Open Geospatial Consortium

Publication Date: 2015-10-30

Approval Date: 2015-06-05

Posted Date: 2015-05-06

Reference number of this document: OGC 15-026

Reference URL for this document: http://www.opengis.net/doc/PER/t11-aviation-feature-schema

Category: Engineering Report

Editor(s): Thomas Forbes, Alberto Olivares, Richard Rombouts

# Testbed-11 Aviation Feature Schema Recommendations Engineering Report



## Warning

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements.

Document type: OGC® Engineering Report
Document subtype: NA
Document stage: Approved for public release
Document language: English

## License Agreement



 

# Contents

Page



 

# Figures

Page



# Tables

Page

## Abstract

Developed by EUROCONTROL, the Aviation Feature Schema (AFX) is a template for application schemas to implement by adding their operational attributes. For example, the Airport Mapping format can be implemented by extending AFX. The AFX defines concepts of geometry and temporality through predefined classes and properties. Therefore, these elements need not be redefined by application schemas. This means implementations of the AFX abide by the same structure, therefore aiding interoperability and allowing the rapid development of schemas. The AFX schema is designed to be generic and easily reusable and it is not intended to replace the standard aviation models such as WXXM and AIXM.

This Engineering Report assesses the suitability of the AFX as a template for lowering the GIS entry level for aviation data, providing recommendations of suitability and areas of improvement. The report is aimed at system and client developers that shall use AFX.

## Business Value

With the goal of lowering the GIS-entry level in aviation the AFX model focuses on portrayal of aviation data, including visualization and mapping of aviation related features. This Engineering Report assesses the suitability of the AFX as a model for the portrayal of aviation data, provides recommendations for improvement and reports on the overall quality of the model.

Existing aviation data exchange formats, such as AIXM, IWXXM and FIXM, focus on the system-to-system exchange of aviation data. System-to-system exchange of information introduces complexities that should not impact the end-user wanting to visualize and map the data. AFX does not seek to replace existing aviation models. Instead AFX focuses on portrayal by promoting mapping and visualization of aviation related features that are important for improved situational awareness.

## Keywords

ogcdocs, ogc documents, testbed-11, aviation, Aviation Feature Schema, AFX, GIS, AIXM, situational awareness, portrayal

 

# Testbed-11

## 1 Introduction

### 1.1 Scope

This OGC® Engineering Report assesses the suitability of the Aviation Feature Schema (AFX) for portrayal, provides recommendations for improvement and reports on the quality of the model. This Engineering Report:

- Provides a description of the AFX.
- Assesses the suitability of the AFX in a wider context.
- Reports on the quality of the AFX.
- Proposes recommendations for AFX improvements.

### 1.2 Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

| Name | Organization |
|---|---|
| Alberto Olivares | Snowflake Software |
| Richard Rombouts | Snowflake Software |
| Thomas Forbes | Snowflake Software |
| Alexis James Brooker | Snowflake Software |
| Rosie Partington | Snowflake Software |
| Johannes Echterhoff | interactive instruments GmbH |

### 1.3 Future work

Improvements to this document are desirable as the OGC community continues its efforts to advance interoperability using OGC and ISO standards within the aeronautical domain.

The observations and recommendations raised in section 8 should be addressed in follow-up activities; these include the following:

- Assessment of GML profiling techniques to restrict AFX implementations to a simple subset of the GML specification.

- Examination of the AFX temporality concept.
- Assess impact of licensing terms on AFX uptake and implementation.
- Further assessment of AFX compliance to ISO/OGC modeling and encoding rules best practice.

### 1.4 Foreword

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

## 2 References

The following documents are referenced in this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

OGC 07-036, *OpenGIS Geography Markup Language (GML) Encoding Standard*

OGC 09-025r1, *OpenGIS Web Feature Service 2.0 Interface Standard*

OGC 05-033r9, *GML Simple Features Profile*

OGC 10-100r3, *Geography Markup Language (GML) simple features profile (with Corrigendum)*

OGC 12-028, *Guidance and Profile of GML for use with Aviation Data*

OGC 06-121r3, *OGC® Web Services Common Standard*

OGC 15-024, *Aviation Guidance Using SBVR Engineering Report*

*ISO 19115:2003 Geographic Information – Metadata*

SESAR 08.01.03 ATM Information Reference Model (AIRM 2.3.1)

*AIXM – Temporality Model v1.0*, online at: http://www.aixm.aero/public/standard_page/download.html

*Aviation Feature Schema (AFX) Manual*, version 2.0.1 – draft

*Airport Mapping Exchange Schema - Primer*, version 1.1

# 3 Conventions

## 3.1 Abbreviated terms

| | |
|---|---|
| AFX | Aviation Feature Schema |
| AIRM | ATM Information Reference Model |
| AIXM | Aeronautical Information Exchange Model |
| AMCM | Airport Mapping Conceptual Model |
| AMXM | Airport Mapping Exchange Model |
| AMXS | Airport Mapping Exchange Schema |
| COI | Community Of Interest |
| COTS | Commercial Off The Shelf |
| FIXM | Flight Information Exchange Model |
| FOSS | Free and Open-Source Software |
| GIS | Geographic Information System |
| GML | Geography Markup Language |
| ICAO | International Civil Aviation Organization |
| ISO | International Organization for Standardization |
| OCL | Object Constraint Language |
| OGC | Open Geospatial Consortium |
| OMG | Object Management Group |
| OWS | OGC Web Services |
| SWIM | System Wide Information Management |
| UGAS | UML to GML Application Schema |
| UML | Unified Modeling Language |
| WXXM | Weather Information Exchange Model |
| WFS | Web Feature Service |
| XML | Extensible Markup Language |

## 3.2 UML notation

Diagrams that appear in this standard may be presented using the Unified Modeling Language (UML) static structure diagram, as described in Subclause 5.2 of [OGC 06-121r3].

 

# 4 Testbed-11 Aviation Feature Schema Recommendations – Overview

Developed by EUROCONTROL, the AFX is a template for implementing application schemas by adding their operational attributes. For example, the Airport Mapping format can be implemented by extending AFX. The AFX defines concepts of geometry and temporality through predefined classes and properties. Therefore these elements need not be redefined in application schemas that use AFX. This means implementations of the AFX abide by the same structure. This approach enhances interoperability and allows the rapid development of new schemas. The AFX schema is designed for aviation data portrayal and to be generic and easily reusable. AFX is not intended to replace the standard aviation exchange models, for instance Aeronautical Information Exchange Model (AIXM), Weather Information Exchange Model (WXXM) and Flight Information Exchange Model (FIXM).

This Engineering Report assesses the suitability of the AFX as a template for lowering the GIS entry-level for aviation data which aims to remove complexities of system-to-system exchange formats such as AIXM and WXXM for aviation data portrayal and mapping, providing recommendations of suitability and areas of improvement. The report is aimed at system and client developers that use AFX.

## 4.1 Motivation and requirements

The Aviation Feature Schema (AFX) format is designed primarily to aid reusability and to be generic. The AFX design is driven by the purpose of portrayal and mapping - not to replace existing aviation exchange standards such as WXXM, FIXM and AIXM. AFX acts as a template for application schemas to implement by adding operational attributes. The motivation for this OGC Engineering Report is to, (1) assess the suitability of the AFX schema for portrayal (mapping and visualization) of aviation data in simple GIS clients, (2) maximize interoperability of AFX, and (3) make exchange, display and manipulation as easy as possible.

The report provides recommendations and improvements for the AFX schema as well as reporting on the overall quality.

 

# 5 AFX Design Overview

## 5.1 AFX Foundation

The AFX is comprised of two components, (1) a UML conceptual model, and (2) an XML schema. By extending, using the AFX template, an application schema may inherit the structure as predefined by the classes and properties within AFX. For example, the concept of temporality and geometry are predefined within AFX. Therefore application schemas do not need to redefine these concepts. By inheriting a common structure to all AFX implementations the AFX enhances interoperability.

The AFX is based on a foundation of standards from the ISO/TC 211 series, including ISO19136[1] (GML) and ISO19115 (Metadata). (see Figure 1). GML serves as a modeling and encoding language for geographic systems as well as an open interchange format for geographic transactions on the Internet. By extending and reusing types from GML, AFX presents multiple benefits that are presented later in the document. ISO19115 (and ISO19139) is the best practice for geospatial metadata, allowing a description of the data to be provided alongside the data.

[1] The Geography Markup Language (GML), an XML grammar defined by the OGC, was adopted as an International Standard in 2007 (ISO19136). GML remains to be maintained and enhanced by the OGC.

 

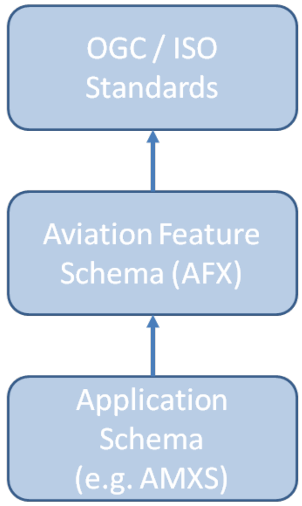


**Figure 1 AFX inheritance and implementation.**

## 5.2 AFX Components

The AFX is comprised of two components:

1. AFX conceptual model (UML).

2. AFX physical model (XSD).

These components provide a structure that should be adopted by the application schemas that implement the AFX format. The structure inherited by implementations of AFX is defined within the classes and properties within the UML and XSD.

