AICM and AIXM 5

Exchange Model goals, requirements and design

By the AIXM 5 Technical Working Group

Edition: AIXM 5
Date: February 2006

Edition No.	Issue Date	Author	Reason for Change
2005	2005	Eddy Porosnicu	First Edition
2005/10	2005/10	Brett Brunk	Converted to proposal document format. Incorporated comments from Brett, Christian and Galdos. Separated messaging versus content. Revised and applied consistent template to use cases. Provided implementation approach.
2005/10/18	2005/10/18	Brett Brunk	Incorporated Shaellyn's edits. Continued to expand implementation and use case section.
2005/11/21	2005/11/21	Brett Brunk	Expanded to cover all AIXM technical proposals.
2005/12/28	2005/12/28	Brett Brunk	Next revision based on review comments.
2006/01/30	2006/01/30	Eddy Porosnicu	Next revision, mainly based on comments from EUROCONTROL Focus Group
2006/02/01	2006/02/01	Brett Brunk	Final editing review. Removal of incomplete material.

TABLE OF CONTENTS

1 Summary

The mission of the Aeronautical Information Exchange Model (AIXM) is to provide a system independent, international data exchange standard for aeronautical data. A standard is needed because of the increasing dependence of aviation on the quality of the aeronautical information: the aviation industry, airlines, traffic management service providers and regulators are increasingly integrating their systems and modernizing from a paper product-based system to a data-driven system.

An information exchange standard is a critical enabler of aeronautical information system modernization. The standard provides:

- A common language for expressing aeronautical information for human and computer interpretation
- Cost savings through software reuse and data modelling reuse
- Increased safety through improved data integrity and timeliness
- Reduced costs of data quality checking and integration

In addition, a common standard for aeronautical information enables new products that will lead to improvements in efficiency, capacity and safety. Examples of such products include:

- Real time maps including latest information updates
- Full situational awareness using systems like Aerodrome Mapping Databases
- Automated data-driven charting
- Electronic flight bags and pilot information briefings

To enable this future, AIXM is envisioned as a major aviation related data exchange standard, designed to support aviation information collection, dissemination and transformation throughout the aeronautical data chain.

AIXM stands for Aeronautical Information Exchange Model. AIXM was developed by EUROCONTROL since 1999, as a data exchange specification based on EUROCONTROL's Aeronautical Information Conceptual Model (AICM). The two models have been primarily developed and implemented as part of the European AIS Database (EAD) Programme. The EAD has provided the first practical use of AICM and AIXM. The success of the EAD has spurred other aviation community members to consider AIXM for adoption as an international standard for aeronautical information exchange.

In preparing to adopt AIXM, a joint analysis of AIXM version 3 and 4 conducted by the United States Federal Aviation Administration and EUROCONTROL showed that AIXM sufficiently covers the scope of the aeronautical domain, but the exchange standard is too narrowly focused on a single application: data collection and dissemination from a centralized database. As a result, AIXM needs to be refactored to support a wider range of applications and aviation requirements. To that end, AIXM 5 is being designed to meet expanded requirements for aeronautical information exchange.

The purpose of this document is to outline the goals, benefits, requirements and design philosophy of AIXM 5. Major tenets of AIXM 5 include:

- Inclusion of a temporality model, including support for the temporary information contained in NOTAM

- Alignment with ISO standards for geospatial information, including the use of the Geographical Mark-up Language (GML)
- Support for the latest industry and ICAO requirements for aeronautical data including obstacles, terminal procedures and airport mapping databases
- Modular and extensible to support current and future aeronautical information messaging requirements and additional data attributing requirements.

This document is one of many documents and models used to describe and explain the AIXM 5 global standard. For more information on the AIXM 5 global standard see www.aixm.aero.

2 Introduction to AIXM 5

2.1 Mission

The goal for AIXM 5 is to provide an extensible, modular aeronautical information exchange standard that can be used to satisfy information exchange requirements for current and future aeronautical information applications. These applications include:

- Automated production of Aeronautical Information Publications (AIPs)
- Automated aeronautical chart creation and publication systems
- Integrated NOTAMs (e.g., xNOTAM)
- Aerodrome Mapping Databases (AMDBs) and related applications
- Electronic Flight Bag data requirements
- Cockpit situational displays and Flight Management System (FMS) data requirements

2.2 Scope

AIXM covers the information dissemination needs for the aeronautical domain. We follow the spirit of the ICAO definition of aeronautical data from Annex 15, which refers to the data necessary to support international air navigation [1]. From a high level point of view, it covers:

- Aerodrome/Heliport data
- Navigation Aids
- Terminal procedures
- En Route structures
- Airspace boundaries
- Air Traffic Control and NOTAM services
- Traffic restrictions
- Other data related with the above major concepts

The data publication requirements stated in the ICAO Annexes are fully supported. In addition, we intend to support a number of emerging requirements, expressed in “user requirements” documents issued by industry. The “airport mapping requirements” stated by RTCA/EUROCAE are the most prominent example.

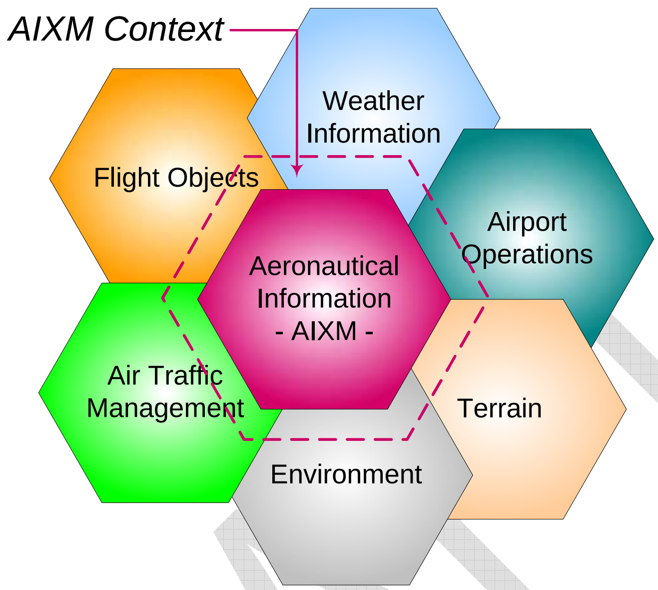
AIXM should also cover military aeronautical information, to the extent by which military data is published in Military AIPs. As long as the military specific features, attributes, list of values remain represent a small proportion (less than 5% in version 4) in the model content, it has not been considered necessary to segregate it from the rest of the model

2.3 AIXM - one of many Aviation Standards

There exist a number of related domains, which are part of the AIXM context and which are not part of the AIXM scope:

- Flights (FLT Objects)
- Weather and meteorology
- Air traffic management

- Terrain
- Airport operations
- Environmental Information



We think that the modelling approach and the set of international standards (such as the ISO 19100, the temporality model, GML) used for AIXM 5, are also applicable to the related domains, with potential positive impact on interoperability.

2.4 Background

AIXM stand for the Aeronautical Information Exchange Model. AIXM was originally developed by EUROCONTROL in 1999. AIXM originated as an XML exchange format based on EUROCONTROL's Aeronautical Information Conceptual Model (AICM).

AICM has six high-level conceptual areas:

- Airspace
- Organizations and Services
- Navigation Aids and Designated Points
- Routes including usage restrictions
- Terminal Procedures, SIDs and STARs
- Aerodromes and Heliports

The role of AICM is to enable systems to manage aeronautical information and to enable humans to communicate and understand the information that is managed. AICM describes the features, properties and relationships in the conceptual areas.

The role of AIXM is to enable systems to exchange aeronautical information in the form of XML encoded data. AIXM is an implementation of AICM as an XML schema. For a more detailed introduction to AICM and AIXM see [13].

AIXM release 3.3 has been used for several years within EUROCONTROL for collection and distribution of aeronautical data within the European AIS Database (EAD). The AIXM release 4.5 is currently being implemented in the EAD and operations based on this release will start by the end of 2006. These initial versions of AIXM focused on encoding and exchanging data identified in ICAO Annex 15, to support creating the AIP (Aeronautical Information Publication) and aeronautical charts.

Since 2002, the United States, Japan and other countries, in partnership with EUROCONTROL, have been working to enhance AIXM as a global standard for aeronautical information exchange.

2.5 AIXM 5 Model objectives

Our objective is to develop a globally applicable aeronautical information exchange standard that takes advantages of existing information engineering standards and can support current and future aeronautical information system requirements.

With the planned version 5, AIXM 4.5 will be refactored to support the latest international aeronautical data exchange requirements in the following areas:

- Expanded terminal procedures model that accommodates the latest PANS-OPS and TERPS requirements for data publication.
- Expanded obstacle model in compliance with the latest ICAO and industry requirements
- Support for static and dynamic aeronautical data (xNOTAM)
- Support for aerodrome mapping as expressed by industry data requirements

This release of AIXM is also intended to be the framework for developing standardized system to system interfaces for all aeronautical information services. Not only traditional AIS operations like aeronautical data collection and aeronautical information distribution, but also:

- Support for avionic systems updates like “electronic flight bag”
- Notice to Airmen (NOTAM)
- Aeronautical chart production
- Procedure design
- Airspace system analysis and design improvements

Finally the AIXM 5 model will incorporate the latest information engineering standards and modelling techniques including:

- Alignment with the ISO 19100 series geo-spatial standards including the ISO 19100 series temporal and geometry models.
- Use of UML (Unified Modelling Language) for the AICM 5 conceptual schema

Incorporating the latest standards into AICM and AIXM will enable adopters to make use of the latest COTS technology and tools to obtain a fast return on investment from AIXM. Some technologies that will be enabled by AICM and AIXM 5 include:

- UML modelling tools for creating software classes, relational databases and XML documents
- COTS GIS
- Web Feature Services (WFS)

2.6 Supporting legacy AIXM adopters

AIXM 5 is based on the foundation of AIXM 3.x and 4.x. In fact, approximately 80% of the conceptual schema used to build AICM 5 is based on the latest AICM 4.5 data model. An important component of the AIXM 5 design is protecting legacy investments in AIXM and providing a path for migrating systems.

2.7 Development Team

The AICM/AIXM 5 development is a joint effort by an international drafting group led by FAA and EUROCONTROL, with the support of a number of States and organizations. On the European side, France, Norway, Switzerland and Ukraine are actively supporting the EUROCONTROL Agency in this activity, through their involvement in the “AIXM 5 Focus Group”. On the United States side, FAA is working in close cooperation with NGA.

The group is open for contribution by any State or organisation that supports our goals and objectives. All deliverables, when considered sufficiently mature to be discussed in a wider group of stakeholders, are being made available for public review through the www.aixm.aero portal site.

3 Document organization and references

3.1 Document description

This document is organized into subsections as described below:

- **Section 4. Current and Future AIM information flows**
  This section provides an overview of aeronautical information management data flows as a motivation for developing an international standard for aeronautical information interchange.
- **Section 5. Version 5 Requirements and Approach**
  In this section we identify the main requirements for the aeronautical information exchange model and discuss our modeling assumptions and approach.
- **Section 6. Architecture**
  Section 6 summarizes the major architectural decisions.
- **Section 7. Requirements Analysis & Design Recommendations**
  This section analyzes information exchange requirements. We review how the requirements were implemented in earlier releases of AIXM, present a conceptual model and design satisfying each requirement.
- **Section 8. AICM and AIXM 5 Implementation**
  Describes AICM and AIXM implementation.
- **Section 9. Aeronautical information exchange use case**
  Details an aeronautical information exchange use case and shows an example of AICM and AIXM information exchange messages used to satisfy the use case information exchange requirements.
- **Section 10. AICM/AIXM Namespace Convention**
  This section summarizes AICM and AIXM Namespace conventions that support feature identification and feature relationships.
- **Section 11.** GML Introduction
  This section is an introduction to GML and GML benefits.

3.2 Definitions and Acronyms

AICM	Aeronautical Information Conceptual Model. Conceptual data model of aeronautical data. The basis for AIXM.
AIM	Aeronautical Information Management. The strategy of managing aeronautical information from origination to use in the aviation system. It is data (collection, storage, transfer) centric as opposed to the traditional product (AIP, chart) centric AIS.
AIP	Aeronautical Information Publication. An ICAO publication containing aeronautical information for a state.
AIRAC	An acronym (aeronautical information regulation and control) signifying a system aimed at advance notification based on common effective dates, of circumstances that necessitate significant changes in operating practices.
AIS	Aeronautical Information System. The system that provides aeronautical data services in support of aviation.
AIXM	Aeronautical Information Exchange Model. An XML-based representation of aeronautical data. To be used for system-

	independent, international aeronautical data exchange.
AMXM	Aerodrome Mapping Exchange Model
AMXS	Aerodrome Mapping Exchange Schema
ARINC	Aviation communications company. Author of common avionics transmission Specifications, such as the ARINC 424 format for transferring aeronautical data to Flight Management Systems (FMS) data providers. http://www.arinc.com/aviation.html
EAD	European AIS (Aeronautical Information System) Database. For centralized collection, storage and distribution of European Civil Aviation Conference (ECAC) member State AIS data.
EUROCAE	European Organization for Civil Aviation Equipment. A forum for government, industry and airlines to discuss technical problems and issue specifications. http://www.eurocae.org
EUROCONTROL	The European Organisation for the Safety of Air Navigation. http://www.eurocontrol.int
FAA	United States Federal Aviation Administration
ICAO	International Civil Aviation Organization
NAVAID	Navigation Aid.
NOTAM	Notice to Airmen. A notification of a temporary event that has caused a change in the operating conditions of the aviation system.
PANS-OPS	ICAO DOC 8168 - Procedures for Air Navigation Services – Aircraft Operations.
RTCA	United States organization that acts as a Federal Advisory Committee providing recommendations regarding national airspace system issues such as communications, navigation, and air traffic management. http://www.rtca.org
SARPS	ICAO Standards and Recommended Practices.
SID	Standard Instrument Departure. A prescribed flight procedure used to navigation an aircraft from takeoff to the boundary of the terminal environment.
STAR	Standard Instrument Arrival. A prescribed flight procedure used to navigate an aircraft from the en route environment to a terminal procedure used to land at an airport.
TERPS	Terminal Procedure rules followed by the FAA and other countries when evaluating the feasibility of terminal procedures. Considers obstacles, terrain, aircraft characteristics and procedure design to determine the safety of the procedure.
UML	Unified Modeling Language

3.3 Data modeling terminology

With version 5, AIXM and AICM are transitioning from Entity-Relationship modeling to UML modeling. In addition, we are aligning AICM and AIXM with the ISO19100 series of geospatial standards.

The following terminology is used when discussing components of the AICM and AIXM models.

Baseline	Part of the AIXM temporal model. A baseline is the state of a feature and all its properties between two permanent changes.

	Typically, baseline changes are communicated through AIP amendment or, exceptionally, through permanent NOTAM
Complex Property	An aeronautical entity that exists within the context of another feature or object. A complexProperty does not have an identifier. Examples include a TimeTable, Segment, the segments of a Navigation Aid Limitation area.
Delta	Part of the AIXM temporal model. A change to one or more properties of a feature. A delta can be a permanent change or a temporary change.
Feature	An abstraction of real world phenomena that has significance in the aeronautical domain and can be exchanged as an independent entity. Features exist independently and can be uniquely identified. Features are dynamic, meaning their properties can change in time. An example is a VOR. The VOR is a feature and has properties, such as Operational Status, that can change.
lowerCamelCase	A naming convention style in which words are concatenated. The first word is lower case and all subsequent words begin with an uppercase letter. For example “startingPoint”.
Object	An exchangeable aeronautical entity that is static. An Object exists independently and can be uniquely identified. An example is a Route Predefined Level Table containing a set of named flight levels.
Property	A feature attribute that describes and characterizes the feature. For example, a runway would have a property indicating the runway width.
Relationships	A feature property that describes an association with another feature. For example, an Airport has one or more runways. The keyword ‘has’ indicates a relationship between the Airport and its runways.
Snapshot	Part of the AIXM temporal model. The state of a feature and all its properties at a point in time.
UpperCamelCase	A naming convention style in which words are concatenated and the first letter of each word is capitalized. For example “SignificantPoint”

3.3.1 UML Modeling

A separate UML model file and document describes the AICM 5 schema in detail. This document contains snippets of UML to illustrate various architectural and designs that are fully realized in the AICM 5 UML model. This section defines the conventions used in the UML models.

We intend to evaluate ISO19103 for adoption in expressing the AICM 5 UML model. Currently, our UML modeling is based on the style used in ISO19136 (GML 3.2).

In AICM we use terms such as feature, property and relationship to describe the aeronautical data model. These concepts are illustrated in the class diagram shown in **Error! Reference source not found.**. This example shows a subset of the Aerodrome model by depicting the AerodromeHeliport and Runway features. The features have properties that describe and characterize the feature. For example, the runway has an identifier, designator, length and width properties. A relationship associates the AerdromeHeliport to the Runway. In this case the relationship shows that the

AerodromeHeliport is an aggregation of Runway features and the Runway is situatedAt an AerodromeHeliport.

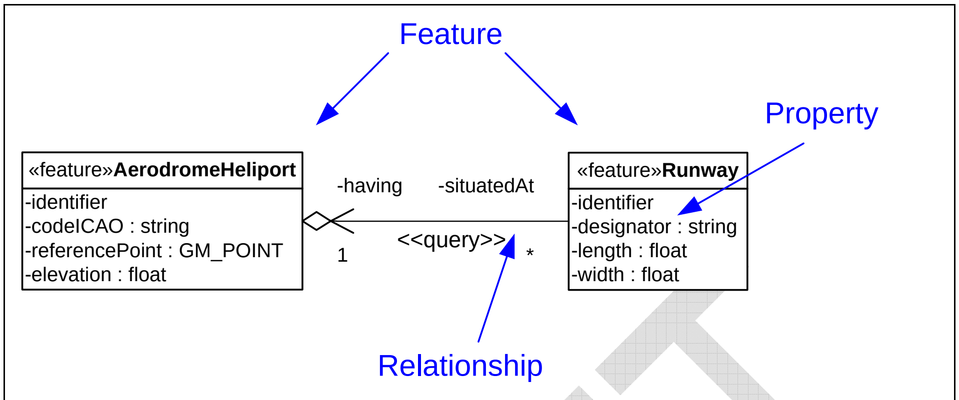


Figure 1: An example UML diagram following the AICM UML modelling convention. Major components of the model are highlighted.

3.3.1.1 Conventions for UML Classes

3.3.1.1.1 Stereotypes

In the UML we use the following stereotypes on classes:

- <<feature>> A descriptor indicating that the class represents an AIXM Feature.
- <<group>> A descriptor indicating that the class represents a collection of commonly used properties.
- <<complexProperty>> A descriptor indicating that the class represents an AIXM Complex Property.

3.3.1.1.2 Abstract Classes

In addition, some classes are abstract. Abstract classes are designated by putting the class name in italics. An abstract class cannot be realized in an implementation such as an XML document. Abstract classes are used as base classes in an inheritance hierarchy. For example, the *AIXMFeature* abstract class describes the basic properties of an AIXM Feature. Every specific AIXM Feature, such as VOR, inherits from the abstract *AIXMFeature* class.

3.3.1.1.3 Parameterized Classes

The abstract AIXM class diagram uses template classes (also known as parameterized classes) to develop the basic specifications of AIXM features and objects. The use of templates in the model will be described in more detail at their point of use.

3.3.1.2 Conventions for UML Relationships

3.3.1.2.1 Generalization Stereotypes

We use the following stereotypes on generalization relationships:

- <<extension>> - the derived class extends the inherited class. This is the normal behaviour and if a generalization relationship does not specify a stereotype, assume extension.
- <<restriction>> – The derived class restricts the inherited class.

3.3.1.2.2 Association Stereotypes

We use the following stereotypes on relationships:

- <<dynamic>> – Applies to relationships with Features. Indicates that the relationship is dynamic and should be implemented as part of the Feature Time Slice.
- <<query>> – Indicates a relationship created by query. Query relationships are dynamic. A <<query>> relationship is specified by selecting a target object by query. Executing the query returns a result set which can contain 0, 1 or more than 1 target object.
- <<static>> – Applies to relationships with Features. Indicates that the relationship is static and the relationship should not be part of the Time Slice.

