data and Message Extensibility
  - Encapsolate property extensions
  - Use GML feature collections for messages

58