### 5.2.1 AFX Conceptual Model (UML)

The AFX conceptual model is represented in Unified Modeling Language (UML) format.

The AFX conceptual model UML includes packages containing the foundation standards, as discussed in Section 6.1; these include the ISO/TC 211 package which provides ISO standards covering areas of digital geographic information and geomatics such as the Geography Markup Language (GML).

Three major components exist within the AFX conceptual model:

1. AviationMessage

2. AviationFeature

3. AviationFeatureState & AviationFeatureTransition

Throughout the AFX conceptual model, classes are colour-coded to represent the following:

 

- **Orange** - An AFX class.
- **Blue** - A foundation class (for example, a class inherited from GML).

#### *5.2.1.1 AviationMessage*

The *AviationMessage* component extends the GML *AbstractFeatureCollection*; this enables a collection of aviation features to be defined within a single message. The *AviationMessage* is associated with *AviationFeature* with a multiplicity of optional-to-many (‘*0..**’). This relationship enables associations to zero to many *AviationFeatures* (Figure 2).

The *AviationMessage* component provides a similar function to the *AIXMBasicMessage* component of the AIXM conceptual model.

The *AviationMessage* class also offers the option to include ISO 19115 compliant metadata in-situ within the data or externally referenced. The ISO 19139 specification provides an XML schema implementation profile derived from the ISO 19115 abstract metadata specification.

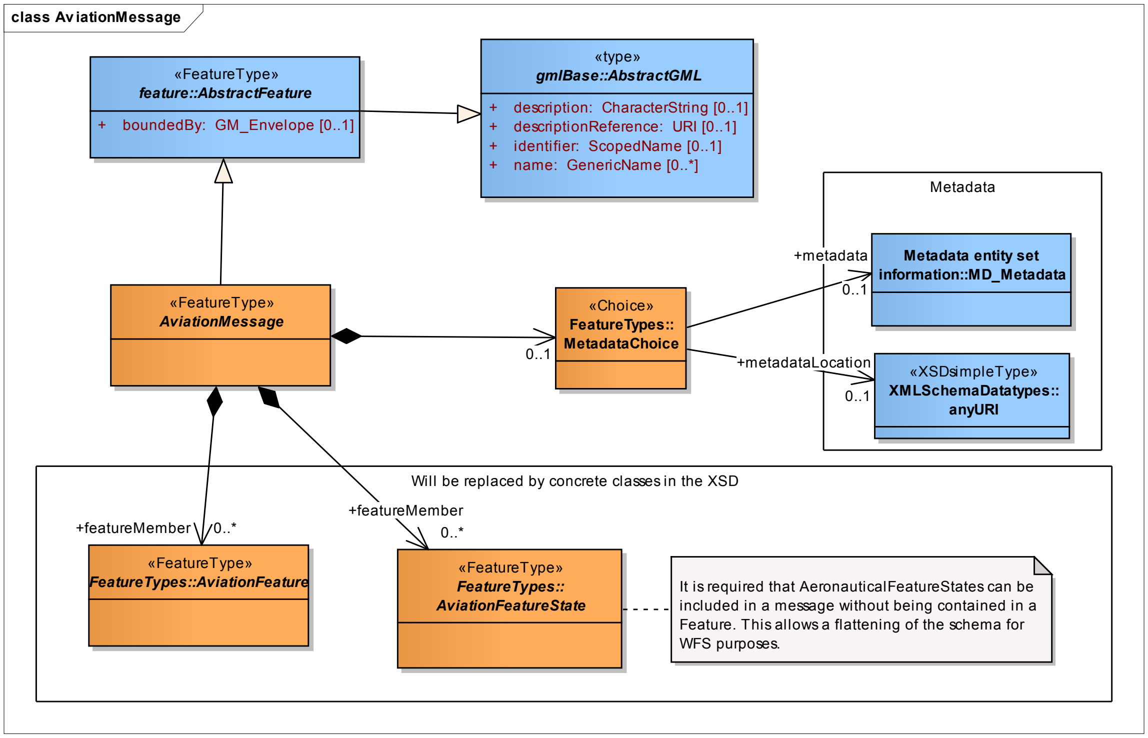


**Figure 2 *AviationMessage* class from AFX conceptual model.**

#### *5.2.1.2 AviationFeature*

The *AviationFeature* class represents an abstraction of a real world phenomenon. The *AviationFeature* class is an abstract class that is derived from 'gml:AbstractFeatureType'; application schemas implementing AFX will populate this class. For example, when implementing the Airport Mapping format by extending AFX, feature types include AM_AerodromeReferencePoint, AM_TaxiwayElement and AM_ApronElement.

The *AviationFeature* is associated with the *TM_Period* class (GML) giving the AviationFeature class a property named 'lifeTime'. The 'lifeTime' property allows the storage of the feature lifetime, for example from initial feature commissioning to feature decommissioning; this is done using a GML TimePeriod. The lifetime property will be a common property of all features extending the *AviationFeature* class.

The *AviationFeature* may contain an unlimited number of *AviationFeatureStates* and/or *AviationFeatureTransitions*; these are discussed further in the following section (see Figure 3).

Similar to the *AviationMessage* class, the *AviationFeature* class also offers the option to include ISO 19115 compliant metadata in-situ within the data or externally referenced. The ISO 19139 specification provides an XML schema implementation profile derived from the ISO 19115 abstract metadata specification.

 

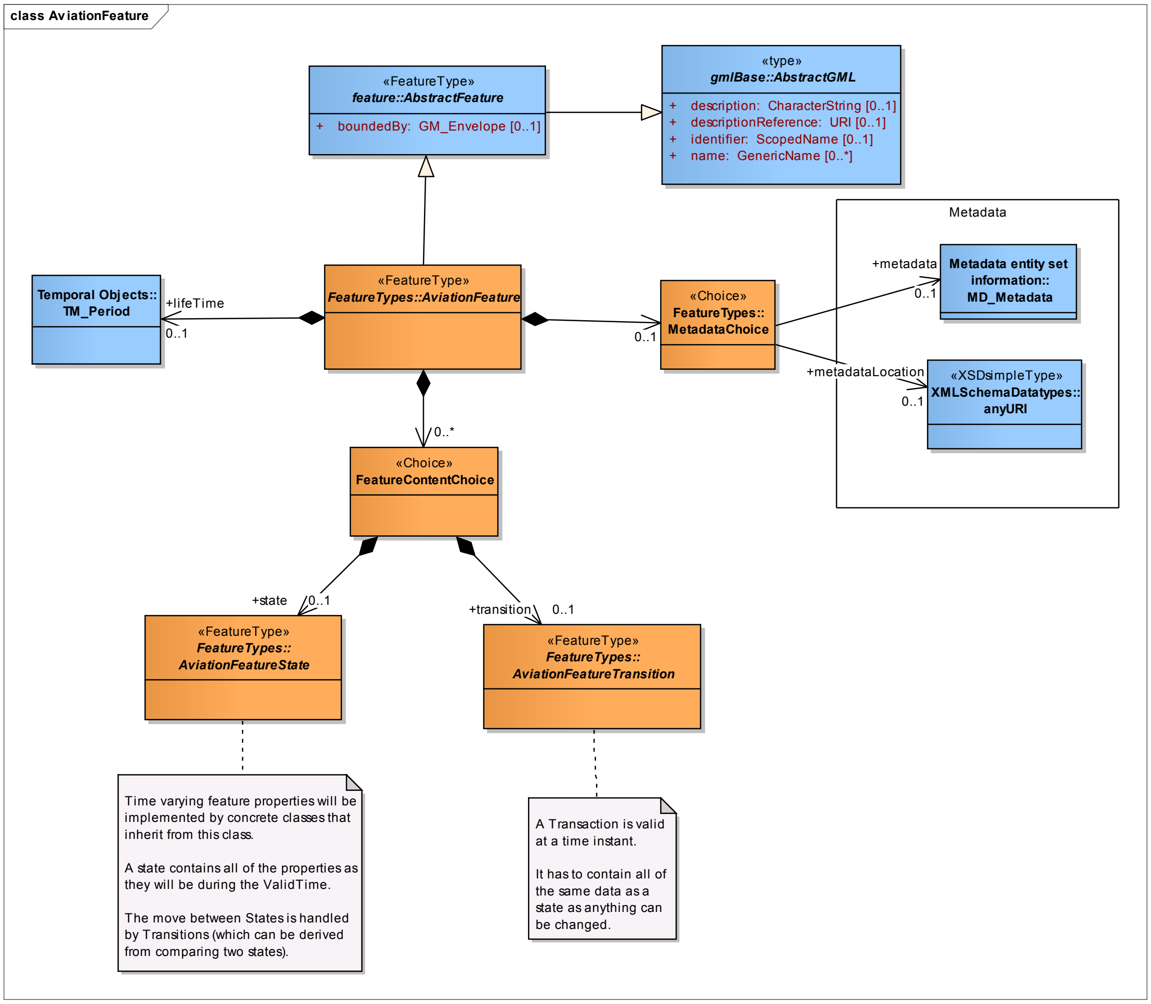


**Figure 3 *AviationFeature* class from AFX conceptual model.**

#### 5.2.1.3 AviationFeatureState and AviationFeatureTransition

Each *AviationFeature* may contain multiple *AviationFeatureStates* and/or *AviationFeatureTransitions*. Both AviationFeatureState and AviationFeatureTransition classes are abstract; the application schema implementing AFX will populate the classes as a specialization. These classes will contain time-varying feature properties, for example an aerodrome's name. An explanation of the class differences is shown below.