3.3.1.2.3 Aggregation and Composition

- o Aggregation is a special form of association that specifies a whole-part relationship between the aggregate (whole) and a component part. The parts lifetime is not tied to the aggregate (whole). It is depicted with an empty diamond at the end of the relationship line, on the side of the aggregated class.
- o Composition is a form of aggregation with strong ownership and coincident lifetime of the parts by the whole. The part is removed when the whole is removed. It depicted with a filled diamond symbol at end of the association line, on side of the composed class.

3.3.1.2.4 Association navigability

Relationship navigability is indicated by the arrows. For the AerodromeHeliport and Runway relationship shown in **Error! Reference source not found.**, the navigability indicates that the Runway feature has a property called situatedAt that contains a query reference to the AerodromeHeliport feature.

3.4 References

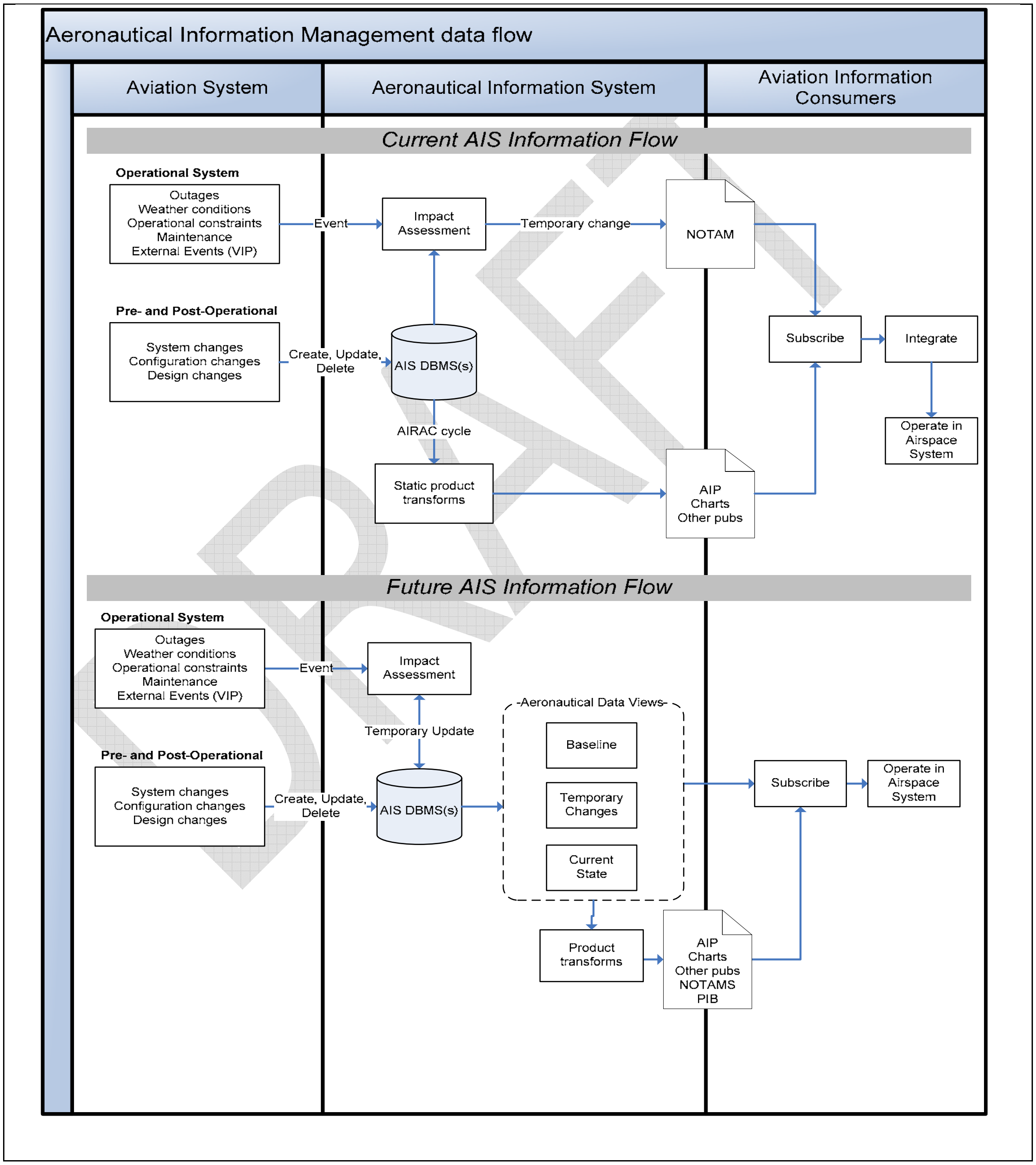
1. Annex 15 to the Convention on International Civil Aviation - Aeronautical Information Services. 12th Edition. ICAO. July 2004.
2. AICM/AIXM – Next Increment. ACCB-05/WP1/20-21/04/2005 Item 10. AIS Technical Sub-Group.
3. AIXM 4.0 – Proposed Changes. ACCB-04/WP1/8-9/09/2004 Item 6. AIS Technical Sub-Group.
4. AIXM Primer. 1.1 Edition. EATMP-021001-01. Jan. 10, 2002. EUROCONTROL. http://www.EUROCONTROL.int/ais/aixm/exchange/aixm_primer.pdf.
5. Airport Mapping Exchange Schema Version 1.1 Primer. 1.3 Edition. EUROCONTROL. September 2005.
6. “Dynamic Features” Tim Wilson and David Burggraf. September 29, 2005. Contract deliverable to FAA from Galdos Systems Inc.

7. GML: Geography Markup Language. Ron Lake, David S. Burggraf, Milan Trninic, Laurie Rae. Wiley 2004.
8. Geography Markup Language (GML). ISO/TC 211/WG 4/PT 19136 OGC GML RWG. February 2004.
9. Interchange Standards for Terrain, Obstacle and Aerodrome Mapping Data. RTCA DO-291/EUROCAE ED-119.
10. Update on EUROCONTROL Work in AIS/AIM Domain. Ken Reid. AIS Global Seminar Meeting. Narita, Japan. August 23-25, 2005.
11. User Requirements for Aerodrome Mapping Information. RTCA DO272A/EUROCAE ED-99A.
12. xNOTAM Project. EUROCONTROL. RTCA SC206 meeting. October 2005.
13. A Tour of the AIXM Concepts. Brett K. Brunk and Eduard Porosnicu. July 2004. 2004 ESRI User's Conference.
14. XML Schema Namespace Management. United States DoD Aerospace Operations. May 2003. http://diides.ncr.disa.mil/xmlreg/user/Documents/AOPNamespaceConvention.2003.05.29.pdf
15. "Object Identification." Tim Wilson. August 26, 2005. Contract deliverable to FAA from Galdos Systems Inc.
16. "19107 Geographic information – Spatial schema." ISO Project Information FACT SHEET 19107. March 23, 2003. www.isotc211.org.
17. AIXM – versioning and extensibility. ACCB-04/WP2/8-9/09/2004 Item 8. AIS Technical Sub-Group.
18. "Proposal for an AIXM profile of the Geography Markup Language (GML)." ACCB-05/WP2/20-21/04/2005 Item 11. AIS Technical Sub-Group. EUROCONTROL.
19. OpenGIS Consortium. www.opengis.org.

Requests for copies of FAA Contract Deliverables and AIM meeting references should be made to Brett.Brunk@faa.gov. EUROCONTROL documents can be found on EUROCONTROL One-Sky online (https://extranet.eurocontrol.int).

4 Current and Future AIM information flows

The key driver for aeronautical data exchange requirements is the current and future AIM (Aeronautical Information Management) data flow. This section summarizes aeronautical information flow today and how it might evolve in the future (See illustration in Figure 4).



4.1 Current AIS data flow

Today, the AI (Aeronautical Information) data flow is product-oriented and characterized by disconnects between static data updated on the AIRAC cycle and temporary changes promulgated through NOTAM. End users (AI Consumers) must subscribe to multiple aeronautical data products in order to obtain a complete view of the airspace system. AI Consumers are responsible for collecting and integrating static and temporary aeronautical data into a complete (and unfortunately sometimes contradictory) view of the operational airspace system.

4.1.1 Static aeronautical data flow

Generally today's AIM systems track static aeronautical data that is updated and published on regular AIRAC cycles. Changes to aeronautical data usually originate from pre- or post-operational system changes that result in new, deleted or updated features. For example, procedures designers may develop new terminal procedures for an Aerodrome that need to be incorporated into the static AIS data set. At the next AIRAC update cycle, the new procedures are distributed and activated

Static AIS systems employ automated and manual transformation engines to create and distribute familiar aeronautical products such as the Aeronautical Information Publication (AIP), charts and AIP Supplements. These publications are made available to AIS consumers for use when operating in the aviation system.

4.1.2 Dynamic AIS data flow

Today, dynamic AIS changes are communicated via NOTAM, AIRAC amendments and permanent NOTAM. As a result of an aviation system event, such as a facility outage, weather issues or other perturbation, one or more change notices may be issued to describe the effect of the event on the aviation system. Generally these changes are temporary changes (e.g., NOTAM) with defined start and end dates, although the start and end dates may not always be known.

NOTAM creation often involves an impact assessment in which static aeronautical data is analyzed to determine which aviation system resources are affected by the event. For example, a NAVAID outage may also affect procedures into an airport. These effects are communicated in the NOTAM, but generally no attempt is made to integrate the NOTAM changes into the aeronautical data set that is stored in the AIM database.

Aeronautical Information Consumers subscribe to NOTAM publications and must integrate the NOTAM information with the static AIS publications in order to safely and effectively operate within the airspace system. In some cases, NOTAM data flow and static data flow are provided by separate units with quite poor connections in between.

4.1.3 Current automation

Today AIS data flow automation is limited and disconnected. The AIXM data exchange standard is used successfully in regional systems such as the EAD (European AIS Database) to electronically transmit aeronautical information from States to a centralized reference database and further downstream to data integrators and end users. In addition, several static and dynamic AIS products are available digitally, for example:

- NOTAM are available on the Internet (http://tfr.faa.gov for example) or through custom distribution channels
- Many charts and publications can be obtained in HTML or PDF

However these digital products are not integrated and they generally express content in human readable formats making it difficult to incorporate into downstream computer processing.

4.2 Future AIS dataflow

In the future, AIM systems are expected to integrate static and temporary change in a more data oriented approach.

4.2.1 AIS inputs

In the future concept, aeronautical data inputs would come from static changes as well as dynamic airspace system events. Pre- and Post-operational design activities will lead to new, deleted and changed aeronautical features. Operational system events such as weather, facility maintenance and other operational constraints will be captured as events. Events may affect one or more aeronautical features. These aviation system events will be integrated with the static aeronautical data.

4.2.2 AIS data representation

Future AIM systems are expected to operate using three aeronautical temporal views:

- Baseline Data – representing the state of a feature and all its properties between two permanent changes, as communicated during regular AIRAC update cycles.
- Temporary Data – representing changes to aeronautical data that result from system events.
- Snapshot Data – representing the current state of a feature at a single point in time. The current state is the aggregation of the Baseline Data with any active Temporary Data.

Automated product transforms can be used to create traditional AIS products such as AIPs and NOTAM. Through electronic data exchange it is expected that these traditional aeronautical products will be enhanced to include “just in time” publication and product customization.

4.2.3 Future AIS data products

The future of AIM is creating integrated data repositories that enable efficient, timely and on-demand access to database driven product generation or information usage. Systems will be able to seamlessly merge static and dynamic aeronautical data updates to provide a current view of the airspace system at a single point in time or over a time interval.

AIS Consumers are expected to have access to traditional AIS products as well as direct electronic data feeds. When static and dynamic aeronautical data are merged, AIM products such as AIPs, charts and other publications can be tailored to include:

- Permanent information as reflected in AIRAC updates (Baseline)
- Permanent information overlaid with temporary information
- Actual system state at a moment in time (Snapshot)

Permanent information overlaid with temporary information might include a view of the static aeronautical data with the temporary changes highlighted. Figure 5 illustrates this approach in an example taken from the xNOTAM Proof of Concept document [12]. This drawing shows a cockpit display of an airport. A taxiway is closed by NOTAM and this is graphically displayed using the red x's.

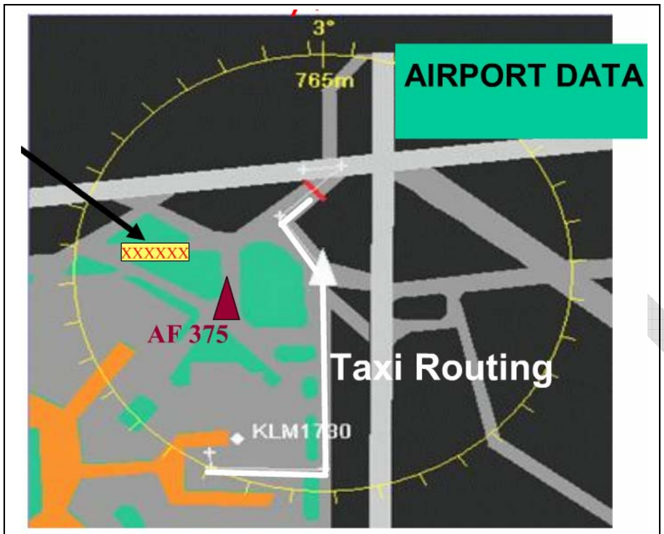


Figure 3: Notional cockpit display showing static data and dynamic data indicating a taxiway closure.

In addition to traditional products, a fully integrated AIM system will be able to support new AIS products such as Obstacle Databases and new user requirements, such as Aerodrome Mapping Databases (AMDBs).

4.3 Implications for data standardization

Mechanisms for international data exchange are crucial to realizing the future AIM system. An integrated view of the global aviation system will require advanced automated systems that can capture and integrate aeronautical data from multiple sources. These future systems can only evolve if common standards are used to represent aeronautical data. AIXM fulfills this goal by providing:

- A common international language for aeronautical data and concepts
- A computer interpretable standard for encoding, transmitting and receiving aeronautical information
- A standard mechanism to represent dynamic data.
- A framework for developing system to system interchange messages.
- A design that incorporates best practices and global standards (e.g., ISO)

5 Version 5 Requirements and Approach

The purpose of this section is to introduce the key modeling goals for AICM and AIXM 5 as well as to describe our modeling and analysis approach.

5.1 Version 5 Requirements

Since its inception AIXM has been based on a desire to develop a standard for aeronautical information dissemination. The core requirements of AIXM 3 were [3]:

- Globally applicability data exchange standard
- Compliant with ICAO SARPS (Standards and Recommended Practices)
- Support international aeronautical information exchange requirements of the EUROCONTROL states

Several states, agencies and companies outside the EUROCONTROL showed interest in adopting AIXM and enhancing AIXM to support use cases beyond AIXM's original intent. As a result, the requirements for AIXM 4.x were expanded to include [2]:

- International aeronautical information dissemination of the stakeholder States
- Support a wide spectrum of aeronautical information services applications
- Protect legacy investments by providing a full backwards compatibility model

Further analysis showed that developing an aeronautical data model that fully supports global needs for international aeronautical information dissemination as well as incorporating contemporary international data modeling standards will require refactoring the AIXM model. To that end, AIXM 5 was envisioned as a major release incorporating the following requirements [2]:

- Full coverage of aeronautical domain data content
- Feature identification and relationships
- Geometry
- Temporality
- Extensible data model
- Extensible exchange message framework
- Modularity
- Alignment with ISO geo-spatial standards

5.2 AICM and AIXM

There is often confusion about the relationship between AICM and AIXM. Most often, new AIXM adopters focus on the exchange language, AIXM. At other times both terms are used interchangeably.

We make a distinction between AICM and AIXM. AICM is meant to be an implementation-independent conceptual model for the aeronautical domain that tries to capture aeronautical information supporting international air navigation. The role of AICM is to enable systems to manage aeronautical information and to enable humans to communicate and understand the information that is managed. For version 5, AICM is represented as an object model using UML.

AIXM 5 is the XML exchange format based on AICM. In other words AIXM 5 implements AICM 5 as an XML schema. The role of AIXM is to enable systems to exchange aeronautical information in the form of XML encoded data.

This document discusses the goals, requirements and design elements for both AICM 5 and AIXM 5.

5.3 Approach

5.3.1 Modeling guidelines

Earlier in this document the AICM/AIXM modeling domain was broadly defined based on ICAO Annex 15 as all aeronautical information required to support international air navigation [1].

We interpret this requirement to mean that the aeronautical domain covers the following features:

- All physical entities that aid in aircraft movements from departure point to arrival point.
    - Aerodrome layout support aircraft movement, takeoff and landing
    - Navigation aids
    - Lighting, landing systems and other physical infrastructure that aids in navigation.
- All physical entities that have to be taken into consideration in order to protect the movement of aircraft in the air
    - Obstacle, both natural and man-made
- All conceptual entities that enable aircraft to operate in the air traffic system
    - Fixes, waypoints and other designated points
    - Routes and procedures
    - Airspace
    - Flow restriction rules
- All organizations, units and services that provide for the provisioning of air navigation
    - Air traffic control facilities
    - Pilot briefing services, NOTAM offices and flight information services

These features have properties that characterize the features and there are relationships that link many of these features with each other. Often the properties and relations of a feature are tied to a specific implementation or system. For example, a charting system may be interested in properties such as feature styling (color, line thickness, etc) that are irrelevant for flight planning systems. Besides charting, other applications include:

- Design activities
- Avionic system updates
- Flight planning and simulation
- Data collection and validation
- Facility maintenance
- Air navigation / Air traffic control

Each of these application areas can have its own view on the properties and relationships of a feature. Many of the properties and relationships used in these specific systems may not be good candidates for the exchange model because the applications are country or system specific.

In AICM and AIXM our focus is on feature properties and relationships that directly support aircraft navigation in the air and on the ground and thus have relevance to pilots, aircraft and air traffic control. Consequently we try to characterize properties that indicate how the feature is used in air navigation. Air navigation is the fundamental purpose of aeronautical data and by focusing the AICM and AIXM model on air navigation it is our intent to provide a generally applicable model that can be readily adapted to support more specialized applications and systems.

Relationships are used to connect features to each other. For instance a Runway is on an Aerodrome and a Route Segment startsAt a Significant Point and endsAt a Significant point. Like properties, it is important to set boundaries on the types of relationships that will be supported in the model.

We use the term *derived relationship* to indicate a relationship that can be calculate or obtained by using more fundamental relationships. For example, a route segment may cross and airspace border. The relationship between the route segment and airspace border is geometric and it could be represented explicitly in the data model. In this example, most would probably say that this relationship does not belong in the model – instead it is derived through a spatial operation. In other situations, characterizing a relationship as fundamental or derived is more difficult to determine. A VOR can be collated with a DME. Is that a derived relationship or a fundamental relationship? In AICM 4.x the relationship is modeled and considered fundamental to AICM, because collocation implies more (frequency pairing, same id) than a simple geographical vicinity.

In version 5 we have tried to avoid modeling derived relationships especially when these relationships cross conceptual areas.

Finally we note that version 5 is based on AICM 4.5 and AIXM 4.5. As a result, version 5 inherits much of its modeling style from previous releases. In some cases the guidelines defined above may have not been followed in the past. In these situations we have decided to keep the legacy model unless the area is undergoing significant refactoring. It is our intent to review these modeling issues in future incremental releases of AICM/AIXM 5.

5.3.2 Alignment with ISO Geo-spatial standards

The ISO Technical Committee 211 (TC211) is responsible for developing the ISO 19100 geographic series of standards. The purpose of these standards is to provide a common framework for developing domain specific standards based on geography. The ISO TC211 standards include temporal, metadata, and spatial schemas. In addition, the major GIS (Geographical Information System) XML standard, GML (Geography Markup Language) is scheduled to be integrated into the TC211 standards by 2007.

To the extent feasible, AIXM should use established standards for data modeling, geographic representation, temporality and metadata. By aligning with the ISO 19100 series, AIXM gains the following benefits:

- Increased global interoperability with other data standards

- Leverages analysis and design decisions developed by the ISO TC211 committee
- Support standard representations for common constructs such as temporality and geometry.
- Increased credibility by demonstrating that AIXM is incorporating international standards.
- Leverages COTS tools and products that are also based on the ISO standards.