**AviationFeatureState**

A feature 'state' is a time period; throughout the time period of a single feature state the feature properties remain the same (see Figure 4). If a feature property changes, a new feature state is instantiated and all feature properties, including those that did not change are copied to the new feature state. These feature states do not overlap. The time instant when a feature state changes is called a 'transition', a description of feature transitions is provided later in the document.

The *AviationFeatureState* is associated with multiple properties:

1. **afx:validTime** – the GML time period for which the feature state is valid for.
2. **afx:geometry** – the GML geometry (limited to point, linestring and polygon) for the feature state.
3. **afx:limits** – upper and lower elevation limits of the feature.
4. **afx:metadata / afx:metadataLocation** – the associated ISO metadata.
5. **gml:boundedBy** – a GML envelope describing the minimum bounding box encompassing the feature.
6. **gml:identifier** – a feature identifier.
7. **gml:description / gml:descriptionReference** – feature description, or reference to feature description.
8. **gml:name** – feature name.

All *AviationFeatureState* properties are optional. This is to ease validation of adaptation and implementation of AFX. For example, an application schema implementing AFX may not be concerned with the metadata or limits of a particular feature. By being optional, the exclusion of these unwanted properties is permitted.

 

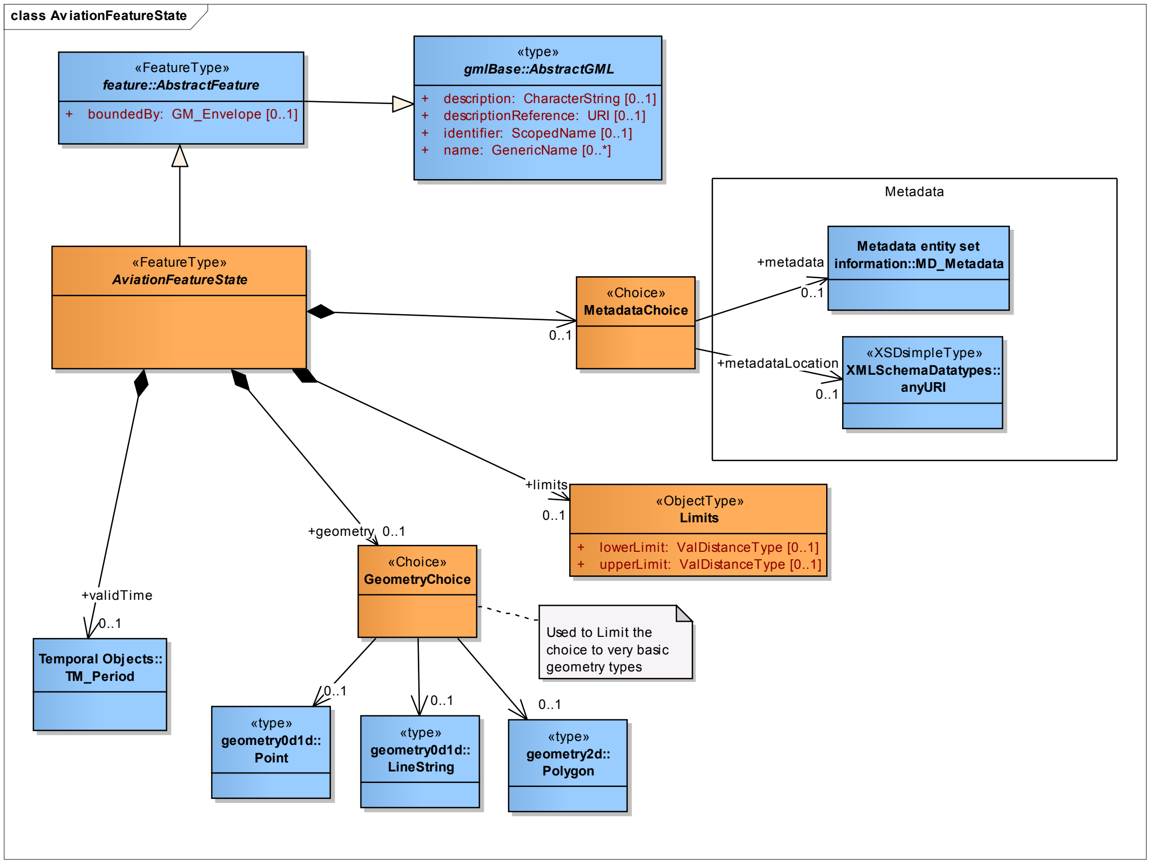


**Figure 4** ***AviationFeatureState*** **class from AFX conceptual model.**

## AviationFeatureTransition

A feature 'transition' is a time instant; a feature transition occurs when a feature state changes. The feature transition may contain all the feature properties contained within the feature state (see Figure 5).

The *AviationFeatureTransition* is associated with multiple properties:

1. **afx:transitionTime** – the GML time instant for which the feature transition is valid for.

2. **afx:geometry** – the GML geometry (limited to point, linestring and polygon) for the feature transition.

3. **afx:limits** – upper and lower elevation limits of the feature.

4. **afx:metadata / afx:metadataLocation** – the associated ISO metadata.

 

5. **gml:boundedBy** – a GML envelope describing the minimum bounding box encompassing the feature.

6. **gml:identifier** – a feature identifier.

7. **gml:description / gml:descriptionReference** – feature description, or reference to feature description.

8. **gml:name** – feature name.

All *AviationFeatureTransition* properties are optional. This is to ease validation of adaptation and implementation of AFX. For example, an application schema implementing AFX may not be concerned with the metadata or limits of a particular feature. By being optional, the exclusion of these unwanted properties is permitted.

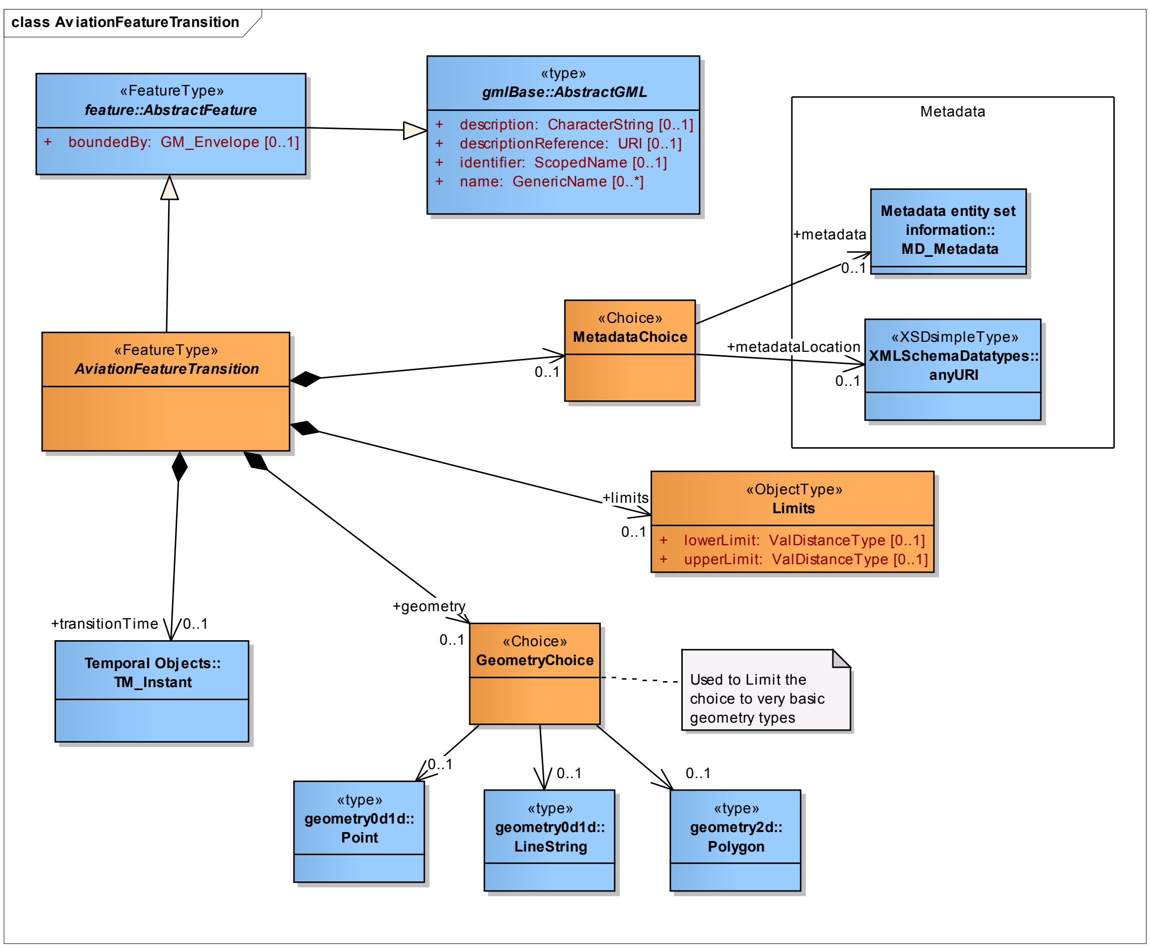


**Figure 5 *AviationFeatureTransition* class from AFX conceptual model.**

### 5.2.2 AFX Physical Model (XSD)

The *AviationFeature* and *AviationMessage* components of the AFX model have corresponding XML schemas (XSDs) based on GML. The schemas provide a physical implementation of the conceptual UML model.

To future proof the AFX XSDs two different namespaces are used for each schema. As such, the schemas could evolve independently down different paths.

#### 5.2.2.1 aviationMessage.xsd

The aviationMessage.xsd schema declares itself a GML FeatureCollection element by extending the AviationMessage element from gml:AbstractFeatureCollection. The AviationMessage element may contain any number of GML features as members. A similar function is observed within AIXM and the AIXMBasicMessage element.

The AviationMessage may also be accompanied with embedded or externally referenced metadata using the metadata and metadataLocation elements. Embedded metadata conforms to ISO 19115 Geographic Information Metadata standards.

#### 5.2.2.2 aviationFeature.xsd

The aviationFeature.xsd schema declares the structure of each feature, for example each feature may include any number of states and/or transitions, associated metadata and a feature lifetime. Each state and / or transition may (or may not) contain a validTime, geometry, limits and metadata property. These data types are defined within the aviationFeature.xsd schema.

 

# 6 AMXS AFX Implementation Example

Developed by EUROCONTROL, the Airport Mapping Exchange Schema (AMXS) is an exchange format for airport data. The AMXS is an XML schema designed to implement the Airport Mapping Exchange Model (AMXM). The AMXS defines 33 feature types for airport mapping. By extending from AFX, the following section demonstrates an example of how the AMXS can implement AFX.

The Airport Mapping format AM_Package inherits from the AFX *AviationMessage* class (Figure 6).

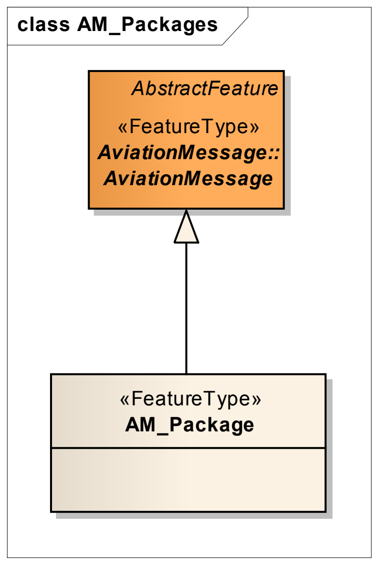


**Figure 6 Example of inheritance from Airport Mapping AM_Package to AFX AviationMessage.**

Each feature type within the Airport Mapping format inherits from the *AviationFeature* AFX class. Ultimately the Airport Mapping features inherit from the *AviationFeatureState*, various properties specific to the Airport Mapping format are added to the *AviationFeatureState* (Figure 7).

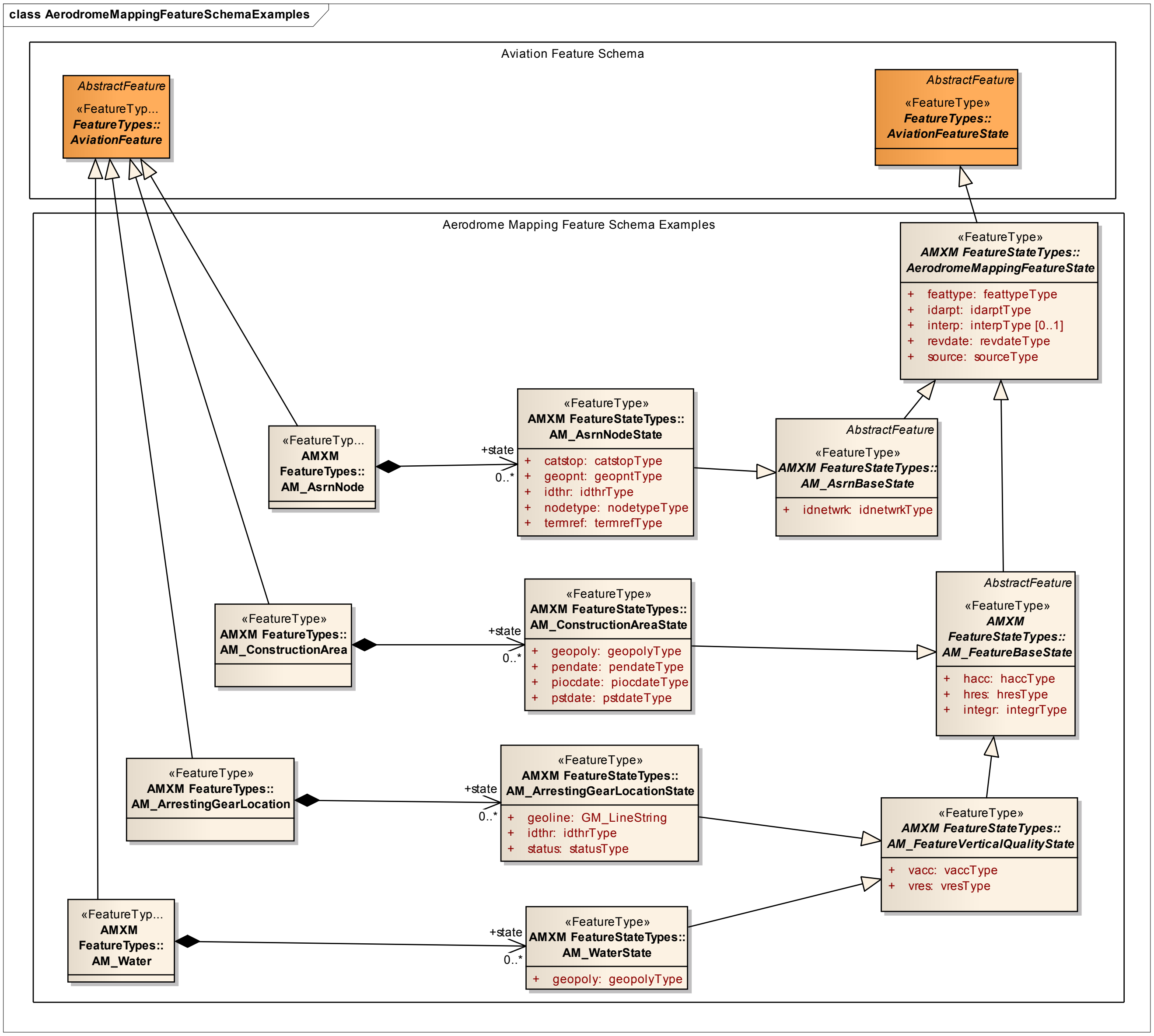


**Figure 7 Example of inheritance from Airport Mapping features to AFX *AviationFeature*.**

## 6.1 AMXS AFX Implementation XML Example

An XML sample generated from the Airport Mapping application schema when implementing AFX is shown below. An AMXS AM_TaxiwayElement feature is presented.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<amxs:AM_Package xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:afx="http://www.eurocontrol.int/im/afx"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:gmd="http://www.isotc211.org/2005/gmd"
xmlns:gco="http://www.isotc211.org/2005/gco"
xmlns:gss="http://www.isotc211.org/2005/gss"
xmlns:gts="http://www.isotc211.org/2005/gts"
xmlns:gsr="http://www.isotc211.org/2005/gsr"
xmlns:afxm="http://www.eurocontrol.int/im/afx/message"
xmlns:amxs="http://www.eurocontrol.int/aim/amxs"
xsi:schemaLocation="http://www.eurocontrol.int/im/afx
afx/xsd/aviationFeature.xsd http://www.eurocontrol.int/im/afx/message
afx/xsd/aviationMessage.xsd http://www.eurocontrol.int/aim/amxs
afx/xsd/applications/aerodromeMapping/amxs.xsd http://www.opengis.net/gml/3.2
gml/3.2.1/gml.xsd">
  <gml:featureMember>
    <amxs:AM_TaxiwayElement gml:id="LOCAL_ID_0">
      <gml:identifier codeSpace="urn:uuid:">cf10b08e-3cd0-4ac3-be6d-
ec423bac7a90_MODIFIED</gml:identifier>
      <afx:lifeTime>
        <gml:TimePeriod gml:id="LOCAL_ID_1">
          <gml:beginPosition>2015-01-01T10:00:00.000</gml:beginPosition>
          <gml:endPosition indeterminatePosition="unknown"/>
        </gml:TimePeriod>
      </afx:lifeTime>
      <afx:state>
        <amxs:AM_TaxiwayElementState gml:id="LOCAL_ID_2">
          <afx:validTime>
            <gml:TimePeriod gml:id="LOCAL_ID_3">
              <gml:beginPosition>2015-01-01T10:00:00.000</gml:beginPosition>
              <gml:endPosition indeterminatePosition="unknown"/>
            </gml:TimePeriod>
          </afx:validTime>
          <afx:geometry>
            <gml:Polygon srsName="urn:ogc:def:crs:EPSG::4326"
gml:id="LOCAL_ID_4">
              <gml:exterior>
                <gml:LinearRing>
                  <gml:posList srsDimension="2" count="28">52.374114061066535 -
31.960971595710426 52.3744611279516 -31.95616981216965 52.37453053140269 -
31.954519552332492 52.37521579457132 -31.943777975195324 52.37535133179981 -
31.941558553078725 52.376147020041884 -31.92837951247149 52.37622521459678 -
31.92756498585794 52.37654100007663 -31.922114654922687 52.376575701802174 -
31.921983559515066 52.376660528242404 -31.921929579053103 52.37784085296367 -
31.922251530702038 52.377870286949445 -31.92180984838786 52.37655754145512 -
31.921374583594957 52.37640766855823 -31.921394132233683 52.37627495351409 -
31.921509302599247 52.37619398282116 -31.92167509973242 52.37615353625479 -
31.921869815775995 52.37490876237152 -31.942638644826488 52.3745158937468 -
31.94927834993428 52.37382493754196 -31.960971595710426 52.37367432444154 -
31.96407068771583 52.374727292287155 -31.964265101088674 52.375111423433644 -
```

 