5.3.3 Methodology

In our analysis and design work we applied the following methodology:

1. Evaluate how well the current AIXM 4.x data exchange specification meets the requirements and identify any short coming that need to be addressed in the AIXM 5 specifications.
2. Review industry and international standards to determine if there is a standard that satisfies the requirements.
    - o If a standard can be found that meets most of the requirement then adopt the standard and adapt requirements to be compliant with the standard
    - o If a standard exists but is inadequate then adapt or extend the standard and work to influence the standards organization.
3. Develop or adapt a model that satisfies the AIXM exchange specification requirements.
4. Implement the requirements at the conceptual level and at the XML exchange specification level.

6 Architecture

The purpose of the model architecture is to present a framework for AICM and AIXM 5.0 so we can ensure that AICM and AIXM 5 are robust enough to support aeronautical information exchange for the long term.

In this section we summarize major design decisions as a series of architectural views:

- **ISO 19100 Series view** – AICM and AIXM within the ISO19100's framework
- **AICM / AIXM Framework view** – The AIXM framework and how it enables system to system information interchange.
- **Implementation view -** Relationship between AICM and AIXM and how legacy investments in AIXM 4.x will be protected and migrated to version 5.
- **Aeronautical Domain view** – Model organization based on loosely coupled subject areas within the aeronautical domain
- **Data modeling compartment view** – separation of data modeling components to simplify versioning and extensibility

6.1 ISO 19100 Series view

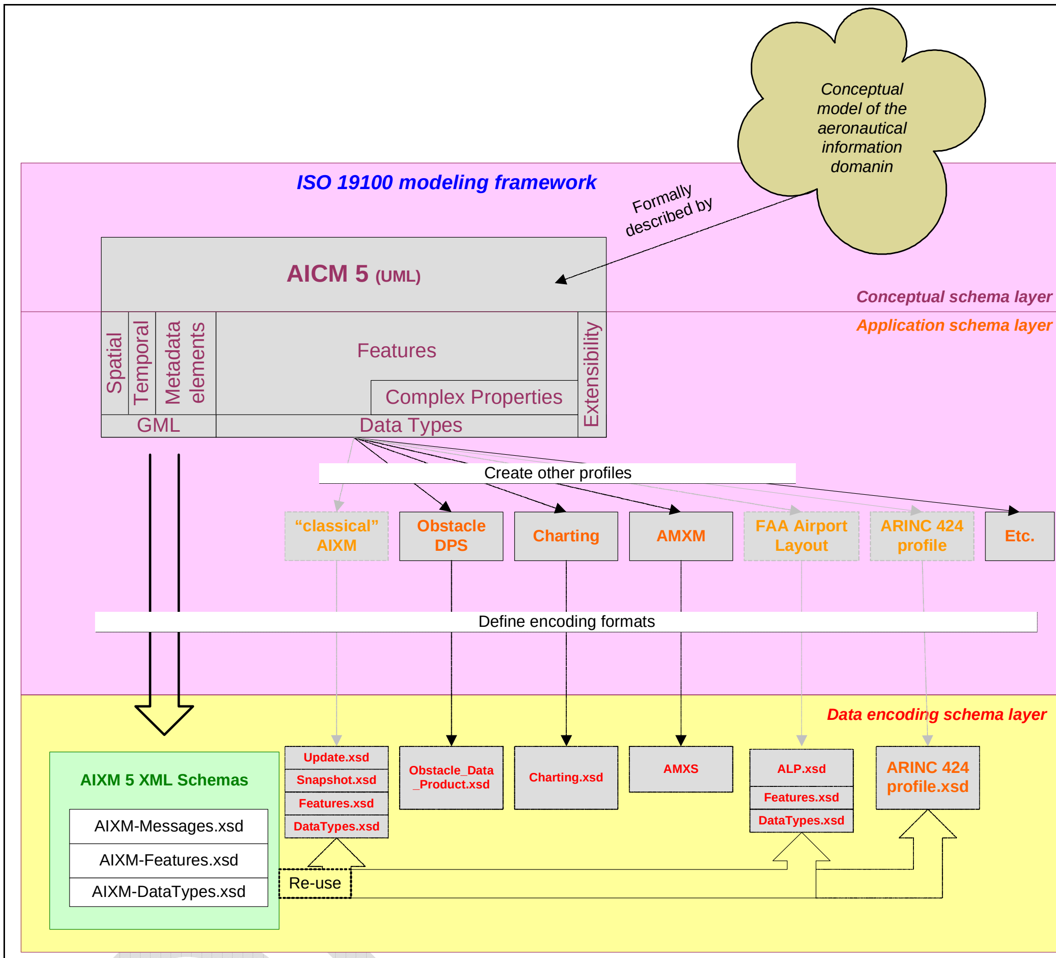


Figure 4: AICM and AIXM within the ISO 19100 Series reference model

Figure 6 shows how AICM and AIXM fit within the ISO 19100 series reference model.

The Application Schema Layer is where the domain model is constructed. ISO 19100 series includes a set of model frameworks to help build application schemas. These include spatial, temporal, and metadata application schemas as well as a general feature model. These modeling frameworks are used to build the AICM 5 conceptual schema in UML and the AIXM 5 data encoding specification in XML. The AIXM exchange schema implements AICM using GML. GML 3.2 is an XML realization of the ISO19100 series standards that is expected to be published in 2007.

One component of the ISO 19100 modeling framework consists of Data Product Specifications. Data Products define data sets that are used in systems. Examples of data products might include:

- **AMDB** – Aerodrome Mapping Databases
- **Airport Layouts** – Full CAD details of an airport.
- **AIXM 4.5 Snapshot/Update** – Data product specifications, as currently used in collection and dissemination of aeronautical data through the EAD.

- **ARINC 424** – a data product specification for transferring aeronautical data to flight management systems data providers.
- **xNOTAM** – a data product specification for communicating notices of temporary changes to the aviation system.

It should be noted that from the above examples, currently, only the AMDB is supported by a Data Product Specification (AMXM) that complies with the ISO 19100 modeling framework.

6.2 AICM / AIXM Framework view

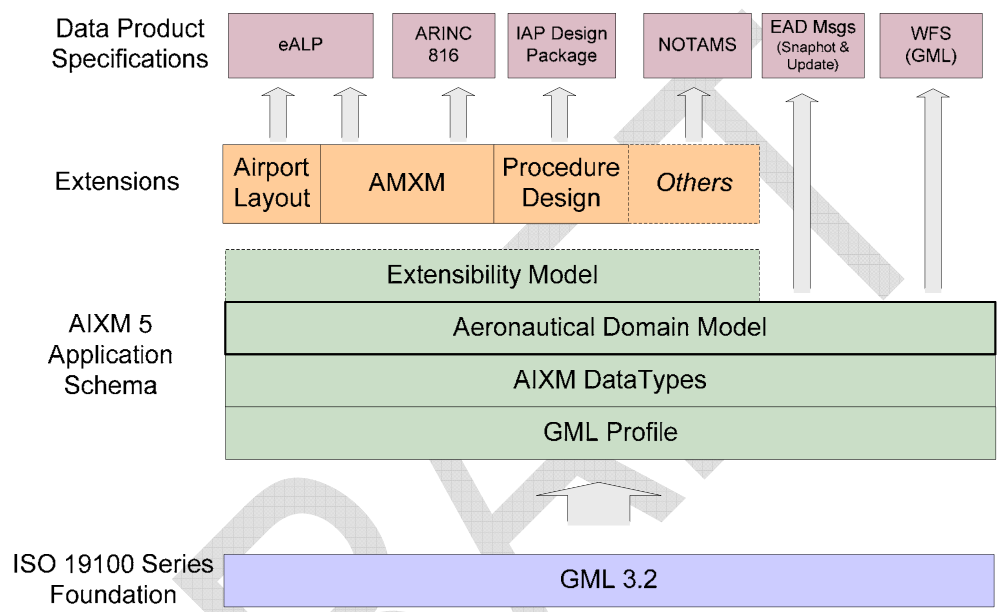


Figure 5: AICM/AIXM Architecture

Building upon the underlying ISO 19100 series foundation, AICM and AIXM have been developed as a framework for supporting aeronautical information exchange. The framework includes:

- A profile of GML that specifies the features of GML that are incorporated into AIXM.
- Common set of data types and values domains used for aeronautical data.
- Core aeronautical data model supporting international air navigation
- Provisions for extending the AIXM data model to support additional applications by allowing for new properties, relationships and messages.

Using the framework, communities of interest can develop extensions to support specialized requirements. Example extensions might include:

- Airport Layout
- Procedures Design

At the highest level, systems take the core AIXM 5 features along with extensions to develop a series of data products or messages that will be used for information exchange.

6.3 Implementation view

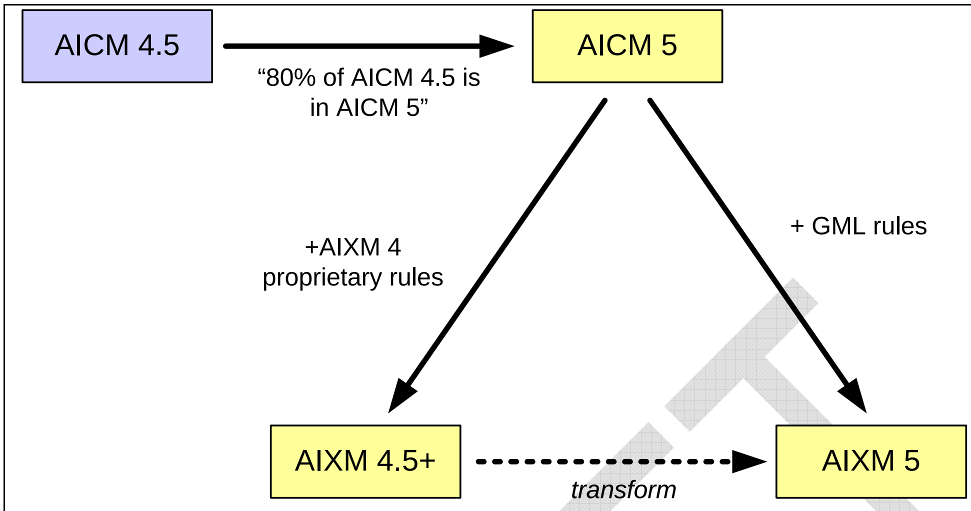


Figure 6: Implementation view (Supporting Legacy Systems)

As much as possible AICM/AIXM 5 should preserve legacy investments and provide a path for systems migrating from AIXM 3.x and AIXM 4.x.

The cost of transiting from legacy AIXM to AIXM 5 will be reduced by basing AICM/AIXM 5 on the AICM 4.5 data model. It is expected that roughly 80% of the current AICM 4.5 model will be carried over and implemented as part of AICM 5. Most of the major design changes such as alignment with the ISO standards, incorporating temporality and modelling AICM in UML should have little effect on the content of the conceptual model.

AICM 5 will then become the basis for the AIXM 5 exchange model by applying GML application schema rules to generate AIXM 5. At the same time it is possible to transform AICM 5 using the proprietary AIXM 4.x encoding rules to create an AIXM 4.5+ that has the same XML style of AIXM 4.5 but also incorporates changes to the aeronautical conceptual areas.

Since both AIXM 4.5+ and AIXM 5 derive from the same conceptual model one could encode the same feature content in both exchange models (Of course, AIXM 4.5+ would remain limited to the <AIXM-Snapshot> and <AIXM-Update> messages only and would not be able to leverage GML capabilities, temporality and other AIXM 5 advanced features). In fact, it should be possible to create a one way transformation that converts AIXM 4.5+ XML into AIXM 5 XML.

6.4 Aeronautical domain view

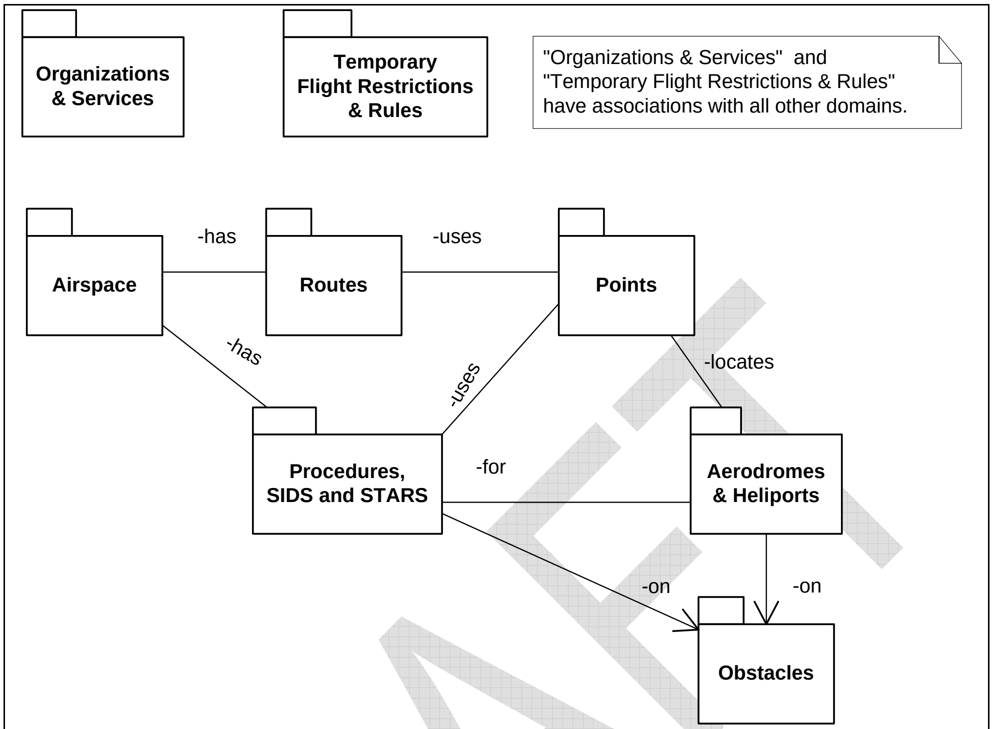


Figure 7: Aeronautical domain view

The Domain View shows how the AICM/AIXM 5 model is subdivided into loosely coupled concept areas. In the diagram the associations are meant to represent connections between the areas. In AIXM there are eight major conceptual areas:

- **Organization & Services.** Describes organization authorities and units responsible for other aeronautical facilities and systems. Describes services provided by aeronautical assets such as Air Traffic Control services provided in Air Traffic Control Sector airspace. Organization & Services has associations to all other concept areas.
- **Temporary Flight Restrictions & Rules.** Encodes flight plan restrictions that limit aircraft access to the airspace system in support of traffic flow management strategies. Temporary Flight Restrictions & Rules has associations to all other concept areas.
- **Airspace**. Defines 2.5D volumes of airspace of different types. Airspace is also used to define protection areas for Routes, Procedures, SIDs and STARS.
- **Routes**. Defines the en-route route structure. Routes use significant points as the start and end of each route segment.
- **Significant point.** Includes navigation aids, fixes and waypoints used to define the trajectory of an aircraft in flight or as aid in navigation.
- **Procedures**, SIDs and STARs. Routes used for arrival, landing and departure at an aerodrome or heliport.
- **Aerodrome & Heliport.** Defines aerodrome and heliport layout and facilities, navigation services and access restrictions.

- **Obstacles**. Defines natural and man-made obstacles and includes associations as controlling obstacles on terminal procedures and as significant obstacles on an aerodrome/heliport.

6.5 Data modeling compartment view

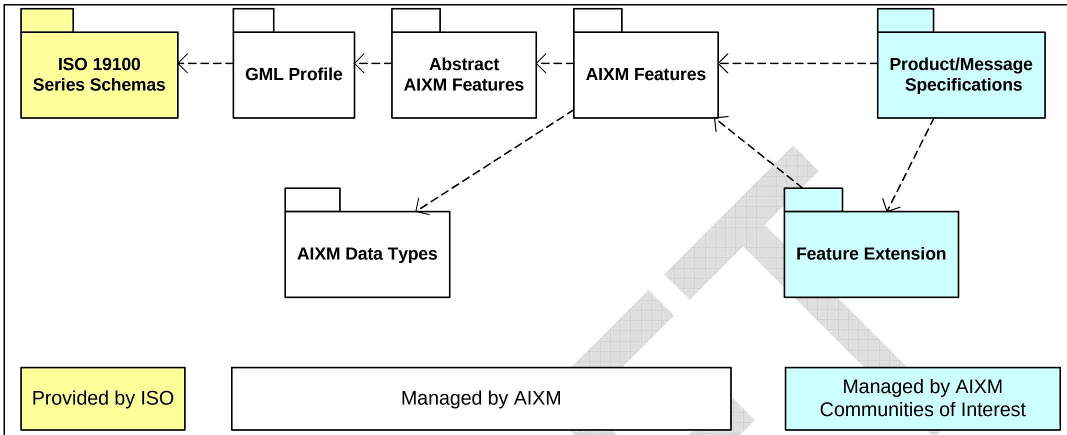


Figure 8: Data modelling compartment view

The data modeling compartment view shows the AICM/AIXM 5 specification broken into information units to make it easier to configuration manage and version. The yellow ISO 19100 Series Schemas are provided by ISO and is not configuration managed by AIXM. The white packages are under AIXM configuration management and these include:

- **GML Profile**. The subset of GML features implemented in AIXM 5.0
- **Abstract AIXM Features**. Establishes the basic AIXM 5.0 feature model including extensions to the GML Profile supporting AIXM 5.0 requirements and anticipated GML 3.2 changes.
- **AIXM Features**. Definition of all AIXM feature types.
- **AIXM Data Types**. Value domains and enumerations of AIXM Feature properties.

The blue packages are developed by communities implementing the AIXM 5.0 specification. These include:

- **Feature Extensions.** Optional additional feature properties and relationships that are added into AIXM
- **Product/Message Specification.** Defined aeronautical interchange messages based on the extensible messaging framework specified by AIXM 5.

7 Requirements Analysis & Design Recommendations

In this section we present our analysis and design decision for each of the AICM/AIXM 6 main model requirements:

- Full coverage of the aeronautical domain
- Feature identification and relationships
- Geometry
- Temporality
- Extensible data model
- Extensible exchange message framework

7.1 Full coverage of aeronautical domain

AIXM is intended to support AIS data requirements by covering the data needs of the aeronautical domain as described in ICAO Annex 15 [1]. According to Annex 15:

> *The object of the aeronautical information service is to ensure the flow of information/data necessary for the safety, regularity and efficiency of international air navigation. The role and importance of aeronautical information/data changed significantly with the implementation of area navigation (RNAV), required navigation performance (RNP) and airborne computer-based navigation systems. Corrupt or erroneous aeronautical information/data can potentially affect the safety of air navigation.*
>
> *To satisfy the uniformity and consistency in the provision of aeronautical information/data that is required for the operational use by computer-based navigation systems, States shall, as far as practicable, avoid standards and procedures other than those established for international use.*

EUROCONTROL originally developed AICM and AIXM to support aeronautical data collection and data product harmonization in the European states. EUROCONTROL based the AICM and AIXM 4 data model on:

- ICAO (International Civil Aviation Organization) standards and recommended practices (SARPS)
- Data concepts contained in Aeronautical Information Publications (AIPs) and which are not covered by ICAO SARPS
- Industry standards such as ARINC 424 (mainly for encoding instrument approach and departure procedures)

We interpret the ICAO recommendation to include the standard aeronautical domains identified and modeled in prior AICM/AIXM versions:

- Aerodromes and Heliports
- Navigation Aids
- Airspace
- Enroute Routes
- Terminal procedures, SIDs and STARs
- Aeronautical Organizations, Units and Services
- Traffic flow restrictions and rules

These domains contain features, properties and relationships described explicitly in the ICAO Annexes such as:

- Runway threshold locations
- Airway definitions
- Instrument landing system (ILS) descriptions
- Etc.

In addition, we recognize that new ICAO amendments, industry requirements and future AIM concept of operations are increasing the scope of aeronautical data necessary to support international air navigation. Therefore, the scope of AIXM/AICM 5 must be expanded to cover those features that might be used for future systems such as:

- Protection surfaces for terminal procedures and airways that might be used as part of an electronic "Flight Bag" to provide pilot alerts
- Temporary flight restriction rules meant to restrict access to aeronautical resources based on system capacity constraints, State regulations and/or strategic and tactical traffic flow management.
- Full situational awareness based on Aerodrome Mapping displays with own ship position.

7.1.1 Review of current AIXM data model enhancements

Since the release of AIXM 3.0 in 2002 there has been increased international interest in evolving AIXM to support international requirements for aeronautical information exchange. In recent years the United States FAA and NGA has been actively working with EUROCONTROL to validate the AIXM model and identify areas where the model requires improvement. The recent AIXM 4.x releases have incorporated:

- Value domain changes to accommodate lists of values collected and published by NGA from international sources.
- Improvements to the data models for lighting systems, airspace and, approach lighting systems.

7.1.2 Design recommendations

International review of the AICM/AIXM model that has occurred over the past few years have identified several conceptual areas and features that need to be validated and possibly refactored. Table 1 lists these issues and provides a recommended prioritization. Based on past experience, we anticipate that the investigations will validate most of the AICM model. The most critical analyses are in three areas:

- Terminal procedures. A more comprehensive terminal procedure model is needed to support world wide conventional and RNAV procedures.
- Aerodrome mapping. AICM needs to be able to support aerodrome mapping database requirements.
- Obstacles. New obstacle data collection and transmission requirements published by ICAO require that the AICM obstacle model be refined.

Table 1: Summary of aeronautical domain areas that need to be validated and possible modified as part of AICM 5

Area to Investigate	Priority	Comments
Terminal procedures, SIDs and STARs	High	Incorporate both TERPS and PANS-OPS. Update data model with latest PANS-OPS criteria.
Airport Mapping	High	Integrate the Airport Mapping Database requirements. Primarily this involves decomposing existing AICM features, such as a Runway made from a composition of Runway Elements.
Obstacles	High	Incorporate Amendment 33 to ICAO Annex 15. This includes describing obstacles as points, lines or polygons.
Military aeronautical information	Medium	Support for aerial refuelling and arresting gear.
Navigation Aids	Medium	Modularize navigation aids and separate physical equipment from navigational services.
VFR Procedures	Low	Support for curves and routes that reference geopolitical and topological features.
Services and Organizations	Low	The services model is not consistent with other model areas in AICM
Metadata	Low	Null value handling and integration of the ISO19100 series metadata recommendations.

The data model enhancements stated for AICM/AIXM 5 are currently being implemented and the recommended model updates are being provided in separate reports.

7.2 Feature identification and relationships

Unambiguous feature identification is needed to allow data providers, value-added data and product suppliers as well as end users to be sure that they are communicating about the same aeronautical features.

Within the aeronautical data chain, definitive feature identification is a safety critical issue. Changes to aeronautical data through publications or NOTAM must clearly identify affected features so that pilots and air traffic service providers can correctly respond to the changing aeronautical environment.

With electronic data standards and electronic transmission of aeronautical data, feature identification becomes more critical because new systems are relying on computers to interpret aeronautical information. Persistence of unique feature identification throughout the data exchange chain therefore becomes a requirement.

7.2.1 Feature identification through natural keys in AIXM 4.x

Traditionally, natural keys have been used to identify features in AIXM. According to the AIXM Primer [4]:

AIXM-XML employs natural keys in order to uniquely identify feature instances. For example, the unique identifier of a VOR is composed of the position of the station (latitude and longitude) plus the radio identification. The unique identifier of a feature is declared as a separate complex type in the AIXM-Features.xsd sub-schema. For example, the unique identifier of the VOR feature is declared as the VorUidType complex type. The declaration of the VorType contains a child element named VorUid, having as type the VorUidType.

Natural keys for commonly defined aeronautical features are straightforward:

Table 2: Examples of natural keys for well known features

Aeronautical Feature	Natural Key	Short key definition
Aerodrome	Identifier	Nominally the ICAO or IATA aerodrome code.
NAVAID	Identifier	Nominally 3-character navigation aid code
	Location	Latitude and Longitude

The natural key for some aeronautical features is more complex and may involve references to other aeronautical features:

Table 3: Examples of natural keys involving relationships to other features

Aeronautical Feature	Natural Key	Short key definition
Route Segment	Route Natural Key	The route segment applies to a route
	Start Significant Point Natural Key	The starting point for the segment
	End Significant Point Natural Key	The ending point for the segment.
Apron	Aerodrome Natural Key	The aerodrome natural key is the identifier of the aerodrome.
	Name	Up to 60 character Apron name.

Aeronautical features such as Runway Direction, ILS and Terminal Procedures can involve references to multiple aeronautical features and also nesting of aeronautical feature keys. For example, Runway Direction natural key is a text designator plus the Runway natural key. But, the Runway natural key is a text designator plus the Aerodrome natural key. The nested natural key for the Runway Direction is illustrated in the XML schema shown in Figure 9.

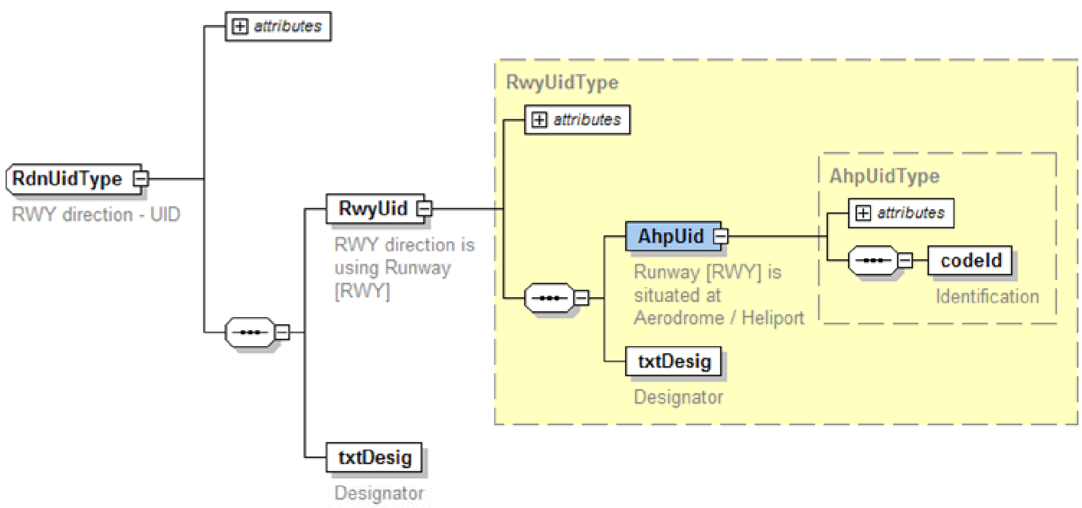


Figure 9: AIXM 4.5 XML Schema illustrating the complex natural key for the Runway Direction feature.

7.2.2 Problems with natural keys

In general, the natural key concept works for well known aeronautical features at large facilities. Major airports, runways and navigation aids are globally known and can be identified by their designators. Problems occur when dealing with traditionally unnamed aeronautical features, features lacking good natural keys, smaller facilities and potential geographic mismatches.

7.2.2.1 Natural keys may change in time

The properties that compose natural keys can change in time. AIXM 4.x includes a special mechanism in AIXM-Update message to notify a change of the natural key. This makes difficult to compare the content of a database over time (two Snapshot messages, for example) as the same feature could have two different natural keys.

7.2.2.2 Unnamed aeronautical features

Many aeronautical features modelled in AIXM do not adhere to standardized naming conventions or lend themselves to natural keys. In these situations AIXM employs three strategies:

- Construct arbitrary natural keys using text names/designators (e.g., Apron, Unit)
- Use sequence numbers or index numbers to track different objects of the same type (e.g,. Unit Address)
- Base the natural key on a composition relationship to another object (e.g., NavaidLimitation is based on the Navaid natural key plus a limitation type).

In these cases, the constructed natural key is "less than natural;" albeit, AIXM includes recommendations to make these constructed natural keys have some meaning.

In addition there is no guarantee that two systems will construct and interpret the keys in the same way: Two systems may name the aprons at an airport in different ways.

7.2.2.3 Feature lacking good natural keys

In some cases, it is not possible to identify a suitable natural key for an aeronautical feature. In these cases the aeronautical feature is converted to a complex type of a parent feature. For example, a procedure leg is defined as a complex element within a procedure, SID or STAR.

Some complex types, like the procedure leg, should probably be aeronautical features. However in AIXM 4.5 procedures legs are not features because a natural key could not be identified. A natural key should not be a prerequisite for classifying features.

7.2.2.4 Smaller facilities

Smaller domestic facilities may not have internationally assigned identifiers and this can make natural keys fail. For example, within the United States there are hundreds of airports without four-letter international airport codes. Similarly one can expect that there are airspace, routes, navigation equipment and other aeronautical systems that likewise do not have global identification.

7.2.2.5 Geographic mismatch

Some natural keys like the natural key specified for Navigation Aids depend on the aeronautical feature location expressed in latitude and longitude. This requires all systems exchanging the navigation aid to use the exact same location in order to positively identify a feature. Perfect location matching can be difficult to achieve due to:

- Different datums (for example, significant points situated on the border of two States using different datum)
- Different location representations: decimal degrees or degrees minutes seconds
- Differences in precision and rounding.

7.2.2.6 Data modelling issues

In addition, to these technical problems with natural keys, the natural key implementation used in AIXM 4.x can lead to some data modelling issues [15]:

- The natural keys have a dual purpose: they are used as both the properties of the feature as well as a composite key for feature identification. This becomes problematic when one of the properties used as a natural key changes.
- Increased implementation effort because the natural keys of each feature are formed differently. This requires an implementing system to develop a feature specific rule set for interpreting each feature and its relationship.

7.2.3 Alternatives for feature identification

This subsection considers three alternatives for feature identification:

- Natural keys
- Global registry
- Artificial keys

7.2.3.1.1 Natural keys – Current approach

Systems like AIXM 4.x, ARINC 424 and NOTAM use natural keys to identify aeronautical features.

Natural keys work for AIXM 4.x because AIXM 4.x targets a regional EAD system. Specifically:

- The EAD members have accepted to adhere to the natural key identification established by AIXM 4.x and to strictly apply the ICAO SARPS on which some of these are based.
- The data stored in the EAD comes directly from clearly identified official sources. Each State is responsible for populating the properties that compose the natural key. There is no need to re-conciliate versions of the same data from different sources, thus no need to de-conflict keys.
- The EAD maintains a staff of analysts who provide manual quality control and harmonization of input AIXM data.

Natural keys in ARINC 424 work because:

- typically, ARINC 424 files contain information about main facilities, with well established identifiers
- the country code is added in almost all records, thus ensuring the geographical separation of potentially mismatching data
- the exhaustive use of sequence numbers (for example, in route segments, vertex and procedure legs)
- the ARINC 424 format is rarely used as an update format, most frequently a file containing a full data set for a wide area.

Similarly, natural keys for NOTAM generally work because NOTAM are created by an originator, who tries to provide sufficient information so that the feature concerned by the temporary situation can be unambiguously identified. Within the context of the originator the natural key descriptions are unique. However, it is very likely that there have existed cases where ambiguous NOTAM have led to pilot misinterpretation, with the potential for accidents.

7.2.3.1.2 Global Aeronautical Registry

A future approach for feature identification might be the establishment of a global aeronautical feature registry. The register would be operated by an international aviation authority and could be used to assign unique identification to all aeronautical features. Data suppliers and consumers could then look up features in the register to obtain a positive feature match. A global register may be the ultimate solution for feature identification; however the mandate and resources for building a global aeronautical registry do not exist.

7.2.3.1.3 Provider specific keys

A third alternative is to rely on artificial keys and internal system processing to handle feature identification. In this alternative, data providers supply an artificial key that is unique within the data provider’s context. Aeronautical data consumers would need to store the artificial keys provided by the data providers or develop internal feature reconciliation processes to identify features.

To see how this works today, consider one of the missions of the United States National Geospatial Intelligence Agency (NGA): collect and integrate the world’s aeronautical data. To that end, NGA collects aeronautical data from a variety of data sources including host country providers and aeronautical companies. Consequently, NGA may receive information about the same aeronautical feature from more than one data provider. To resolve this duplication, NGA has developed internal reconciliation algorithms and processes to match transmitted data with the NGA internal database of aeronautical features.

Another example is obstacle data within the United States. Today it is not possible to identify obstacles using natural keys because a consistent set of key properties does not exist. Every publication cycle the United States National Aeronautical Charting Organization (NACO) publishes the Digital Obstacle File (DOF) containing a list of known obstacles throughout the United States. This file has become a *de facto* standard for obstacles in the United States and the NACO generated artificial DOF number has become *de facto* the unique key for obstacles.

7.2.3.2 Design recommendations

Our analysis suggests that none of the feature identification approaches satisfy our design requirements completely:

- Natural keys work well for well known features but fail for local aeronautical features and features without natural keys, such as runway markings
- A global aeronautical registry is probably decades from realization
- Provider specific keys may be acceptable in some circumstances but place a burden on the data receiver to reconcile data sources and potentially store the artificial keys of all data providers.

Instead we propose a hybrid approach where:

- Feature identification is based on an artificial global unique value and
- Relationships are specified with queries, which may use either the feature artificial identifier or any collection of a feature’s properties.

7.2.3.2.1 Identifier property

In the recommended approach all AIXM features have an identifier property that is meant to be a globally unique identifier for the object. In the future this might contain the feature key for a globally managed aeronautical data registry. Alternatively the identifier can contain an artificial key provided by the data supplier.

To ensure that the artificial key is globally unique, we recommend that the identifier include the namespace of the data supplier. A recommended namespace convention is included in Section 10.

7.2.3.2.2 Query Relationships

Features are identified in relationships by specifying a subset of feature properties that can be used to uniquely identify the target feature. The subset of properties used for feature identification is encoded by the data supplier. Figure 13 shows the generic situation for two related features: Feature1 and Feature2. The relationship between the features is stereotyped with <<query>> and we show two possible queries used to define the association.

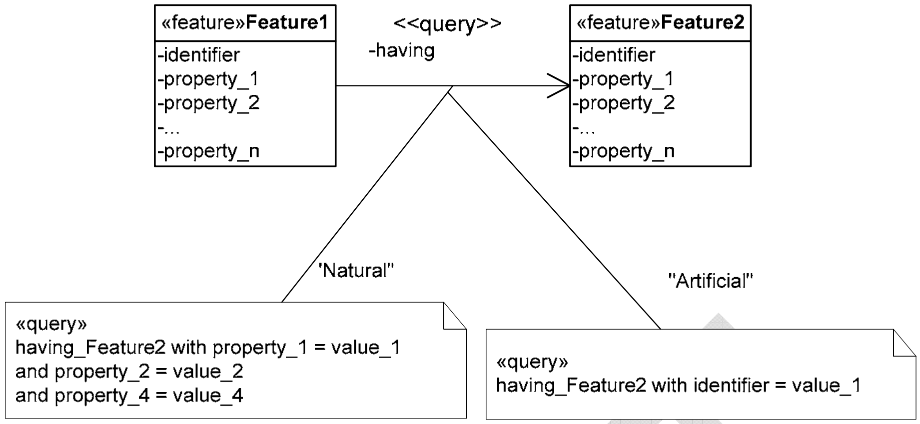


Figure 10: Relationship between two features using the <<query>> approach.

As an example consider the object diagram showing Runway Direction 20L on Runway 02R/20L at Aerodrome MABC (Figure 11).

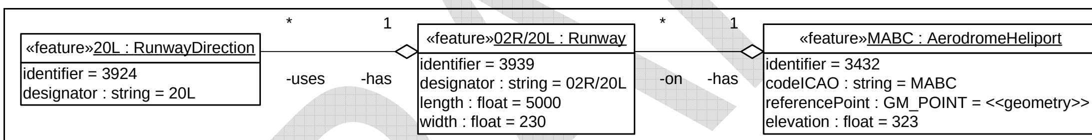


Figure 11: Example of feature relationships for Runway Direction 20L at Aerodrome MABC.

Following this design recommendation, the relationship between the Runway Direction and the Runway can be expressed in alternative ways by providing a query. Some examples in natural language are:

```
Alternative 1: Using an artificial identifier
Runway Direction    uses   Runway where identifier = 3939

Alternative 2: Using a natural identifier
Runway Direction    uses   Runway where designator = 20L/02R
                    and    on Aerodrome where codeID = MABC

Alternative 3: Using a combination natural identifier and
artificial identifier
Runway Direction    uses   Runway where identifier = 3939
                or
                    uses   Runway where designator = 02R/20L
                    and    on Aerodrome where codeID = MABC
```

The result of the query can return 0, 1 or more records:

Table 4: Interpretation of <<Query>> result set

Result Set	Interpretation
0	Target feature not found. The query can be incorrect. Data receiver may need to consult with the data provider to identify the target feature.
1	Exact match found. The normal situation.
2+	More than one target feature meets the relationship criteria. For associations with a multiplicity of 1 the query is ambiguous. May be appropriate for associations with cardinality greater than 1.

7.2.3.2.3 Advantages

The query approach to feature identification and feature relationships has several advantages:

- It supports both natural identification and artificial identification
- The properties used for natural identification are not hard coded in the exchange standard. This allows data supplies to use flexible rules when encoding natural identifies. For internationally known features like an international airport simple natural identifiers can be used while for local features like a small domestic airport other properties can be used.
- The query is implementation independent. For a database implementation the query can be implemented using SQL while for a GML-compliant XML exchange schema the query can be implemented using xlink:href.

7.2.3.2.4 Disadvantages

There may also exist disadvantages of this approach, such as:

- For a data receiver, the exact composition of such 'relationship queries' is unknown in advance and it could even vary from instance to instance of the same feature. If this is a problem, then the user community affected by the problem should come together and agree, feature per feature, on the identifying properties.

7.3 Geometry

Geometry is an integral part of aeronautical feature definitions. In the aeronautical domain, feature geometries may be 2 or 3 dimensions. Some examples include (see Figure 12):

- Navigation aid represented as a point with an elevation
- Runway element represented as a line segment
- FIR (Flight Information Region) airspace represented as a prism (vertically extruded horizontal polygon)

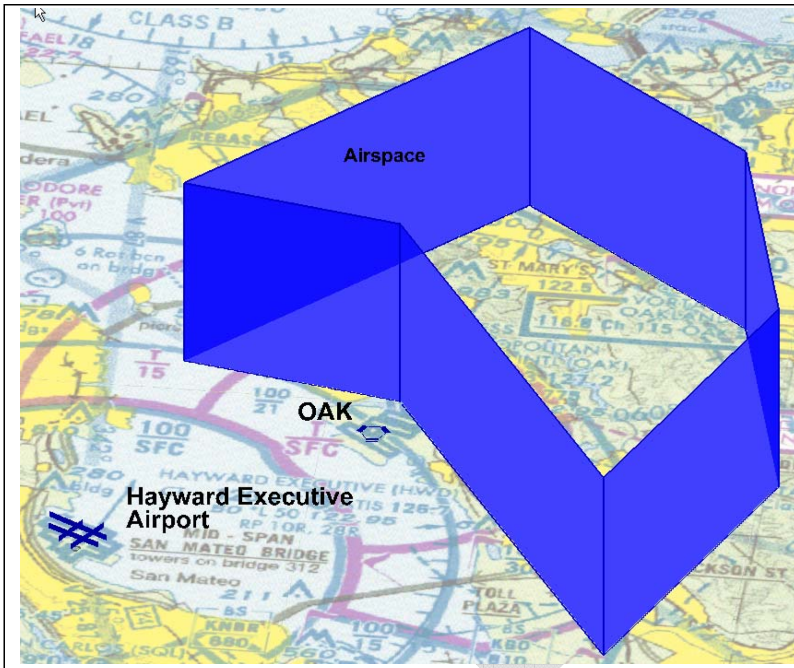


Figure 12: Example of typical aeronautical geometries

Currently AIXM uses an aeronautical-specific model for representing geographical information with definitions that are specific to the feature type [4]. For example:

- Airspace is defined using an airspace aggregation model where complex 3D airspace is constructed by union, subtraction and intersection of simple prisms.
- Airport surfaces like aprons and clearways are defined as closed curves using a custom point and path model
- Navigation aid limitations and Minimum Safe Altitude (MSA) areas are protected airspace defined as circle segments relative to a significant point.