```
31.964228391727417 52.37514852880121 -31.963669143754903 52.37413010408671 -
31.963484475292006 52.37403221201539 -31.963411376762416 52.37398348915221 -
31.963265930776455 52.374114061066535 -31.960971595710426</gml:posList>
                </gml:LinearRing>
              </gml:exterior>
            </gml:Polygon>
          </afx:geometry>
          <amxs:interp>Baseline</amxs:interp>
          <amxs:feattype>taxiway</amxs:feattype>
          <amxs:source>Eurocontrol</amxs:source>
          <amxs:idlin>NORMAL</amxs:idlin>
          <amxs:idarpt>EADD</amxs:idarpt>
        </amxs:AM_TaxiwayElementState>
      </afx:state>
    </amxs:AM_TaxiwayElement>
  </gml:featureMember>
</amxs:AM_Package>
```


### 6.2 AMXS AFS Implementation WFS Example

As the AFX and application schemas that implement the AFX extend GML, the OGC/ISO Web Feature Service (WFS) interface standard can be used as a request-response interface for AFX data. The design of the AFX is founded upon the principle of interoperability and ease of use. As a result the AFX has been designed to account for limitations of WFS 1.1.0. The AFX uses a restricted set of GML 3.2 constructs and allows ISO metadata to be included either *in situ* within the messages, or referenced to an external location.

The AMXS implementation of AFX is able to be served via a WFS. An example request for an AM_TaxiwayElement feature is provided below.