The main advantage of the AIXM 4.x approach is that geometry definitions reflect how they are created in the aeronautical domain; however this means that COTS GIS software cannot readily interpret AIXM geometries without extensive customization.

7.3.1 Design recommendation

We recommend that AICM/AIXM feature geometry properties be standardized based on the ISO19107 spatial schema standard. The ISO19107 schema describes spatial objects with spatial characteristics like size, shape and topology [16]. Using ISO19107 as the basis for geometric representation in AIXM has several advantages:

- Standardized geometric representation
- ISO19107 is the basis for Geometry Markup Language (GML), an XML grammar for spatial features.
- Increases the potential for AICM/AIXM implementers to leverage COTS GIS tools.

7.3.1.1 Normative and Informative Geometry

The disadvantage of GML is that geometries are represented without regard for the domain information on how they were created. For example in GML the Service Volume segments depicted for the OAK VOR in Figure 13 would be described as GML polygons. However, in the aeronautical domain they are constructed as bearings and distances from the VOR. By itself the GML geometry does not describe how the Service volume is constructed. Where custom aeronautical definitions must be maintained, additional feature properties must be included to supplement the standard geometry description with the aeronautical specific information. This is illustrated for the Navaid Limitation feature in Figure 13 where the AIXM 5 feature definition includes fromAngle, toAngle, innerDistance and outerDistance in addition the ISO19107 GM_POLYGON extent property.

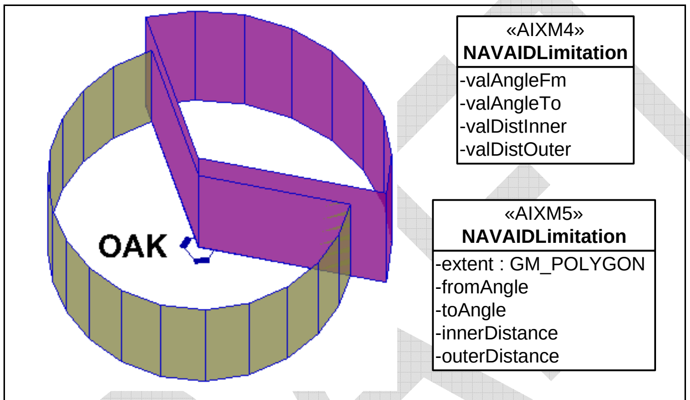


Figure 13: Illustration of how aeronautical properties that define geometry are preserved in AICM/AIXM 5.

This duplication of geometry data leads to a situation where the two definitions may not be consistent. In AICM and AIXM we identify the aeronautical properties as normative or as the official definition of the geometry while the ISO19107 geometry is an informative property.

7.4 Temporality

There are two levels at which aeronautical feature instances are affected by time [AICM Edition 1]:

- Every feature has a start of life and end of life
- The properties of a feature or the target of any feature relationship can change within the lifetime of the feature

AIXM is intended to support data exchange between systems; therefore, AIXM must support a temporality model that accurately represents the temporal state of the aeronautical features. Examples of temporal data that might be exchanged using AIXM include:

- Regular AIRAC cycle updates or data amendments
- Temporary changes to aeronautical data such as those changes currently recorded in NOTAM.

- Permanent changes to aeronautical data (e.g., PNOTAM) that notify the aviation community of a permanent change that will be included in the next set of publications.

From these principles and examples we derive a set of high level requirements:

- Feature instances have a date and time for start of effectivity and end of effectivity
- Any feature property may change at any time. The value of a feature property may have a start of effectivity and end of effectivity
- Feature property changes can be classified as temporary or permanent based on the need to support current AIM concepts of operation. For example permanent changes may be captured as AIRAC amendments and included on paper charts and publications. Temporary changes are typically NOTAM.

7.4.1 Assessment of AIXM 3.x and 4.x Temporal Model

In AIXM 3.x and 4.x, features and feature properties are implicitly static: AIXM 4.x and 3.x features do not contain properties for expressing feature-level or property-level temporality. Instead, aeronautical data temporality is handled in the two AIXM messages, <AIXM-Snapshot> and <AIXM-Update>, that comprise the AIXM 3.x and AIXM 4.x data exchange specification.

7.4.1.1 <AIXM-Snapshot>

The <AIXM-Snapshot> message has the following format:

```
<AIXM-Snapshot   xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"   xsi:noNamespaceSchemaLocation="AIXM-Snapshot.xsd"
version="1"      origin=""      created="2005-05-23T10:00:00"
effective="2005-05-23T10:00:00">
   <Ahp>…</Ahp>
   <Ahp>…</Ahp>
   …
</AIXM-Snapshot>
```

The <AIXM-Snapshot> element contains an attribute *created* for recording the date that the AIXM exchange message was assembled and *effective* for recording the valid date for the snapshot message.

An <AIXM-Snapshot> provides the state of a set of aeronautical features at a point in time.

7.4.1.2 <AIXM-Update>

The <AIXM-Update> supports adding, removing and modifying aeronautical features. Like the <AIXM-Snapshot>, the <AIXM-Update> has attributes for specifying the created date and the effective date. A partial <AIXM-Update> looks like:

```
<AIXM-update     xmlns:xsi=”http://www.w3.org/2001/XMLSchema-
instance”     xsi:noNamespaceSchemaLocation=”AIXM-Update.xsd”
effective=”2005-10-10T10:00:00” created=”2005-10-10T10:00:00”
version=”2” origin=”FAA”>
<Group>
   <New>
      <Aas>…</Aas>
      …
   </New>
   <Changed>
      <Aas>…</Aas>
      …
```

```
    </Changed>
    <Withdrawn>
       <Aas>…</Aas>
       …
    </Withdrawn>
</Group>
</AIXM-update>
```

The <AIXM-Update> supports permanent changes to features that occur at a point in time.

7.4.1.3 Conclusions

AIXM 3.x and 4.x provide limited temporality support. It is possible to exchange feature state at a point in time and communicate permanent changes. AIXM 3.x and 4.x do not support temporary changes to aeronautical data.

In addition, AIXM 3.x and 4.x embed temporality in the exchange message rather than in the aeronautical features. Consequently temporality becomes a property of the message rather than the aeronautical features. The message properties describe how receiving systems should interpret the message content.

7.4.2 Conceptual model

To refine our understanding of temporality and how it applies to the aeronautical domain, we have developed a conceptual model of temporality. In this model we have the concept of "version" and "delta" (see Figure 14):

- **Version** – The state of a feature and the value of all its properties in the time period between two changes.
- **Delta** – The difference between two consecutive versions. The delta contains only those properties that have changed from one version to the next.

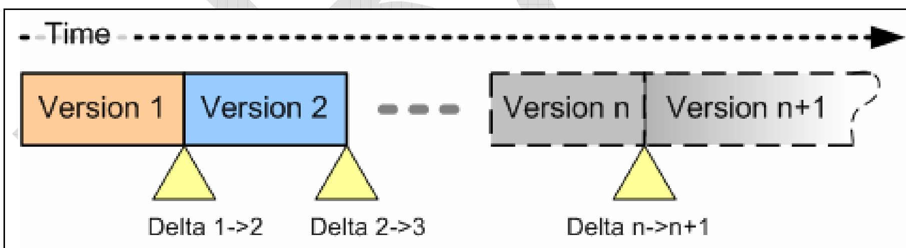


Figure 14: Relationship between Versions and Deltas

We also recognize that, in the aeronautical domain, feature state can be permanent or temporary. The distinction between temporary and permanent is rooted in current aeronautical system operational concepts:

- **Permanent** - Permanent feature state is typically reported as an AIP, AIP amendment (AMDT) or a permanent NOTAM. The permanent state of a feature is normally incorporated into static publications and aeronautical charts.
- **Temporary** – Temporary feature state is normally associated with NOTAM. Temporary states are normally transmitted to operational systems but they are not normally charted or printed.

The combination of “version” and “delta” along with the operational distinction between “permanent” and “temporary” leads to five temporal components that need to be supported in AICM and AIXM 5:

- **Baseline** – The state of a feature and all of the feature properties as a result of a permanent change. The Baseline state of a feature also exists when the feature is initially created. The baseline state lasts until the next permanent change.
- **Version** – The state of a feature and all the feature properties during the time period between two changes.
- **Permanent Delta** – A set of properties that have changed or will change permanently. The permanent delta will result in a new baseline.
- **Temporary Delta** – A set of values for one or more feature properties that are effective for a limited time. The result is a temporary change to an underlying feature version.
- **Snapshot** – Feature state at a time instant. The snapshot is the result of a combination of feature versions and any deltas that are active at the time instant.

Figure 18 illustrates the temporal model by showing changes to a navigation aid between one AIRAC cycle to the next. In this example, NAVAID AML has a frequency change from 126.00 MHz to 132.5 MHz between two AIRAC update cycles. Changes in the operational status of the AML NAVAID during the frequency upgrade lead to two NOTAM. Based on this diagram we can identify the following temporal components:

- The diagram shows two Baselines. The first baseline has a NAVAID frequency of 126.00 MHz and the second baseline has the new frequency of 132.50 MHz.
- A Permanent Delta can be used to describe the different between the two baselines. In this example, the permanent delta would indicate that the AML NAVAID frequency was changed.
- Each NOTAM can be expressed as a temporary delta that changes the Operational Status of the NAVAID.
- Based on the changes shown in the diagram, four versions of the NAVAID feature can be identified. Each version begins and ends at the boundary of a Permanent or Temporary Delta.
- Finally an arbitrary number of Snapshots can be created. Each snapshot shows the aggregate feature state by combining the Baseline feature state and any changes reported in Deltas. Note that the feature state of a Snapshot is the same as the feature state of the version that spans the time instant specified in the Snapshot.

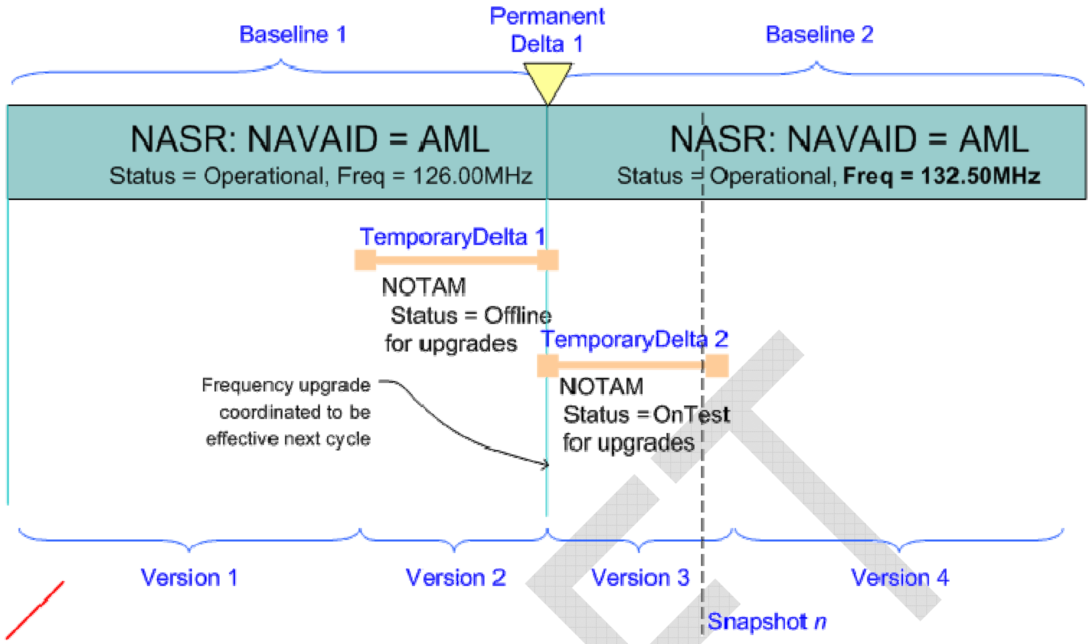


Figure 15: Illustration of AICM/AIXM temporal component model for a Navigation Aid undergoing a frequency upgrade.

The temporality model is comprehensive and flexible enough to represent static and dynamic data in the aeronautical domain.

7.4.3 Implications for aeronautical information systems

The conceptual temporal model described in the previous section provides considerable flexibility for systems that implement temporality. A system that tried to fully implement the AICM temporality model would be very complex. AICM incorporates a complete temporal model to ensure that AICM can support all current and future aeronautical applications; however there is no requirement for systems implementing AICM need to support all temporal components. Indeed:

- Systems may only store baselines and disregard any temporary or permanent changes. Examples include AIP publishers, paper chart publishers and ARINC 424 based systems.
- Systems may only transmit and store temporary changes. Examples include the conventional NOTAM office.
- Systems may only require periodic snapshots providing the current state of the system. An example is a passive monitoring system designed to report system status at selected time intervals.
- Systems may want a new revision after every change without making a distinction between a temporary and a permanent change. Examples include traffic management and flight plan processing systems.
- Some systems may be developed that can process and interpret all of the temporal components and provide users with Baseline, Deltas and the actual Snapshot/Revision at any given moment in time.

The AICM/AIXM contains a complete temporal model; however, as the examples illustrate it is the responsibility of interacting systems to negotiate specific temporal data exchange requirements as well as to integrate temporality into their internal subsystems.

More than one combination of temporal components can be used to express the same temporal information. As illustrated in Figure 19, a Version can be constructed by considering a Baseline and incorporating PermanentDeltas and any Temporal Deltas that are active during the effective start and end of the version.

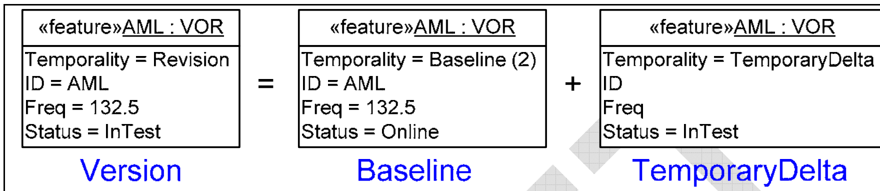


Figure 16: Two ways to represent the same temporal state.

The conceptual temporal model explains how temporal information can be encoded in various forms for information exchange; however, an equally important implementation aspect is synchronization issues that could be experienced by systems realizing the AICM temporality model. For example, two systems might produce different snapshots at a time t if they are not fully synchronized. These synchronization problems are manifest today and it is not within the scope of AICM to impose a methodology for eliminating data integrity issues associated with database synchronization. However the AICM temporal model does provide the framework for supporting system synchronization.

7.4.4 Design recommendations

We assert that temporality affects all aeronautical features, so a single temporal model needs to be applied across the entire aeronautical domain. Key assumptions in this design approach include:

- Temporality is an essential characteristic of aeronautical information systems.
- A general temporal model should be uniformly applied to all aeronautical feature types.
- Since temporality applies to all aeronautical feature types, temporality should be abstracted from the task of modeling object properties.

AICM and AIXM will support all of the components described in the temporal concept model:

- Versions and Baselines
- Deltas (Temporary and Permanent)
- Snapshots

7.4.4.1 TimeSlice Model

To implement temporality in data exchange we will adopt the TimeSlice data content model that is defined as part of GML 3.1.1* because the TimeSlice model closely matches AIXM's temporal requirements. According to the GML 3.1.1 specification [8], a TimeSlice encapsulates the time varying properties of a dynamic feature. A dynamic feature is any feature that varies in time.

The TimeSlice model is illustrated schematically and in UML in Figure 17 [UML adapted from 8]. As shown in the model, the AIXM Feature has a static component which contains properties for the start of existence, end of existence and the artificial identifier. All other properties of the feature are assumed to be temporal. The temporal feature properties are encapsulated into a TimeSlice object.

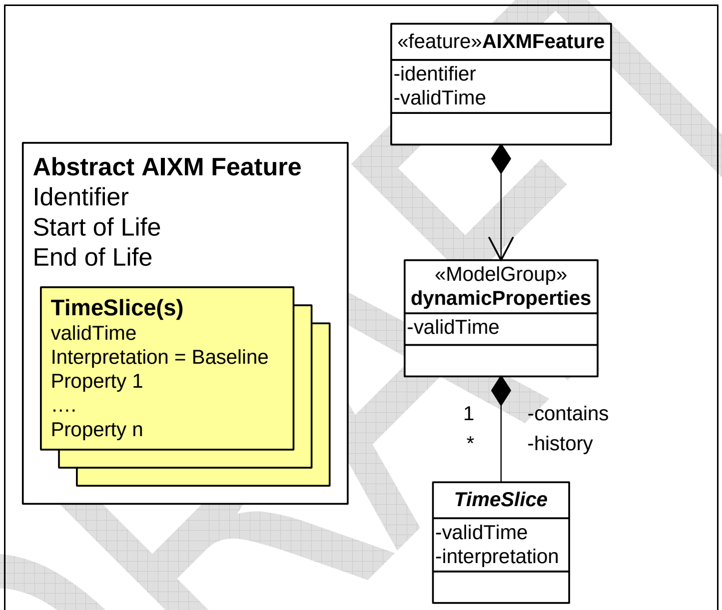


Figure 17: TimeSlice model

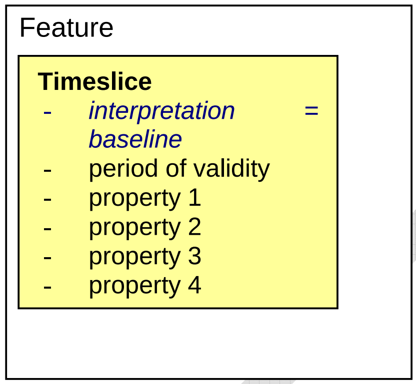
Each TimeSlice object contains a valid time interval and an interpretation property. The interpretation property indicates the temporal component that is being modeled. Valid values for the interpretation are:

- Baseline
- Revision
- TemporaryDelta
- PermanentDelta
- Snapshot

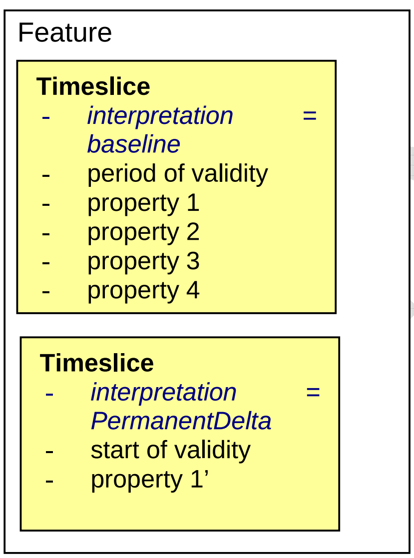
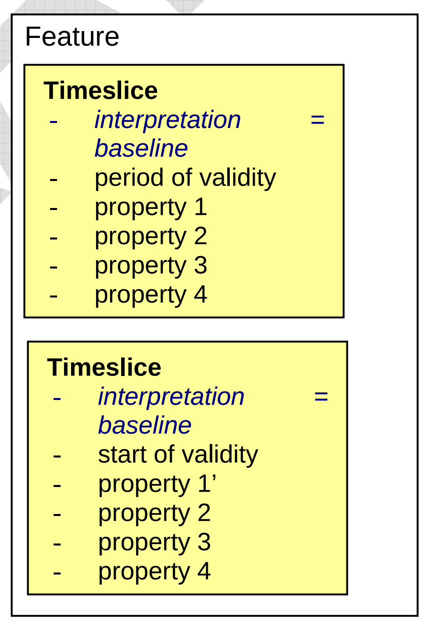
* Further analysis is needed to determine if the GML 3.1.1 TimeSlice model is compliant with the ISO19108 Temporal Schema. If it is then AICM should be based on 19108.

7.4.4.2 Communicating feature property changes

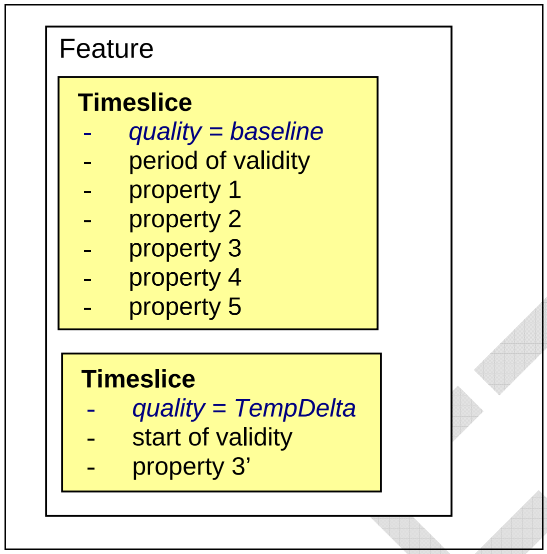
Depending on the temporal implementation employed by the exchanging systems, different methods can be used to communicate feature changes. A baseline is communicated as one TimeSlice, containing all properties that have a value for the specified time period



A permanent change may be communicated as a sequence of two TimeSlices; the following two possibilities exist:

one containing the previous baseline and one containing all properties that change (permanent delta)	*one containing the previous baseline and one containing the new baseline*
	

A temporary change may be communicated as a sequence of two TimeSlices: one containing the baseline and one containing all properties that change (permanent delta)



7.4.4.3 Integration with the feature identification

In order to support the general recommendations for feature identification, some temporal components will need to be combined to fully define the aeronautical feature. If a system is using a natural property identification approach to feature identification then simply transmitting a Delta will not be sufficient to identify the feature because the Delta may not contain the properties referenced in the Feature Relationship query. In this case the Delta will need to be transmitted with a Version that was active at the time the Delta became effective.

7.4.4.4 Integration with feature relationship

The TimeSlice model for AIXM features makes feature relationships more complicated. Feature relationships, like other properties, can change with time so they need to be encoded within the feature TimeSlice.

Feature relationships are encoding using queries against sets of feature property data (see Section 7.2). The feature relationship query needs to incorporate the TimeSlice model into the queries. We have the following recommendations for integrating feature relationships with the TimeSlice model:

- The feature artificial identifier is time invariant, so expressing feature relationships with the feature artificial identifier is trivial.
- Delta TimeSlices cannot be used in feature relationships because the Delta TimeSlice does not contain all of the feature properties.
- Feature relationships that encode natural identification must reference a Baseline or Version TimeSlice.
    - The feature relationship must include the interpretation property to indicate if the relationship is based on a Baseline or Version.
    - We recommend that feature relationships be based on Baseline features.

Following these guidelines the feature relationships shown in Figure 11 would be re-written as:

- Example 1: Using an artificial identifier
  Runway Direction **uses** Runway where identifier = 3939

- Example 2: Using a natural identifier (alternative 1)
  Runway Direction **uses** Runway 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

- Example 3: Using a combination natural identifier and artificial identifier (alternative 2)
  Runway Direction **uses** Runway where identifier = 3939
  or **uses** Runway where interpretation = Baseline
  and designator = 20L/02R
  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

Example 1 has not changed because the artificial identifier is time invariant. Examples 2 and 3 indicate that the relationships reference the Runway Baseline TimeSlice that is active between November 21 and December 21, 2005. Also note that the secondary relationship from the Runway to the Aerodrome must also encode the Baseline, startPositon and endPosition.

7.5 Extending features

EUROCONTROL AISTEC/ACCB-04/WP1 discussed expanding the scope of AIXM to "develop AIXM as a globally applicable aeronautical data exchange specification, satisfying the needs for international aeronautical information dissemination of the stakeholder States, including temporary changes (NOTAM), <u>with a standard extension mechanism</u> [Emphasis added], which enables the use of AIXM for a wider spectrum of aeronautical services applications." [3].

As AIXM is augmented to meet other application requirements and the needs of other States, AIXM will need to be extended to support additional data types, attributes and messages. The requirements and scope of these extensions cannot be anticipated; therefore, AIXM requires a standard mechanism for adapting AIXM to specific application system requirements while simultaneously preserving the base AIXM standard.

The Extension Model allows AIXM XML documents to contain additional properties that might be specific to a country, a system interface specification or other use case. Examples of extensions include:

- A field to store country specific spelling of an airport name
- A code type classifying an airspace using country specific airspace types
- Styling codes used to mark-up an aeronautical feature so that it can be displayed on a chart or cockpit display

In other words, extensions may be needed whenever local or system specific information needs to be transmitted in an AIXM XML document.

Extensibility has two important advantages [17]:

- Increased adoption of AIXM by allowing AIXM to be used for applications for which it was not originally designed
- Decrease pressure on the AIXM configuration control board by allowing local extensions to AIXM

The high level requirements for extensibility include:

- Allow flexibility so that applications can add new AIXM Feature properties and relationships
- Establish a standard approach for implementing and documenting extensions
- Provide encoding rules so systems that can read the AIXM base standard will be able to read AIXM XML documents that contain extensions.

7.5.1 Guidelines for creating extensions

The extension model provides enormous flexibility for augmenting and expanding the core AICM and AIXM information exchange models. The intent of AIXM is to provide a common language for aeronautical information and a common format for data exchange. The use of extensions erodes AIXM's purpose by thwarting international aeronautical data harmonization efforts. For this reason, the use of extensions should be carefully managed by AIXM adopters.

In general the following guidelines should be used when considering feature extensions:

- New properties and relationships that have international application in support of modelling aeronautical data for use in international air navigation should be consider for direct addition to the AICM/AIXM model through the AIXM configuration management process
- Extensions supporting application specific implementations not directly covered by AIXM are candidates for extensions (see discussion in Section 5.3.1)
- Ideally extensions should be managed by communities of AIXM users so that common extensions can be shared
- Globally applicable extensions should be reviewed by the AIXM configuration control board as candidates for adoption into the base AICM/AIXM model.

7.5.2 Analysis of extension requirements

There are three ways that the AIXM data can be extended:

Extension	Description	Examples
Properties and Relationships	Add new properties or new AIXM Feature relationship to an AIXM Feature	• Add a LocalName field to the AerodromeHeliport AIXM Feature. • Add a relationship between Airspace and the AerodromeHeliport
Code Types	Add additional lists of values to existing AIXM code types.	• A country may have a specific airspace activity for “Unmanned Aerial Vehicles Testing”. This new activity does not exist in the current list of values for the codeActivity in the Airspace AIXM feature.
AIXM Features	Create a new AIXM Feature describing an aeronautical object.	• Create a new feature called “Military Aerial Refueling Route”

7.5.3 Design recommendation

7.5.3.1 New Properties and Relationships

The need for feature property extensions applies to all AIXM features, so a system wide approach is recommended. The UML diagram in Figure 18 illustrates the extension model. Each AIXM Feature TimeSlice contains an extension property that is associated with an Abstract Extension Interface. Specific extensions realize the Abstract Extension Interface and are expected to be uniquely identified through a Namespace. The AIXM namespace recommendations are discussed in Section 10.

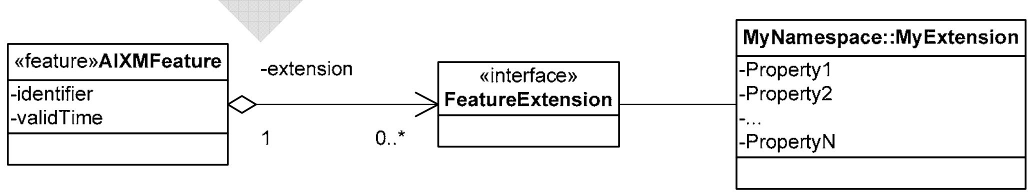


Figure 18: UML model for feature property and relationship extensions

The extension model includes support for new Properties, Relationships and Code Types. Guidelines for adding property and relationship extensions are given below.

7.5.3.1.1 New properties

New properties are included as properties of an Extension class that derives from AbstractFeaturesExtension Class.

A new property must comply with the following conventions:

1. The property name must be meaningful
    a. The property description must be written in lowerCamelCase
    b. The property description must include only well known abbreviations.
    c. The property name must be written in English, in order to remain consistent with the overall AIXM model
2. The resulting property names must be different than the standard AIXM property names for the AIXM Feature that is being extended.
    a. For example, extending Aerodrome Heliport to include a new property named codeActivity is acceptable because this property name does not exist in the base Aerodrome Heliport Feature. However, adding a new property named codeActivity would not be acceptable for the Airspace Feature because Airspace already contains a property named codeActivity.
3. Identify the value domain type for each new property.
    a. Standard domains should be based on the existing AIXM data types.
    b. If a standard value domain type cannot be used then a new value domain type is required

An example is adding Acceleration property to an AIXM feature. Acceleration is a numerical quantity expressed in units like m/s^2. Assume that this new property is being added to the Procedure Leg to record the recommended aircraft acceleration. Applying the rules listed above we obtain the following:

1. Acceleration is a meaningful property description so the AIXM property name becomes “acceleration.”
2. A review of the standard Procedure Leg properties shows that the name “acceleration” is unique.
3. A review of AIXM value domains shows that AIXM does not have a value domain for acceleration so new value domains will be required to document this extension.

7.5.3.1.2 New relationships

New relationships are included as properties of an Extension class that derives from the AbstractFeatureExtension Class.

A new relationship must comply with the following AIXM conventions:

1. The relationship name shall consist of three concatenated parts as defined below:
    a. A meaningful relationship role name that describes the association. The role name shall be written in English, in lowerCamelCase and include only well known abbreviations.
    b. An underscore separator
    c. The name of the target feature type in the relationship in UpperCamelCase.
2. The relationship name must be different than the standard AIXM Feature association names.

As an example consider a relationship to an emergency Aerodrome where the relationship describes the aerodrome that should be used in emergency situations. Applying the conventions listed above we obtain the following:

1. The relationship name should constructed as:
    a. The association role name is “forEmergency” and this is written in lowerCamelCase
    b. Followed by “_”
    c. Followed by the target AIXM feature, “AerodromeHeliport”

   The resulting association name is forEmergency_AerodromeHeliport.
2. A review of the Aerodrome Heliport feature shows that forEmergency_AerodromeHeliport is a unique name.
3. The forEmergency_AerodromeHeliport relationship should contain a query identifying the Aerodrome used for emergencies.

7.5.3.1.3 Updates to code values

An AIXM adopter may desire to expand one of the standard AIXM code lists. For example, an adopter may desire to add new Activity types to the list of Airspace code_Actvity.

Extending AIXM by adding additional domain values to a standard AIXM code type is not directly supported. The standard code lists provided in AIXM are meant to be internationally applicable, clearly defined lists. Allowing local extensions to these lists of code values can lead to problems when systems try to interpret the data. Instead this extension can be handled using either of the following solutions:

1. Create a new property to store the additional code values. For details on adding new properties to AIXM Features see 7.5.3.1.1.
2. Work through the AIXM configuration control board to submit the additional code values for adoption into AIXM.

For example, suppose the United States needs to add new Airspace Code_Activity such as “Bungee Jumping.” This extension would require that the existing Airspace codeAcivity value domain be amended with an additional value. This extension is forbidden. Instead there are two options:

1. Create a new property to contain the addition code value, or
2. Submit the request to the AIXM configuration control board

The “Bungee Jumping” Airspace activity is a local extension, so submitting this to the AIXM configuration control board is not appropriate. Instead this extension is handled by:

1. Setting Airspace Code_Activity to “OTHER”
2. Extending Airspace by adding a new property called “activityUS” inside a USFeatureExtension class.

7.5.3.2 New features

The extensible modelling framework adopted for AICM and AIXM 5 will make it easy to add new features in the same style as the standard AICM/AIXM features. Conceptually a new feature is added by:

- Deriving the Feature Type from the abstract AIXM Feature.
- Deriving a new TimeSlice object from the abstract AIXM TimeSlice object

- Deriving a new Extension interface from the abstract FeatureExtension interface.

New features shall follow the AIXM naming convention for feature, properties and relationships names. The English language shall be used.
As we will see when we discuss AIXM 5 implementation, AIXM Features will be implemented as GML features. So any GML-compliant feature could be included in an AIXM 5 exchange message.

Despite the fact that the modelling framework supports new custom features there is no guarantee that AIXM data receivers will be able to interpret the custom features. Instead we recommend that requests for new features be submitted through the AIXM configuration control board.

7.5.3.3 Ensuring compatibility

Modelling extensions using the FeatureExtension interface shown in Figure 18 is meant to ensure compatibility with systems that can only understand the base AIXM exchange messages or systems that understand a subset of feature extensions.

The extension interface encapsulates feature extensions in a defined location within the Feature TimeSlice. So systems that only read the base AIXM model can skip the contents of the extensions.

In addition, all extensions must have a namespace. Systems that can read selected extensions can identify valid extensions through their namespace and easily skip unknown extensions.

7.6 Extensible exchange message framework

Closely related to the requirement for providing a standard mechanism for extending the properties and relationships of a feature (Section 7.5) is the requirement to support a range of data exchange use cases through an extensible messaging framework.

Currently AIXM 4.x supports two messages: <AIXM-Update> and <AIXM-Snapshot>. These messages are an integral part of the AIXM 4.x specification. Both messages are customized to support aeronautical information collection and transmission by the EAD.

The close coupling between the AIXM 4.x data model and the AIXM 4.x messages makes it difficult to apply the AIXM model to other aeronautical information systems.

The high level requirements for an extensible message framework include:
- Separation of the AIXM data model from the messages
- Standard approach to encapsulating AIXM data into the messages
- Support for message content such as message properties, source/client metadata and operation parameters.

7.6.1 Message Structure

We can think of messages as having two components (see Figure 19):
- Actions
- Feature Payload

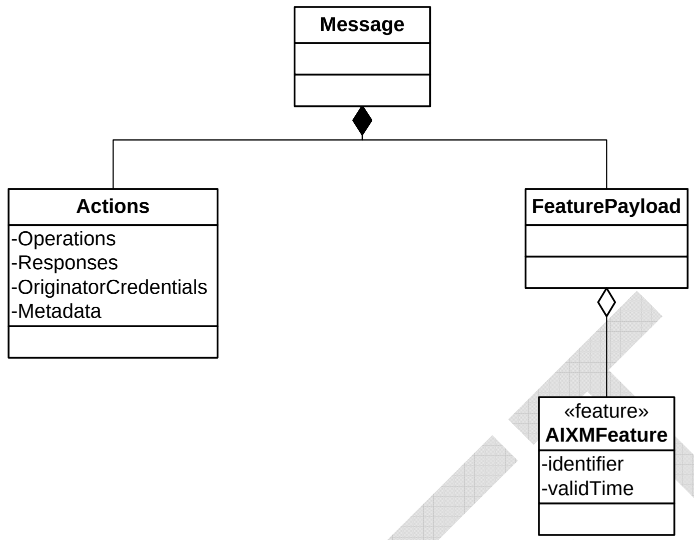


Figure 19: A message has an Actions component and a Feature Payload component.

Messages contain the data content as “FeaturePayload” and tell receiving systems how to “Act” on or interpret the message content. The FeaturePayload contains a collection of AIXMFeatures. The Actions may contain:

- Operations to be performed
- Responses
- Message originator credentials
- Other message meta-data and properties.

Message formatting and interpretation can be as important as the data content. However, whereas the data content model can be explicitly defined, messages are often application specific. Sometimes, like in the case of a Web Feature Service, the messages are well known standards. In other cases, system to system interchange may occur through negotiate messages. For example, the AIXM 4.x <AIXM-Update> and <AIXM-Snapshot> are custom messages designed to support EAD activities.

For AIXM to be adopted for other systems, we can anticipate additional messages:

- xNOTAM (Notice of a temporary change)
- Obstacle data product (as specified in ICAO Amendment 33 to Annex 15)
- Aerodrome Mapping Databases (AMDBs)

7.6.2 Design recommendations

We adopted a generic message pattern from GML that can be used to construct custom AIXM messages.

Figure 20 is a UML diagram of the AIXM message format. Messages are derived from AIXM features and thus have the same generic properties and follow the same patterns for designating properties.[†] In addition, the AIXM messages can contain zero or more collections of AIXM data.

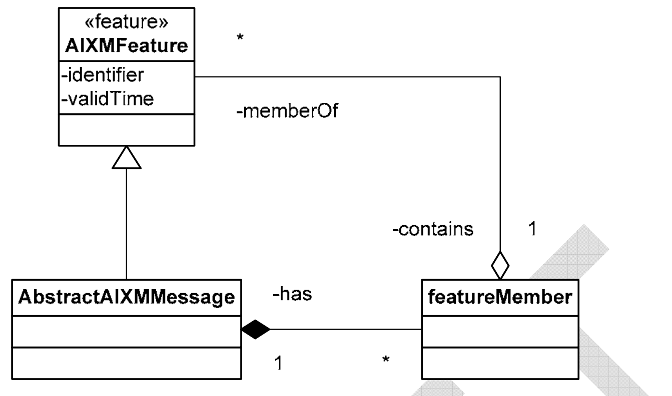


Figure 20: AICM/AIXM message model

The messaging framework illustrated in Figure 20 has these advantages:

- Defined locations for AIXM feature data so that even systems that do not understand the specific message should be able to navigate to the feature data.
- Support for any message properties.

[†] This message pattern is exactly the gml:FeatureCollection and the message framework specifications will be covered in more details when GML is introduced.

8 AICM and AIXM 5 Implementation

In this section we explain how the AICM and AIXM 5 specification is implemented to satisfy the desired requirements, design and architecture. AICM 5 is the natural evolution of the AICM 4.5 conceptual model with the addition of the new design requirements detailed in Section 7. Converting AIXM into a GML application schema is the most extensive and critical part of the AIXM specification so most of this section is devoted to discussing AIXM 5's implementation of the GML specification. For those readers unfamiliar with GML, a short introduction is provided in Section 11.

We begin this section by discussing specific GML and ISO 19100 series modeling styles that affect the structure of the AICM and AIXM data models.

8.1 GML and ISO 19100 Model Constraints and Styles

To the extent possible AICM and AIXM 5 adopt ISO 19100 and GML modeling constraints and styles. Some of the important concepts are discussed in the subsections below.

8.1.1 Implications of GML's Object-Property model

GML follows a structured object-property pattern: Features have one or more properties that encode the attributes or relationships of that feature. The content of a property is a value or another feature or complex object.

The GML object-property model changes the structure of complex AIXM properties such as the TimeTable. The structure of the TimeTable in AIXM 4.0 is:

```
<Vtt>
   <codeWorkHr>TIMSH</codeWorkHr>
   <Timsh>
       <codeTimeRef>UTCW</codeTimeRef>
       …
   </Timsh>
   <Timsh>
       <codeTimeRef>UTCW</codeTimeRef>
       …
   </Timsh>
   …
</Vtt>

Where <Vtt> is the VOR operating hours Time Table
```

Following the GML object-property model the TimeTable changes as shown:

```
<operatingHours>
   <TimeTable>
       <codeWorkHr>TIMSH</codeWorkHr>
       <timeSheets>
           <TimeSheet>
               <codeTimeRef>UTCW</codeTimeRef>
               …
           </TimeSheet>
           <TimeSheet>
               <codeTimeRef>UTCW</codeTimeRef>
               …
           </TimeSheet>
```


Figure 21: TimeTable structure in AIXM 4.0

```
        …
      </timeSheets>
      …
   </TimeTable>
</operatingHours>

Where <operationgHours> is the operating hours for the VOR.
```

In the object-property approach, the <operatingHours> property contains a complete TimeTable object. The TimeTable object contains a property called timeSheets that contains an array of TimeSheet objects. A UML diagram for the timeTable property is provided in Figure 25. Notice that the properties involved in the object-property relationship are encoded as UML associations between the parent object and the object referenced in the property.

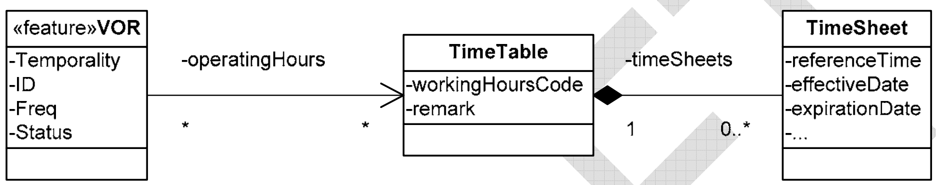


Figure 22: Object-property model for VOR operating hours.

8.1.2 Feature and Property naming conventions

The GML specification recommends three naming conventions:

- Features should be named in UpperCamelCase
- Properties and relationships should be named in lowerCamelCase
- Relationship property names should indicate the role of the relationship (e.g., Runway Direction *uses* a Runway)

The AIXM 5.0 specification follows the GML naming recommendations and includes additional rules.