```
http://[host]:[port]/[wfsName]/[servletURL]?service=wfs&version=1.1.0
&request=GetFeature&typename=AM_TaxiwayElement
```

Many GIS tools are able to connect to a WFS instance to visualize and map data responses. Figure 8 illustrates the portrayal of airport mapping information in a FOSS[2] (QGIS v2.8.1).

[2] Free and Open Source Software

 

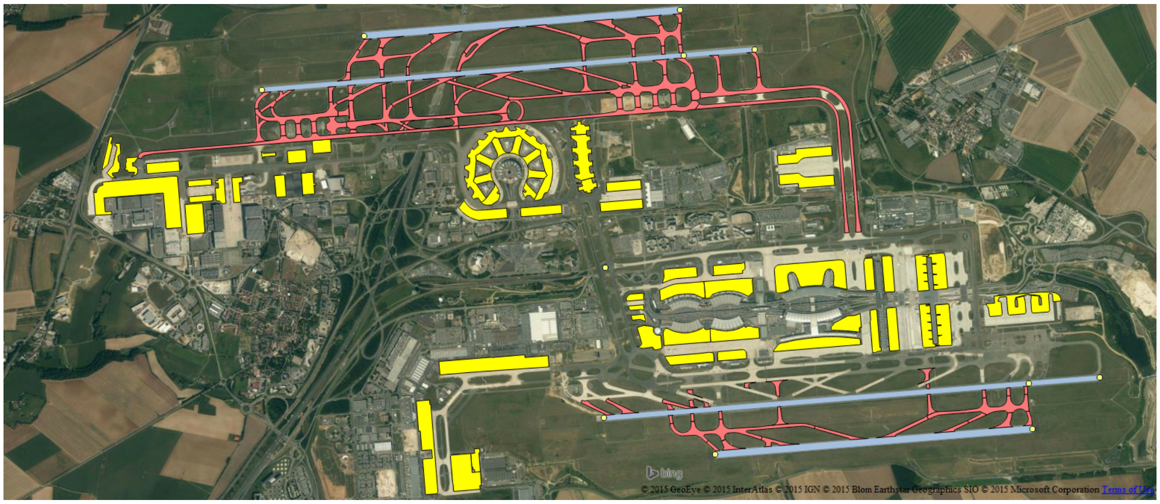

**Figure 8 Example of AMXS AFX implementation visualized in QGIS via WFS 1.1.**

# 7 AFX Assessment and Discussion

The ATM Information Reference Model (AIRM) aims to provide a conceptual blueprint for physical implementations in order to encourage semantic interoperability and common understanding. The AIRM is a consolidated logical UML model used as a common reference for civil, military and civil-military information constructs, relevant to the different domain applications within ATM currently in the SESAR development phase.

Mature aviation data exchange formats exist as physical implementations of the conceptual AIRM, for example, AIXM, WXXM and FIXM (Figure 9). These physical implementations have been designed primarily for the exchange of aeronautical, weather and flight information between systems using XML-based encodings. These aviation data models are inherently complex due to the complex and often transient real-world phenomena that they represent, coupled with the business need for system-to-system data transfer.

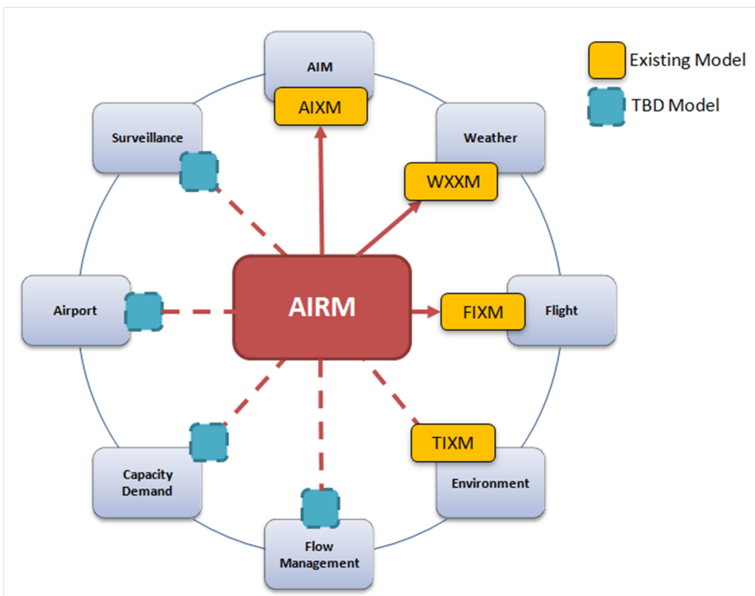


**Figure 9 Physical implementations of the AIRM for ATM application domains.**

Although these aforementioned exchange formats represent real-world phenomena similar to those that AFX implementations represent, the complex encodings of AIXM, WXXM and FIXM are suited for system-to-system data exchanges. The complexities of such encodings do not meet the business needs for aviation data portrayal and mapping; this business need has driven the development of the Aviation Feature Schema (AFX).

The AFX format is designed primarily to aid reusability and to be generic. The design is driven by the purpose of portrayal and mapping - not to replace existing aviation exchange standards such as AIXM, WXXM and FIXM. The AFX acts as a template for application schemas to implement by adding operational attributes; for example the Airport Mapping format can be implemented through extending AFX. The AFX attempts to realize ultimate interoperability by defining concepts of temporality and geometry, removing the need for the application schema that implements AFX to redefine these concepts. Ultimately the AFX promotes rapid development of schemas by allowing different applications schemas to adopt the same overarching structure, thereby aiding interoperability.

As a format for portrayal and mapping, the AFX is designed with several key principles in mind, (1) to be exchanged via web services, for example via the OGC/ISO Web Feature Service (WFS) interface standard, and (2) to lower the GIS entry-level in aviation so that custom-built tools are not required to map and portray the data, as often observed with more complex aviation models such as AIXM.

### 7.1 Compliance to GML Application Schema Encoding Rules

The Aviation Feature Schema (AFX) as well as the Aerodrome Mapping schema (AMXM), which is an application schema built on top of AFX, have been analyzed in Testbed 11 (see Figure 10).

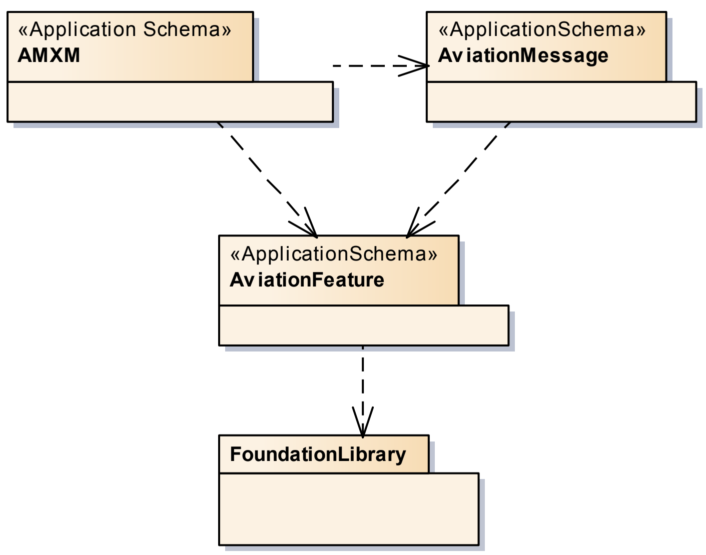


**Figure 10 Overview of the aviation feature schemas and their dependencies.**

The review of the conceptual schemas of AFX and AMXM as well as their XML implementation schemas revealed that they do not fully comply with the GML application schema encoding rules.

The schemas use stereotypes that are not defined for GML application schema, such as <<XSDsimpleType>>, <<XSDcomplexType>>, <<XSDrestriction>>, <<deprecated>>, and <<Choice>> (see Figure 11).

 

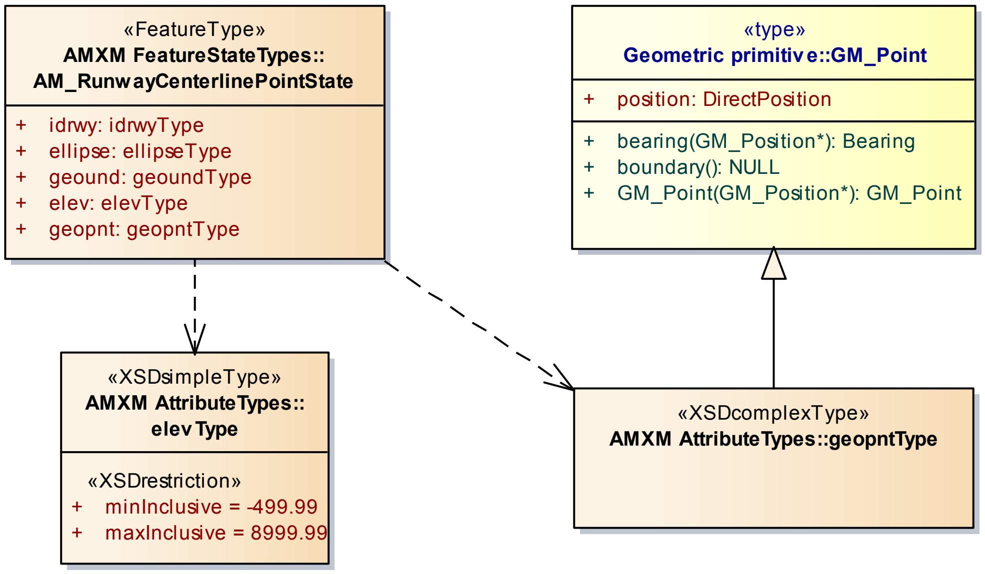


**Figure 11 XML schema specific datatypes defined in Aerodrome Mapping schema.**

Some of these stereotypes could be mapped to well-known stereotypes or ignored altogether. For example, <<Choice>> could be mapped to <<union>>.

Especially the use of the <<XSD…>> stereotypes is problematic. They are used to specify datatypes, with the value type as well as minimum and maximum value range where appropriate. This modeling approach pollutes the conceptual schema with implementation details; in this case the conceptual schema is heavily geared towards an XML Schema implementation. This is an issue whenever other implementation schemas, for example a JSON or database schema, are being derived from the conceptual schema. The issue could easily be avoided by following ISO/OGC best practices on application schema modeling. One could define <<dataType>> classes that have a "value" property of the appropriate type (e.g. "CharacterString", "Real", "Boolean", "Integer") and constraints (e.g. declared using OCL) that specify allowed value ranges.

That the ISO/OGC best practices on application schema modeling are not followed in aviation feature schemas is a general issue for all software components that rely on compliance of schemas and actual data with these best practices. This requires customization and can be problematic with respect to interoperability with ISO/OGC standards based software.

There are more examples of the aviation feature schemas not following the GML application schema encoding rules.

The conceptual model does not define the tagged value "inlineOrByReference" for the properties of a class. Application schema processing software such as ShapeChange that read the conceptual schema, for example to derive Schematron code, therefore assumes that the default defined for this tagged value applies. The default is “inlineOrByReference”[3]. However, in the AFX XML Schema the encoding is just “inline” (example: property “state” in class FeatureContentChoice). This can lead to unexpected results and even to wrong results. The issue could easily be resolved by declaring the tagged value “inlineOrByReference” in the conceptual model whenever the XML encoding deviates from the encoding that results from the default for this tagged value.

Another example is the modeling of <<Choice>> types. A <<Choice>> resembles a <<union>> type, which is covered by the GML application schema encoding rules. However, in the conceptual schema of AFX no name is assigned to the property that is of the choice/union type (see Figure 12).

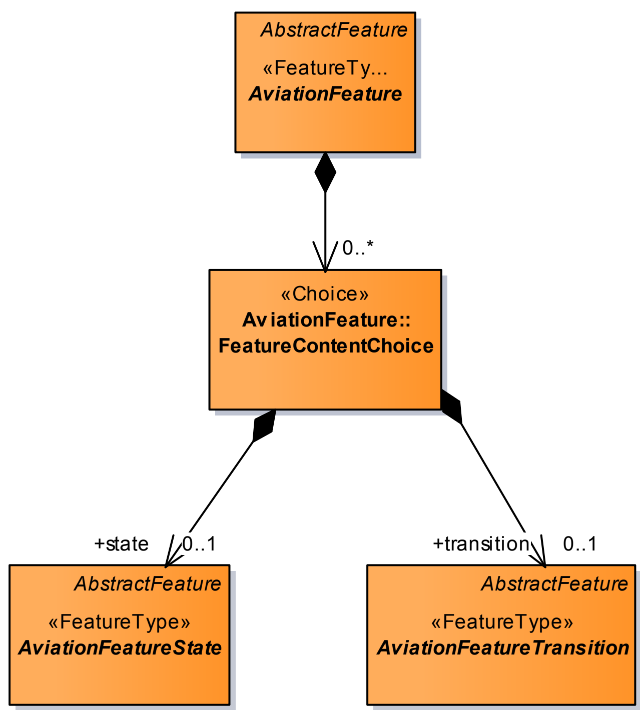


**Figure 12 Model of a choice between a state and a transition in the Aviation Feature Schema.**

This is not compliant with GML application schema encoding rules. It is even different to how <<choice>> types are modeled in AIXM[4]. From a practical perspective, this can be an issue for example when formulating SBVR constraints. Software like the SBVR-to-

[3] The other enumeration values defined for the “inlineOrByReference” tagged value are “inline” and “byReference”.

[4] The Aviation community appears to use modeling and encoding rules that are not only different to those defined by ISO/OGC but also to each other.

 

Schematron derivation tool developed in Testbed 11 (OGC 15-024) would need to be customized to support processing aviation feature schemas. Again, this issue could easily be solved by following the ISO/OGC best practices, in this case with respect to how properties and union types are modeled.

## 7.2 Developing a Format for Portrayal

A format for portrayal of aviation data could be developed using a number of methods. Firstly, a top-down approach could be undertaken, whereby a mature aviation exchange standard is adopted and repurposed for the function of portrayal. Or secondly, a bottom-up approach could be undertaken, whereby an aviation data portrayal standard is developed from scratch and restrictions are imposed via profiling. The former approach is investigated by assessing the plausibility of profiling AIXM specifically for the purpose of data portrayal. The latter approach is investigated by assessing the application of a profile to the existing AFX.