8.1.2.1 AIXM Features

For AIXM features:

- Instead of the traditional 3-character abbreviations used in prior versions of AIXM, AIXM 5.0 will use long names. This means features like Rsg should be written as “RouteSegment”
- Common aeronautical abbreviations like VOR, DME, NDB, TACAN, SID and STAR can be used as AIXM feature names. For abbreviations the names should be spelled in UPPER case.

8.1.2.2 AIXM Properties

For AIXM properties:

- Meaningful names will be used.
- Common aeronautical abbreviations are allowed; the abbreviations should be spelled in UPPER case.

Figure 23: TimeTable property in AIXM 5.0

- o Abbreviations should not be first world in a property name so that the property name can be lowerCamelCase.

In AIXM 4.x property names were prefixed with their type (e.g., val for value, txt for text, etc). This convention will not be applied to AIXM 5. This convention can lead to configuration management problems if the value domain of a property changes. Consider the recent example of RNP where the value domain was changed from a code list (codeRNP in AIXM 4.x style property name) to a value domain (valRNP in AIXM 4.x style property names)

8.1.2.3 AIXM Relationships

For AIXM properties that implement relationships the names will be constructed in three parts:

- The first part indicates the role of the relationship in lowerCamelCase
- The second part is an underscore, "_", separator
- The third part of the name of the feature type that is pointed to by the relationship.

For example, the starting point of a Route Segment would be written as "startingAt_SignificantPoint".

In the AICM model the third part of the relationship name is omitted because it can be derived from the target of the relationship.

8.1.2.4 AIXM data types and domains

AIXM 5.0 will keep the naming convention and data type definitions defined in AIXM 4.x.

8.1.3 Modeling relationships

8.1.3.1 Navigability

As part of the UML modeling and the resulting AIXM XML schema, we have to make choices about relationship navigability. For example a runway and aerodrome are related. The reciprocal relationships can be expressed as:

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

In AIXM this relationship could be modeled:

- In the Runway XML element with an association to the Aerodrome
- In the Aerodrome XML element with an association to the Runway
- In both the Runway and Aerodrome XML elements.

In AIXM 4.5 this particular relationship is modeled in the Runway as an association to the Aerodome.

In our UML modeling we represent the preferred direction for the relationship using an arrow on the relationship. The arrow points to the target of the relationship and the feature at the end opposite the arrow contains the property. For the Aerodrome and Runway features this is illustrated in Figure 24.

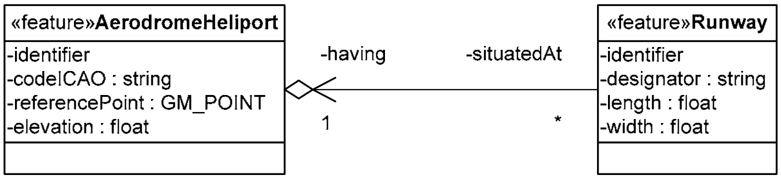


Figure 24: Illustrating relationship navigability in UML.

8.1.3.2 Encoding

A related issue is how feature relationships are encoding in the XML document. There are two options in GML:

- Encode the relationship "inline" so that the target of the relationship is explicitly contained in the XML
- Encode a remote reference using an xlink:href (see Section 8.3.3 for more on xlink:href)

These alternatives are illustrated below:

```
Inline
<AerodromeHelport>
   <having_Runway>
      <Runway>
         …
      </Runway>
   </having_Runway>
   …
</AerodromeHeliport>

Remote
<AerodromeHelport>
   <having_Runway xlink:href = "#\\Runway[…]"/>
   …
</AerodromeHeliport>
```

We propose to encode feature relationship based on their location within the feature "inclusion" hierarchy. The inclusion hierarchy describes the relative position of two features within a chain of composite and component relationships.

For a set of features like the AerodromeHeliport, Runway, Apron, Gatestand, Runway Direction and Runway Marking the inclusion hierarchy is shown in Figure 25.

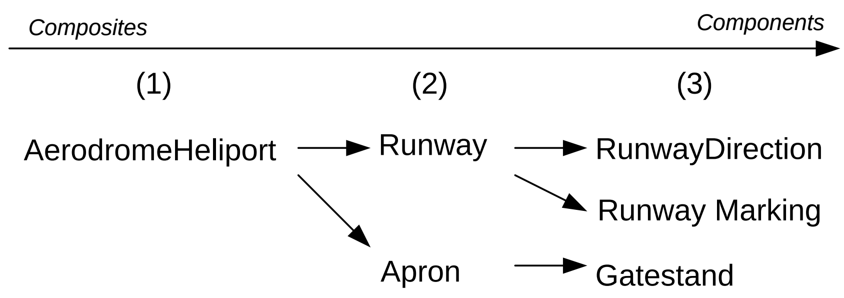


Figure 25: Partial inclusion hierarchy for the Aerodrome concept area.

Based on the hierarchy we can recommend encoding rules for feature relationships. Given two features A and B. If the relationship between A and B is modeled in A, this it is modeled:

1. Inline or remote if A is higher in the hierarchy
2. Remote if A is lower in the hierarchy
3. Remote if A is at the same level in the hierarchy.

The rationale for these rules is that we want to prevent the encoding of “higher” level features inline as a property of a ”lower” feature (e.g., an Aerodrome fully encoded inside a Taxiway).

Using these rules and the partial hierarchy in Figure 25 we can identify encoding rules for various combinations of A and B as shown in the table below:

Table 5: Examples of feature relationship encoding rules.

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

8.2 AICM UML model

This section introduces the basic features of the AICM 5 model. The complete AICM model will be made available separately as UML and a companion document.

8.2.1 AICM Component Diagram

The UML package diagram in Figure 26 shows the organization of the AICM model. The model has four levels from bottom to top:

- **Data types level** – Description of AICM-specific data types and value domains
- **Conceptual model level** – Model of aeronautical features, properties and relationships. The diagram depicts a subset of conceptual areas; the full AICM 5 model incorporates all aeronautical conceptual areas.

- **AbstractAIXM level** – Description of base AIXM Feature classes interfaces and, relationships used to model temporality and extensibility.
- **GMLProfile level** – Description of how the AbstractAIXM feature model hooks into GML so that AIXM 5 can be realized.

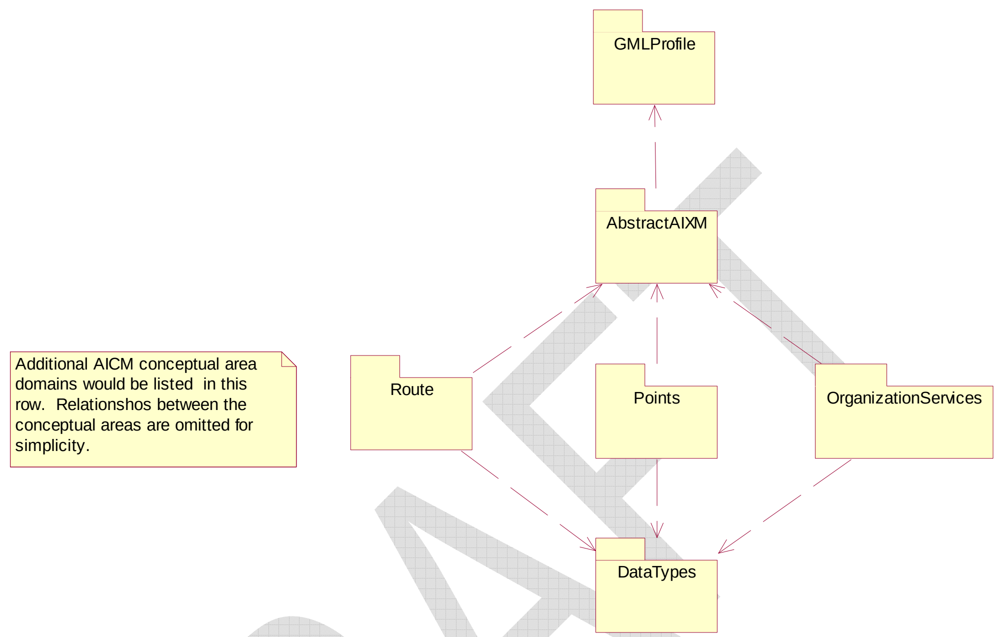


Figure 26: AICM 5 diagram showing major packages in the AICM model.

8.2.2 AbstractAIXM package

The AbstractAIXM package (Figure 27) provides the building blocks for creating AIXM Features incorporating temporality, extensibility and feature relationships. In addition, the AbstractAIXM package serves as the linking point between the AICM model and GML so that AIXM 5 can be implemented in GML.

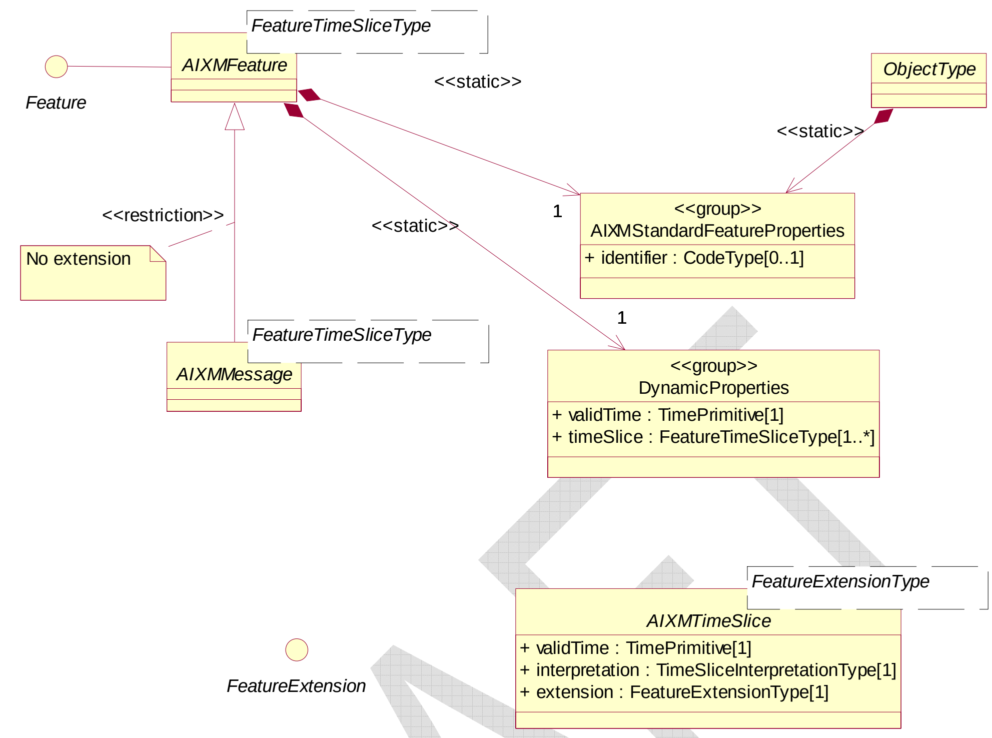


Figure 27: AICM 5 AbstractAIXM package

8.2.2.1 AIXMFeature

The key class on the diagram is the AIXMFeature. AIXMFeature is an abstract parameterized class used to instantiate all AIXM features. When instantiating an AIXM Feature from AIXMFeature, it takes a parameter called FeatureTimeSliceType indicating the FeatureTimeSlice object that is bound to the instantiated AIXM Feature.

The AIXMFeature includes two property groups: AIXMStandardFeatureProperties and DynamicProperties. Property groups are indicated with the <<group>> stereotype and they are always related to other classes through composition. The properties contained in <<group>> classes are assumed to be merged into the aggregating class. So in this example, AIXMFeature actually has three properties:

- identifier (from AIXMStandardFeatureProperties)
- validTime (from DynamicProperties)
- timeSlice (from DynamicProperties)

The <<group>> classes are used in the AICM and AIXM model so that commonly used properties can be grouped and reused for different classes.

Note also that the composition relationships are for AIXMStandardFeatureProperties and DynamicProperties are stereotyped with <<static>>. The <<static>> stereotype is used to indicate that these properties exist at the Feature level and are not dynamic.

The AIXMFeature class realizes the Feature interface.

The AIXMFeature parameterized class takes in a FeatureTimeSliceType parameter that is a TimeSlice class derived from the abstract AIXMTimeSlice.

8.2.2.2 AIXMTimeSlice

The AIXMTimeSlice class is also a parameterized class that is used to instantiate all AIXM Feature TimeSlice objects. When instantiating an AIXM Feature TimeSlice, it takes a parameter called FeatureExtensionType which is a specific Feature Extension interface.

The AIXMTimeSlice abstract class has three properties:

- validTime
- interpretation
- extension

The interpretation property is based on the AICM temporal model and is used to classify the TimeSlice as baseline, version, temporary delta, permanent delta or snapshot. The extension property is an interface to one or more optional extensions.

8.2.2.3 AIXMMessage

AIXMMessage is an abstract parameterized class derived from AIXMFeature. The AIXMMessage restricts AIXMFeature by removing the extension property. Using this construction AIXMMessage is also an AIXMFeature: AIXMMessage follows the AIXMFeature property-object pattern and can contain a collection of message timeslices.

8.2.2.4 FeatureExtension

FeatureExtension is a common interface specification that all extensions must realize.

8.2.3 Example Instantiation: VOR

Figure 28 shows how the AbstractAIXM package can be used to instantiate a specific AIXM Feature – the VOR.

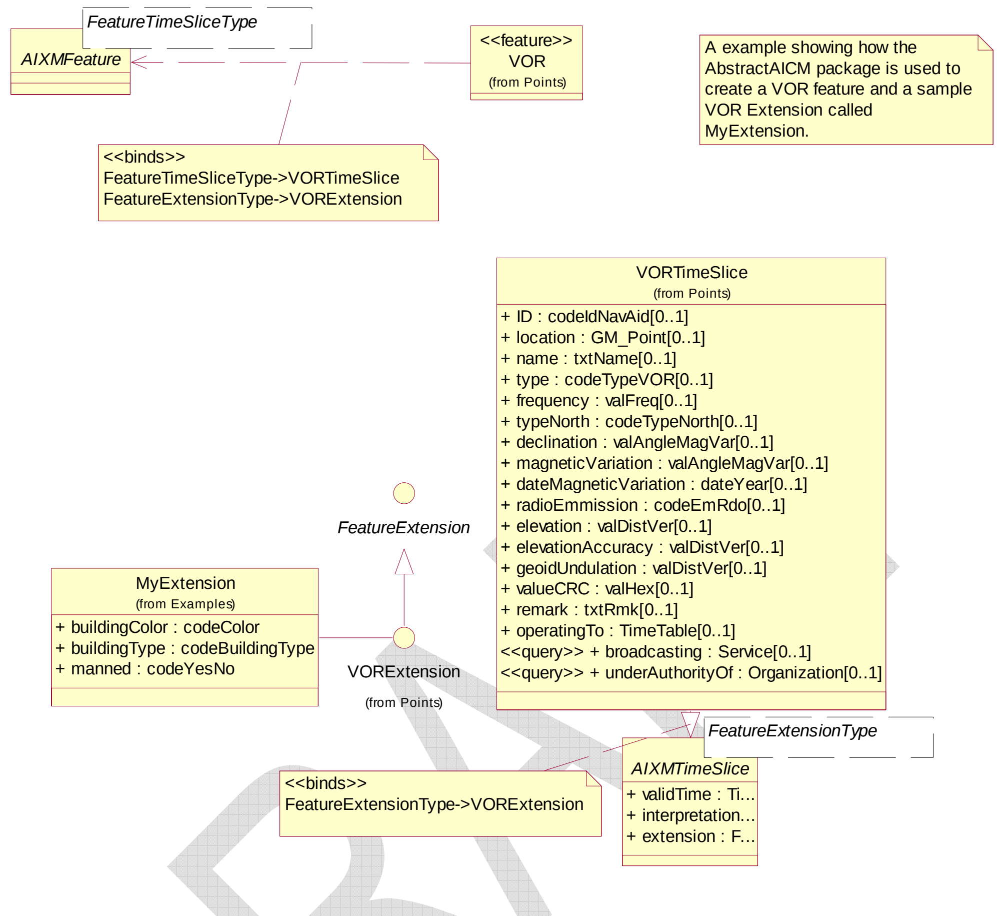


Figure 28: Example of the AbstractAIXM package instantiated for a VOR feature.

8.2.3.1 VOR

The VOR class instantiates the abstract AIXMFeature parameterized class and binds FeatureTimeSliceType to VORTimeSlice. To improve model interpretation features are stereotyped with <<feature>>. The VOR feature contains all of the properties of the AIXMFeature class including: identifier, validTime and a collection of VORTimeSlices.

Note that the VOR Feature does not contain any specific VOR properties. This is because all VOR properties are assumed to be dynamic so they are included as part of the VORTimeSlice.

8.2.3.2 VORTimeSlice

The VORTimeSlice instantiates AIXMTimeSlice parameterized class and binds FeatureExtensionType to the VORExtension interface. The VORTimeSlice extends the base AIXMTimeSlice class by including all of the VOR properties from the AICM 4.5 model. Each property is made optional indicating that the any of the properties may change in time.

The VORTimeSlice class includes two inline relationships:

- broadcasting is a relationship to a broadcasting Service.
- underAuthorityOf is a relationship to an Organization.

Both relationships are stereotyped with <<query>> indicating that the relationship is instantiated using a query against the target feature properties.

8.2.3.3 VOR extensions

The VORExtension interface derives from the abstract FeatureExtension interface and all custom VOR extensions must realize the VORExtension interface. In this diagram, an example MyExtension class realizes the VORExtension interface and includes custom properties for representing characteristics of the VOR equipment building.

8.3 AIXM XML Schema

8.3.1 Developing a GML Profile for AIXM

Profiling simplifies AIXM's implementation of GML by restricting AIXM to the limited set of GML features required by AIXM. The goals of the GML Profile for AIXM are:

- Remove pre-GML 3.1.1 deprecated elements
- Restrict GML to a single choice where multiple options are provided.
- Limit GML to selected geometry and temporal properties
- Augment the GML 3.1.1. Profile with additional properties that are likely to be part of GML 3.2 (ISO 19136)

The GML Profile for AIXM was generated using the instructions and tools provided in the GML 3.1.1 specification, Annex G [8]. After analyzing AIXM geometry requirements we identified the following list of allowable geometry values:

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

The following temporal features are included in the GML Profile:

- DynamicFeature
- TimeInstant
- TimePeriod

After the XML Stylesheet tools are used to extract the GML Profile, we further restricted the profile by following the methodology and rules contained in Section 22.1 of the GML 3.1.1 Specification [8]. As part of this customization we deleted global types, deprecated elements and local optional elements that were not needed for the AIXM application schema.

8.3.2 Feature identification in GML

8.3.2.1 Identification in GML 3.1.x

In GML 3.1.x all GML objects have an optional gml:ID. The gml:ID must be unique within an instance document. Ideally the gml:IDs are globally unique so that features are unique across distributed systems.

8.3.2.2 Expected GML 3.2 evolution

The GML 3.2 specification is expected to include the following:

- gml:ID will be mandatory and of type XSD type ID
- gml:ID will be the local feature identifier. That is, the gml:ID will be unique within the XML instance document.
- An optional property called gml:identifier will be used to provide globally unique identification on features. The gml:identifier will be a combination of context (e.g., namespace) along with an identification string.

8.3.2.3 AIXM 5 implementation

AIXM 5 will incorporate the recommended gml:identifier property on all features. The property will be used to globally identify aeronautical features. We recommend that the gml:identifier include the namespace of the data provider as well as an artificial key that is unique to the data provider.

8.3.3 Feature Relationships in GML

In GML associations are created using properties. Association properties can point to the target object by either including the target object inline or through an xlink:href. The xlink:href specification supports the use of XPointer and XPath to identify XML elements in local or remote resources. For more on XLink, XPointer and XPath see http://www.brics.dk/~amoeller/XML/linking/,

At its most basic XLink has a format “XPointer#XPath” where

- XPointer points to a remote resource
- XPath locates a specific XML element within the context specified by the XPointer.

This concept is illustrated in Figure 29.