### 7.2.1 Profiling AIXM

AIXM, although dissimilar to AFX with respect to model complexities and use cases, could be used as a foundation for producing a similar result to the AFX model by using profiling methods.

By profiling AIXM, a controlled subset of the AIXM model for representing real-world aviation phenomena could be produced; profiling AIXM in this way would make interoperability and wider implementation more achievable. Profiling the AIXM model allows AIXM to be repurposed from its original use case of 'system-to-system exchanges', to 'exchange for portrayal and mapping' of aviation data. The use of AIXM profiles to simply represent aviation data is already being investigated; however in some cases new schemas have been created, for example the Simple Aeronautical Features (SAF) schema used in IWXXM to represent aeronautical features such as aerodromes and runways.

Table 1 presents advantages and disadvantages of AIXM profiling to produce a data format for aviation data portrayal and mapping.

**Table 1 Advantages and disadvantages of profiling AIXM for portrayal and mapping.**

| Advantages | Disadvantages |
|---|---|
| The same exchange model is used; therefore existing AIXM systems would support this. | The complexity of the temporality model and change management mechanism would remain. |
| A simplified AIXM model would enable wider GIS software support. | Retain the same structural encodings of concepts, but use cases are very different. |
| Multiple features types already defined within AIXM. | Problems that result in reverse associations and other extensions are inherited. |

 

Profiling AIXM would not meet the requirements of a schema for simple GIS exploitation for airport mapping.

#### 7.2.2 Profiling AFX with GML Simple Features Profile

Alternatively to profiling AIXM, AFX implementations could be subject to GML profile compliance. Currently, the implementation of AFX is flexible, allowing the end-user to add operational attributes with varying complexities. AFX implementations could be restricted to a subset of the XML-schema and GML scope which would ensure a vendor's implementations remained interoperable between clients, and usable with simple GIS tools.

The profile itself would define strict rules that adhere to common GIS modeling approaches, including the structure and types used. Numerous AFX implementations may be developed that adhere to the profile while retaining interoperability between vendors and applications.

Features must be defined using a well-known and simplified feature model that can be represented using only the GML and XML-schema components defined within the profile.

##### 7.2.2.1 GML Simple Feature Profile

Table 2 presents a comparison of the GML packages included within multiple versions of the GML specification, the GML Simple Features Profile and the AFX Implementation as presented in section 6.

**Table 2 GML packages included in GML versions and profiles.**

| GML Packages | GML 2.1 | GML 3.1 | GML 3.2 | GML Simple Feature Profile | AFX Implementation Example |
|---|---|---|---|---|---|
| Feature | ✓ | ✓ | ✓ | ✓ | ✓ |
| Feature Collection | ✓ | ✓ | ✓ | ✓ | ✓ |
| Feature Relationships | ✓ | ✓ | ✓ | ✓ | ✓ |
| Basic Geometry | ✓ | ✓ | ✓ | ✓ | ✓ |
| Primitive and Aggregate Geometries | | ✓ | ✓ | ✓ | ✓ |
| Complex Geometry | | ✓ | ✓ | | |
| Topology | | ✓ | ✓ | [5] | [5] |
| Coverages | | ✓ | ✓ | | |
| Temporal and Dynamic Features | | ✓ | ✓ | | |

[5] It is worth noting, ground-based routing would not be possible due to the lack of support for topology within the GML Simple Feature Profile, however this may not be applicable if AFX implementations are used solely for portrayal (visualization and mapping).

 

| GML Packages | GML 2.1 | GML 3.1 | GML 3.2 | GML Simple Feature Profile | AFX Implementation Example |
|---|---|---|---|---|---|
| **Coordinate Reference Systems** | | ✓ | ✓ | | |
| **Observations, values, units of measure** | | ✓ | ✓ | | |
| **Default Styling** | | ✓ | | | |

As presented by Table 2, the AFX implementation presented in section 6 uses the same GML packages as the GML Simple Features Profile; therefore the AMXS AFX implementation could adhere to the GML Simple Features Profile.

The AMXS was designed with interoperability and compatibility in mind. The AMXS design is intended to make the implementations compatible with third-party GML-compliant applications, and therefore enable end users to use the AMXS data in Geographic Information Systems (GISs).

Implementations of AFX that do not use application schemas that have been designed with portrayal and simplicity in mind, such as AIXM, will inherit complexities from those application schemas.

It is worth noting, ground-based routing would not be possible due to the lack of support for topology within the GML Simple Feature Profile, however this may not be applicable if AFX implementations are used solely for portrayal (visualization and mapping).

Three levels of compliance to the GML Simple Features Profile are defined; Table 3 presents the level compliance criteria.

**Table 3 Capabilities of Level 0, 1 and 2 of the GML Simple Features Profile.**

| | Level 0 (SF-0) | Level 1 (SF-1) | Level 2 (SF-2) |
|---|---|---|---|
| **Restricted set of built-in non-spatial property type** | Yes[1] | Yes[1] | No |
| **Restricted set of spatial property types** | Yes[2] | Yes[2] | Yes[2] |
| **Use of nillable and xsi:nil** | No | Yes | Yes |
| **User-defined property types** | No | Yes | Yes |
| **Cardinality of properties** | 0 or 1 | 0..unbounded | 0..unbounded |
| **Non-spatial property values references** | Yes[3] | Yes[3] | Yes |
| **Spatial property values references** | Yes[3] | Yes[3] | Yes |

[1] String, integer, measurement, date, real, binary, Boolean, URI.

[2] Point, Curve (LineString), Surface (Polygon), Geometry, MultiPoint, MultiCurve, MultiSurface, MultiGeometry.

[3] In level 0 and 1, remote values for properties are supported only through the use of the type gml:ReferenceType. The more generalized GML property-type pattern allowing mixed inline and by-reference encoded property values within the same instance document is disallowed.

In order to maximize interoperability and GIS entry level for aviation data, SF-0 should be followed.

Compliance level SF-0 is the most restrictive, SF-1 less restrictive and SF-2 the least restrictive whilst still abiding to the GML Simple Features Profile.

Implementations of AFX will require multiple instances of properties within a single feature due to the *AviationFeatureState* and *AviationFeatureTransition* properties; there SF-0 is not applicable.

Compliance level SF-1 does not allow the use of inline and by-reference property values within the same instance document. As AFX allows ISO metadata to be included via inline or by-reference within the same instance document, compliance level SF-1 is too restrictive for AFX implementations to adhere to.

Compliance level SF-2, essentially the OGC Simple Features Specification places no restrictions on non-spatial properties. Spatial properties remain subject to the types listed in Table 3 and the restriction on remote or referenced geometric property values observed in SF-0 and SF-1 is removed.

 

Whilst AFX implementations would require use of restricted capabilities in SF-0 and SF-1, SF-2 allows restriction to be placed on AFX implementations, but does not restrict required capabilities.

### 7.2.3 Geometry in AFX Implementations

While the *GeometryChoice* element of the AFX model restricts a feature's geometry to be encoded as a Point, LineString or Polygon, complex representations of geometries exist within other aviation data models. For example, the AIXM 5.1 model allows complex geometry types such as ArcByCenterPoint, CircleByCenterPoint and GeodesicString, see Figure 13 and Figure 14.

Complex geometry types such as those noted are unlikely to be supported out-the-box by most GIS tools and for these complex geometries to be encoded in AFX implementations, geometries must first undergo simplification, densification and/or generalization.


**Figure 13 Example of complex airspace (using ArcByCenterPoint and GeodesicString) as encoded in AIXM 5.1.**

```
…
<aixm:horizontalProjection>
    <aixm:Surface srsName="urn:ogc:def:crs:OGC:1.3:CRS84" gml:id="id_37">
        <gml:patches>
            <gml:PolygonPatch>
                <gml:exterior>
                    <gml:Ring>
                        <gml:curveMember>
                            <gml:Curve gml:id=" id_38">
                                <gml:segments>
                                    <gml:GeodesicString>
```

```
                                        <gml:pos>7.251956389 47.1243625</gml:pos>
                                        <gml:pos>7.383561389 47.133427778</gml:pos>
                                        <gml:pos>7.3835614496982975 47.13342785950983</gml:pos>
                                    </gml:GeodesicString>
                                    <gml:ArcByCenterPoint numArc="1">
                                        <gml:pos>7.328103333 47.058838056</gml:pos>
                                        <gml:radius uom="NM">5.0216</gml:radius>
                                        <gml:startAngle uom="deg">63.101621054710535</gml:startAngle>
                                        <gml:endAngle uom="deg">38.88147229538013</gml:endAngle>
                                    </gml:ArcByCenterPoint>
                                    <gml:GeodesicString>
                                        <gml:pos>7.423488784644399 47.11130883169543</gml:pos>
                                        <gml:pos>7.423488889 47.111308889</gml:pos>
                                        <gml:pos>7.587576111 46.971555278</gml:pos>
                                        <gml:pos>7.58757614775253 46.97155529814927</gml:pos>
                                    </gml:GeodesicString>
                                    <gml:ArcByCenterPoint numArc="1">
                                        <gml:pos>7.492278889 46.91922</gml:pos>
                                        <gml:radius uom="NM">5.0216</gml:radius>
                                        <gml:startAngle uom="deg">398.76127305766596</gml:startAngle>
                                        <gml:endAngle uom="deg">218.76154575464236</gml:endAngle>
                                    </gml:ArcByCenterPoint>
                                    <gml:GeodesicString>
                                        <gml:pos>7.397167536611564 46.86680474221078</gml:pos>
                                        <gml:pos>7.3971675 46.866804722</gml:pos>
                                        <gml:pos>7.232905 47.006287778</gml:pos>
                                        <gml:pos>7.232904847039378 47.006287693416944</gml:pos>
                                    </gml:GeodesicString>
                                    <gml:ArcByCenterPoint numArc="1">
                                        <gml:pos>7.328103333 47.058838056</gml:pos>
                                        <gml:radius uom="NM">5.0216</gml:radius>
                                        <gml:startAngle uom="deg">218.88133327744194</gml:startAngle>
                                        <gml:endAngle uom="deg">128.4101813384966</gml:endAngle>
                                    </gml:ArcByCenterPoint>
                                    <gml:GeodesicString>
                                        <gml:pos>7.251956511621649 47.12436239465567</gml:pos>
                                        <gml:pos>7.251956389 47.1243625</gml:pos>
                                    </gml:GeodesicString>
                                </gml:segments>
                            </gml:Curve>
                        </gml:curveMember>
                    </gml:Ring>
                </gml:exterior>
            </gml:PolygonPatch>
        </gml:patches>
        <aixm:horizontalAccuracy uom="M">5</aixm:horizontalAccuracy>
    </aixm:Surface>
</aixm:horizontalProjection>
…
```

**Figure 14 Example of complex airspace geometry as encoded in AIXM 5.1.**

AFX implementations offer the flexibility for a vendor to forgo the optional AFX defined geometry element, and to define their own geometries. Without restriction on usable geometry types a vendor may define a complex geometry type that may be incompatible simple GIS tools. By AFX implementations adhering to the GML Simple Feature Profile, AFX implementations would be restricted to the following geometry types:

- Point
- Curve (LineString)

- Surface (Polygon)
- Geometry
- Multipoint
- MultiCurve
- MultiSurface
- MultiGeometry

The benefit of restricting usable geometry types is to foster maximum interoperability of the geospatial data; support for these simple geometry types is offered by many off-the-shelf GIS tools.

The AFX implementation example using AMXS as presented in section 6 utilizes the following geometry types, (1) Point, (2) LineString, and (3) Polygon. This implementation example would therefore comply with the GML simple features profile.

#### 7.2.3.1 Generalised Geometries

When AIXM 5.1 data is rendered in a GIS display, the use case for exploitation can vary from small to very large scale. The need for densification of AIXM geometries for representation for AMXS raises the question of accuracy requirements. It could be the case that multiple representations of AMXS data be generated for varying scales to support GIS or web-app use cases.

### 7.2.4 Temporality in AFX Implementations

Many aviation data exchange models include complex representations of time; for example AIXM is built upon an exhaustive temporality model; this complexity is required due to the dynamic and often transient nature of the real-world phenomena the model represents. For example, the AIXM temporality model is capable of modeling permanent and temporary change, intrinsic change and feature states at points in time.

The AFX model defines a temporality concept that is inherited by the application schema that implements AFX. The temporality model consists of two major temporal components, (1) feature states, and (2) feature state transitions (Figure 15). The temporality model adopted by AFX describes feature states and transitions. A transition is a change in one or more feature properties. A feature state is a feature property set valid over a period of time. These temporal components are similar to AIXM 'BASELINE' and 'PERMDELTA' time slices. The AFX temporality model lacks a concept of temporary change, intrinsic change or feature states at points in time.

The differences between the complex temporality model of AIXM and the less complex representation of temporality in AFX results from the differing business needs of the models. Since AFX implementations are designed for portrayal and mapping of aviation data, a comprehensive temporality model is not required.

The AFX models the feature lifetime element outside of the temporality model (unlike AIXM) so when modeling historic data, this structure is acceptable. However, when exchanging future data, situations where the planned feature lifetime may alter. Therefore the feature lifetime element should be incorporated within the temporality model.

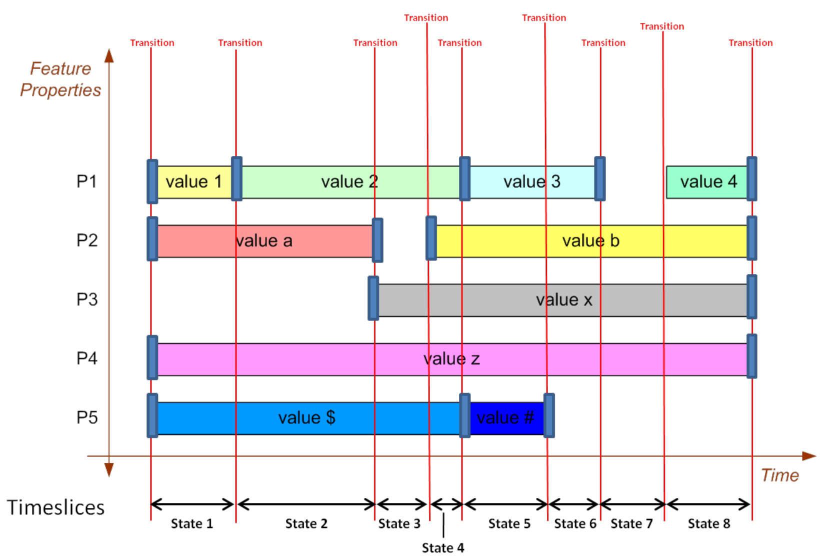


**Figure 15 AFX temporality concept – feature states and transitions.**

#### 7.2.5 Change Management

AFX does not include a method of version control or change management. For example, within AIXM's exhaustive temporality model the 'sequenceNumber' and 'correctionNumber' elements allow the vendor to submit updates and corrections to time slices.

An AFX implementation may construct a version of change management itself, by adding elements that allow a state or transition to be version controlled. For instance, similar constructs to 'sequenceNumber' and 'correctionNumber' could be instantiated within the *AviationFeatureState* and *AviationFeatureTransition* classes.

Alternatively, similar to the change management approach adopted by FIXM, the change management of feature states/transitions is the responsibility of the end-user. This change management approach is undertaken on the client-side.

 

# 8 Observations and Recommendations

## 8.1 Observations

This section provides observations based on the overall design and technical implementation analysis of the AFX as discussed in sections 5, 6 and 7. These observations are presented in Table 4.

**Table 4 AFX Observations.**

| Observation Number | Description | Impact |
|---|---|---|
| **1.** | AFX implementations do not restrict elements that can be inherited from application schema implementations. | Complex geometries, types and temporality models could be introduced to AFX implementations - these may be incompatible with COTS GIS clients. |
| **2.** | AFX is licensed under BSD license template as provided by the Open Source Initiative. | May have Intellectual Property Right implications for those implementing AFX. |
| **3.** | There is no constraint to say that there must be at least one *AviationFeatureState* or *AviationFeature* in an *AviationMessage* element. | An *AviationMessage* containing zero *AviationFeatureStates* or *AviationFeatures* would be a valid message. |
| **4.** | The *lifeTime* element is not static when considering future data, therefore should reside within the temporal model. | The lifetime of a feature is a dynamic element and may change throughout time, particularly when considering future data. This should be modeled within the *AviationFeatureState* and *AviationFeatureTransition* classes. |
| **5.** | Unlike other aviation related exchange models, for example AIXM, FIXM and WXXM, the AFX does not extend the AIRM. | For interoperability reasons it would be recommended to use the AIRM reference model to extend domain specific application schemas from. |
| **6.** | The AFX is not aligned with the Simple Aeronautical Features schema within IWXXM. | Risk of non-interoperable implementations despite the similarity of the logical entities encoded in the physical schema. |
| **7.** | Maximum GIS support would be reached through GML Simple Features Profile compliance level SF-0 adherence. | GML subset is smaller than the GML Simple Features Profile; therefore GML support by GIS client would be minimal.<br>However current, temporal and metadata requirements do not allow AFX implementations to reach the SF-0 compliance level. |
| **8.** | Geometry scales could be created depending on GIS use cases. | GIS use cases for AFX implementations need be developed. |
| **9.** | The review of the conceptual schemas of AFX and AMXM as well as their XML implementation schemas revealed that they do not fully comply with the GML application schema encoding rules. | The fact that ISO/OGC best practices on application schema modeling are not followed in aviation feature schemas is a general issue for all software components that rely on compliance of schemas and actual data with these best practices. |

| Observation Number | Description | Impact |
| --- | --- | --- |
| **10.** | The aviation feature schemas have been implemented using AFX specific modeling and encoding rules, rather than following ISO/OGC best practices. | In principle a community of interest (COI) can always define its own modeling and encoding rules. This is especially true if existing standards do not cover or are – or appear to be – incompatible with specific requirements of the COI[6]. However, this does not seem to be the case here. |

[6] The modeling and encoding of dynamic features to meet the needs of the aviation community, for example, is not yet covered by ISO/OGC standards. Thus specific rules had to be developed by the Aviation community to model and encode AIXM (even though some of the basic rules, for example to model properties and unions, could have been adopted from ISO/OGC standards).

FIXM is an example where ISO/OGC modeling and encoding rules do not appear to satisfy the requirements of the Aviation community. The results of OGC Testbed 10 show how the FIXM model and encoding could be modified to be compliant with ISO/OGC standards, with minimal impact on the actual encoding (for further details, see OGC document 14-037). Actual proof that ISO/OGC based modeling and encoding of FIXM does not satisfy the requirements has yet to be provided.

## 8.2 Recommendations

Based on the technical analysis (sections 5 and 7), observations (section 8) and implementations (section 6) of the AFX, this section provides suggested recommendations and improvements for the AFX (Table 5).

**Table 5 AFX Recommendations.**

| Recommendation Number | Recommendation |
|---|---|
| **1.** | Utilize GML profiling techniques to restrict implementations of AFX to a simple subset of the GML specification, therefore ensuring compliance with simple GIS clients. |
| **2.** | Specify a minimum occurrence of the *AviationFeatureState*, *AviationFeatureTransition* and *AviationFeature* elements.<br>By differentiating the *AviationFeatureState* and *AviationFeatureTransition* elements, it has forced a minimum occurrence of '0', however by defining a generic element with a property that defines whether the it refers to a feature state or transition would allow a minimum occurrence of '1' to be set. |
| **3.** | Move the *lifeTime* element from outside of the temporal concept to within the feature state or transition elements as the feature lifetime may alter throughout a feature's lifetime. |
| **4.** | Assess the impact of the AFX being licensed under the BSD license template as provided by the Open Source Initiative. IPR of implementations may impact uptake and implementation of AFX. |
| **5.** | The aviation feature schemas have been implemented using AFX specific modeling and encoding rules, rather than following ISO/OGC best practices. A more detailed analysis of AFX modeling requirements should be performed, to determine if any of them cannot be covered by ISO/OGC modeling and encoding rules. If it turns out that this is not the case, in other words that ISO/OGC rules satisfy all requirements, then it is recommended that future versions of AFX be implemented following these rules, to improve interoperability within the wider geospatial community. |
| **6.** | The well-established and proven encoding rules defined by ISO/OGC standards should be re-used by the aviation feature schemas. Doing so would also allow ISO/OGC standards based commercial off the shelf software to process AFX data without requiring (AFX) specific extensions and customizations. Adhering to ISO/OGC standards, in this case to the modeling principles defined by these standards, enables interoperability within the wider geospatial community. |

 

## Annex A: Revision history

| Date | Release | Editor | Primary clauses modified | Description |
|---|---|---|---|---|
| 24/02/2015 | Version 0.1 | Alberto Olivares & Richard Rombouts | All | Initial draft. |
| 30/04/2015 | Version 0.2 | Thomas Forbes & Alexis James Brooker & Rosie Partington | All | Pre-Final draft. |
| Jun 18, 2015 | NA | Carl Reed | Various | Preparation for publication |
| Jun 26, 2015 | Version 0.3 | Thomas Forbes & Richard Rombouts | Various | Minor edits after review by OGC |
| | | | | |