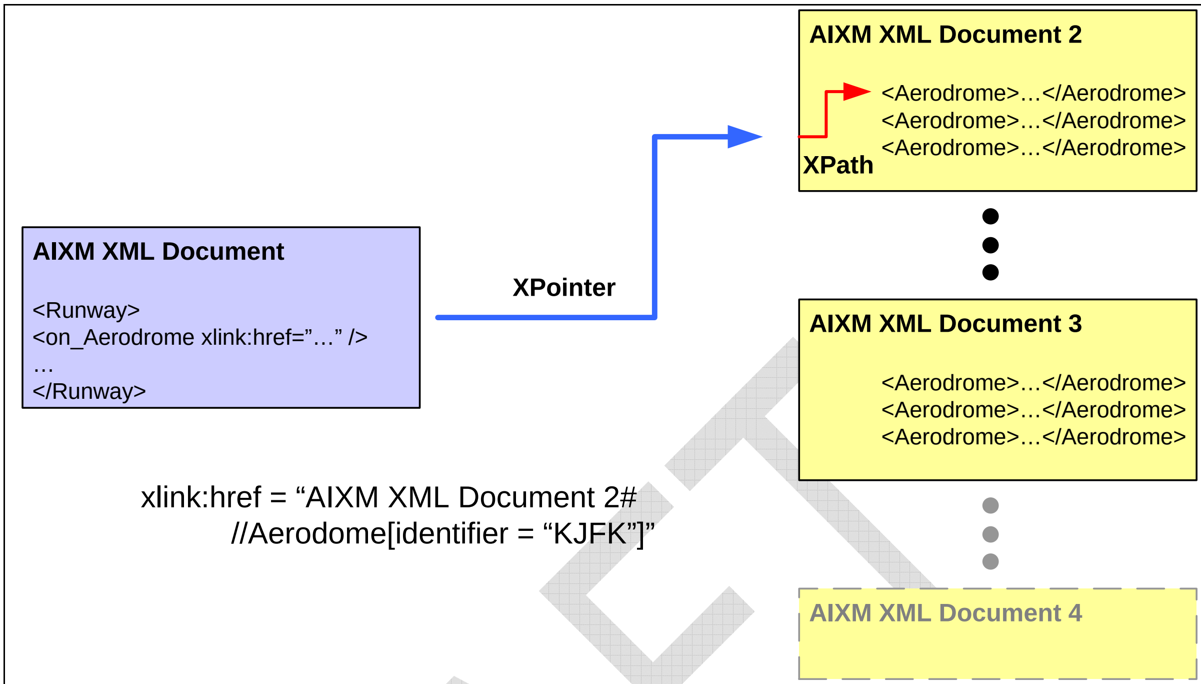


Figure 29: Example of a remote xlink:href pointing to an aerodrome in a separate resource.

Xlink accommodates the <<query>> design requirement identified and discussed in Section 7.2. Xlink works directly and simply for artificial identifiers coded using the Feature identifier property. In addition, Xlink can accommodate natural identifying property in a complex by standard syntax.

For example a possible xlink to an Aerodrome located in the FAA's NASR database might look like:

```
xlink:href="http://www.faa.gov/nasr/database.xml#
   //aixm:AerodromeHeliport[
      aixm:icaoCode="EDDF" and
      aixm:position/gml:Point/gml:pos='50.0333 8.5704']"
```

A more complex example is the natural identification for a Runway. In this case we logically assume that the associations are inline when the xlink is created:

```
xlink:href='http://www.faa.gov/nasr/database.xml#
   //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"
      ]
   ]
'
```

In the Runway example, when constructing the xlink the AerodromeHeliport reference is logically assumed to be inline.

The xlink mechanism provides a standardized syntax for representing relationships between features using queries. As a result the XML document can accommodate the use of artificial identifiers and/or flexible use of natural properties to identify features.

8.3.3.1 Resolving Xlinks

With xlink encoding, the association property points to the location of the associated feature. Xlinks can be either local or remote.

A local xlink resolves to a location within the same AIXM XML document. Local xlinks are the easiest for AIXM document consumers to resolve since all the features and associated features are included within the same document. Of course, local xlinks can greatly increase the size of the transmitted documented. Depending on the application requirements and the data requirements this can be a problem. For example, for a data file containing a snapshot of an entire AIM database; local xlinks would be most efficient since all of the features must be included in the AIXM document anyway.

Remote xlinks can be used to reference AIXM features that are not included in the AIXM document. In order for an AIXM Document consumer to be able to resolve a remote xlink the consumer must either 1) already know about the features being remotely referenced or 2) the AIXM document supplier should provide a mechanism for the consumer to request information about the remote features.

An AIXM document may include both local and remote xlinks depending on the criticality of the data. Also there may be business cases where transmitting documents must always include locale copies of associated features. One can imagine safety critical applications where all referenced features must be transmitted so that the state of the system is unambiguously known.

Note that the mechanism used to resolve remote xlinks is beyond the scope of this standard. It is assumed that service level agreements between AIXM data providers and AIXM data consumers will document the data transmission approach used as well as protocols for requesting information on remotely referenced features.

9 Aeronautical information exchange use case

The purpose of this section is to present a scenario that uses AIXM for data exchange between systems to illustrate how AIXM 5 framework supports a range of aeronautical system to system exchange requirements.

This section is intentionally left blank. It is be completed after the AIXM 5 Public Design Review to be held February 7-8, 2006.

10 AICM/AIXM Namespace Convention

The AIXM namespace convention is based on the naming convention adopted by the Aerospace Operations division of the United States Department of Defense (DOD) [14].

The convention specifies:

- Use URL syntax with the http URI scheme
- Separate hierarchical components of the namespace with a slash "/"
- Separate subcomponents with periods "."
- Use lowerCamelCase and unreserved characters

The namespace should have the following syntax:

http://organization//unit/system/resource

- Organization is the fully qualified name of the organization providing the data
    - Us.gov.dot.faa
    - com.jeppesen
- Unit is the subdivision within the organization that is providing the data
    - airTrafficOrganization.aeronauticalInformationManagement
- System is the database or application that generated the data
    - NASR
- Resource is a system component or operation used
    - eNASR

So a namespace for the electronic NASR interface to the FAA's NASR (National Airspace System Resources) database might be:

http://us.gov.dot.faa/airTrafficOrganization/aeronauticalInformationManagement/NASR/eNASR

11 GML Introduction

This introduction is excerpted from “AIXM Profile of GML” presented at the EUROCONTROL AISTEC 10 meeting in April 2005 [18].

Geography Mark-up Language (GML) is an internationally adopted standard for exchanging geographical features using XML [7]. GML was developed by the Open Geospatial Consortium (OGC), an international consortium of companies, government agencies, and universities participating in a consensus process to develop publicly available geo-processing specifications [19]. In addition, GML is compliant with the ISO Technical Committee 211 19100s series standards [8]. Amendment 33 to ICAO Annex 15 asks for compliance with a number of standards from this ISO series for terrain and obstacle database products.
GML includes an extensive set of XML schemas for expressing simple geometries like:

- Points
- Lines
- Polygons
- GML also supports complex geometries, topologies and temporal data like:
- Surfaces
- Curves and splines
- Directed graphs and networks
- Observations
- Coverages

In addition, the GML specification includes rules for incorporating these geometries into GML Feature Types that represent real world objects. The figure below shows a simplified view of an Aerodrome and how the aerodrome’s attributes might be mapped to GML.

Aeronautical entity	LFJB Mauleon (465410N, 000452W)
Classified in GML as…	GML Feature :: Aerodrome GML Property :: Name = “Mauleon” GML Property :: CodeID = “LFJB” GML PointProperty :: Airport Reference Point = “465410N, 000452W”
Geometry represented in GML as…	<gml:Point> <gml:pos>46.90278 0.08111</gml:pos> </gml:Point>

Figure 30: Mapping an aerodrome to GML

Figure 31: Representing real world aeronautical features in GML.

As illustrated in Figure 30, GML includes a model for representing:

- **Features** : abstraction of a real world phenomena [8]
- **Properties** : describe some aspect of the Feature
- **Geometry Properties** (e.g., Point Properties): represent a geometric aspect of the feature. [8]

GML provides the building blocks for representing features, properties and geometries. In addition, GML includes an extensive set of predefined geometry types like points, lines and polygons. However, GML does not contain specific geographic entities. That is, you will not find a road or runway defined in GML. Instead GML provides a standard framework that can be used to *define* a road or runway in a consistent way. By using the GML framework, specific geographic entities like the road and runway can be generically interpreted by any tool that can understand GML.

When a domain uses the GML rules to create a specific vocabulary of geographic objects (e.g., aeronautical objects like runways and navigation aids) that vocabulary is described in a **GML application schema**.

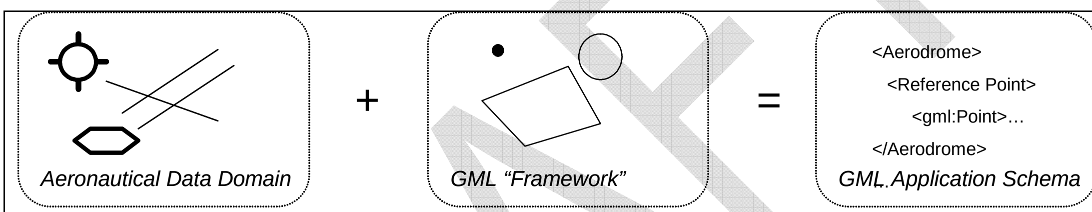


Figure 32: Definition of a GML Application Schema

11.1 Definition of a GML Profile

GML was designed to meet the needs of virtually all geographic systems. Because of this, GML is a complex standard suitable for representing any type of geography feature. The GML standard recognizes that specific domains may only need a subset of GML features; therefore the GML standard includes provisions for adopting subsets of GML for specific applications [8]. The subset of GML appropriate for a specific application is called a **GML profile**.

A GML profile makes it easier to apply GML for a specific application because the profile:

- Restricts the application to use a subset of geography features
- Reduces the complexity by simplifying the GML data model

For example, AIXM may not need to include support for GML curves and splines, so the AIXM profile of GML can specify that GML curves and splines are not valid when using GML in AIXM.

11.2 Benefits of incorporating GML into AIXM

11.2.1 Compliance with established international standards

AIXM is emerging as an international standard for exchanging aeronautical data, as evidenced by its increased use in aeronautical information systems developed in Europe, United States, Japan and the rest of the world. Successful transition of AIXM into an international standard with broad government and vendor support requires that AIXM leverage existing/emerging data standards that provide a demonstrated value for aeronautical data providers and consumers.

Amendment 33 to ICAO Annex 15 states that:

> (10.5.1) "*To allow and support the interchange and use of sets of electronic terrain and obstacle data among different data providers and data users, the ISO 19100 series of standards for geographic information shall be used as a general data modelling framework.*"

As obstacles are part of the AIXM Scope, it is desirable that the relevant ISO standards considered by ICAO are considered in the AIXM development. This will guarantee that States using AIXM are de-facto compliant with the ICAO requirements.

Up to now, AIXM has relied on a custom model for representing geographical data. For example, and airspace border is described as a series of vertex, which are characterised by position (lat/long), datum, elevation, accuracy, etc. In order for a system to be able to process this information, custom code has to be developed. GML offers a standardized and internationally accepted model for describing geographical data.

11.2.2 Cost savings in information system development

By incorporating GML, AIXM will be able to leverage existing COTS tools and systems that can process, visualize and exchange GML data:

- It would be possible to build an AIXM XML document that can be ingested into generic GML viewers such as those produced by COTS vendors as well as custom visualization tools such as EUROCONTROL's SkyView.
- It would be possible to leverage other OGC standards such as the Web Feature Service (WFS)[‡] to develop system to system interfaces for data exchange.

As a consequence, incorporating GML into AIXM will reduce system costs and development time by enabling system developers to leverage GML-compatible COTS products

11.2.3 Enhanced capabilities for existing AIXM systems

In order to make use of the geometrical descriptions contained in AIXM, systems such as the EAD convert this information into a full geometrical model, such as Oracle Spatial. For example, if the geometry of an airspace is described as the union of two other airspace parts in AIXM, the EAD will build an internal geometrical representation for that airspace. A similar process takes place in any other systems that use AIXM airspace association data. This might be seen as an unnecessary multiplication of the

[‡] See also AISTEC-10/IP3, Agenda Item 13, "Geospatial Interoperability – WFS".

effort. The incorporation of GML into AIXM would allow systems such as the EAD to export the already "digested" information about airspace associations, in the form of GML geometries. This could then be used directly by any interested system which supports the GML standard.

11.3 Temporality in GML

GML includes two content models that can be used to encode temporal changes to features [7,8]:

- Dynamic Features
- Observation model

The GML Dynamic Features content model is used to describe features that change in time. In addition GML includes an Observation content model that can be used to encode measurements made on a feature. More information can be found in [7,8].

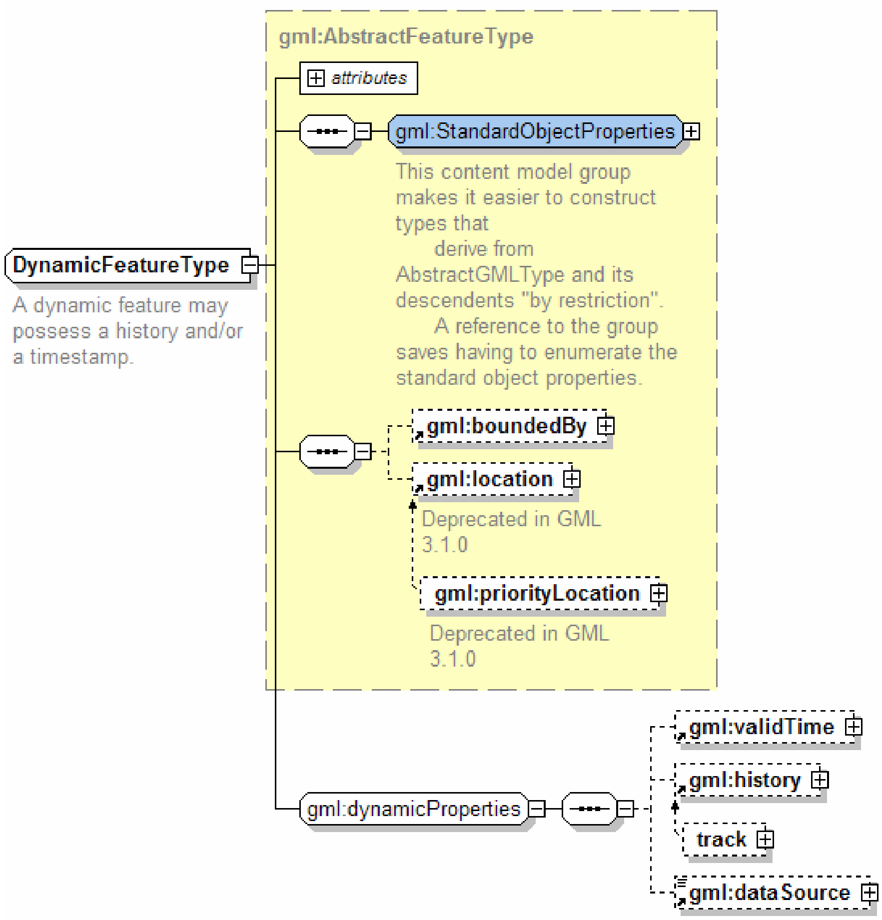


Figure 33: GML Dynamic Feature Model

11.3.1 Dynamic Features [6]

The GML Dynamic Features content model is used to describe features that change in time. In GML, dynamic features must derive from gml:DynamicFeatureType. The gml:DynamicFeature type adds temporal properties to the standard gml:AbstractFeatureType. The definition of the gml:DynamicFeature is depicted in Figure 33.

The gml:validTime element contains a time instance or time period for describing the state of a feature and its properties. The gml:history element contains a set of gml:TimeSlices and it is used to model dynamic properties in accordance with the TimeSlice model.

11.3.1.1.1 GML Snapshot model

The GML Snapshot model for dynamic features represents the state of a feature at a time instance or over a time interval. In the GML Snapshot model the time interval is encoded along with all of the feature properties. According to Galdos [6] GML Snapshot:

- Is typically used to update a feature that has already been defined.
- May contain redundant information since all properties – even time invariant properties – must be re-specified for every snapshot instance.
- Each snapshot instance of a feature must have a unique gml:id if the snapshots are included in the same instance document (gml:identifier could be used to assert that the feature instances actually correspond to the same feature.).

The excerpt from [6] shows how the SnapShot model could be used to encode the state of the Portugal FIR airspace feature at three time intervals:

```
<aixm:Airspace gml:id="id1">
 <gml:identifier    codeSpace="urn:UUID:">d6cd9be0-15d4-11da-
8cd6-0800200c9a66</gml:identifier>
 <gml:validTime>
  <gml:TimePeriod>
   <gml:beginPosition>2004-12-12T12:12:12</gml:beginPosition>
   <gml:endPosition>2005-08-31T23:59:59</gml:endPosition>
  </gml:TimePeriod>
 </gml:validTime>
 <aixm:txtName>PORTUGAL FIR</aixm:txtName>
 <aixm:lowerLimit>1000</aixm:lowerLimit>
 <aixm:upperLimit>4000</aixm:upperLimit>
 <gml:dataSource xlink:href="SomeSourceResourceIdentifier"/>
</aixm:Airspace>

<aixm:Airspace gml:id="id2">
 <gml:identifier    codeSpace="urn:UUID:">d6cd9be0-15d4-11da-
8cd6-0800200c9a66</gml:identifier>
 <gml:validTime>
  <gml:TimePeriod>
   <gml:beginPosition>2005-09-01T00:00:00</gml:beginPosition>
   <gml:endPosition>2005-09-31T23:59:59</gml:endPosition>
  </gml:TimePeriod>
 </gml:validTime>
 <aixm:txtName>PORTUGAL FIR</aixm:txtName>
 <aixm:lowerLimit>2000</aixm:lowerLimit>
 <aixm:upperLimit>4000</aixm:upperLimit>
 <gml:dataSource xlink:href="SomeSourceResourceIdentifier"/>
</aixm:Airspace>

<aixm:Airspace gml:id="id3">
```

```
 <gml:identifier    codeSpace="urn:UUID:">d6cd9be0-15d4-11da-
8cd6-0800200c9a66</gml:identifier>
 <gml:validTime>
  <gml:TimePeriod>
   <gml:beginPosition>2005-10-01T00:00:00</gml:beginPosition>
   <gml:endPosition indeterminatePosition="unknown"/>
  </gml:TimePeriod>
 </gml:validTime>
 <aixm:txtName>PORTUGAL FIR</aixm:txtName>
 <aixm:lowerLimit>1000</aixm:lowerLimit>
 <aixm:upperLimit>4000</aixm:upperLimit>
 <gml:dataSource xlink:href="SomeSourceResourceIdentifier"/>
</aixm:Airspace>
```

11.3.1.2 TimeSlice model

The GML TimeSlice model can be used to encode temporal properties of a feature over time. A canonical example of the TimeSlice model is modeling an aircraft in flight: Many of the aircraft properties would remain static and the GML TimeSlice model could be used to specify a Flight position property over time. According to [6] GMLTimeSlice:

- Allows time varying feature properties to be expressed as properties of a TimeSlice object. This approach may be more economical because only time varying properties must be repeated.
- A single feature instance can contain several TimeSlices thus avoiding the gml:id issue that affects the Snapshot model.
- TimeSlices can overlap allowing two TimeSlices to affect different feature properties.

An excerpt from [6] shows how the TimeSlice model could be used to model the time varying properties of the Portugal FIR airspace feature:

```
<aixm:Airspace gml:id="id1">
 <gml:identifier    codeSpace="urn:UUID:">d6cd9be0-15d4-11da-
8cd6-0800200c9a66</gml:identifier>
 <aixm:effectivity>
  <aixm:Status>
   <gml:validTime>
    <gml:TimePeriod>
     <gml:beginPosition>2005-09-
01T00:00:00</gml:beginPosition>
     <gml:endPosition>2005-09-31T23:59:59</gml:endPosition>
    </gml:TimePeriod>
   </gml:validTime>
   <aixm:lowerLimit>2000</aixm:lowerLimit>
   <gml:dataSource xlink:href="SomeResourceIdentifier"/>
  </aixm:Status>
  <aixm:Status>
   <gml:validTime>
    <gml:TimePeriod>
     <gml:beginPosition>2004-12-
12T12:12:12</gml:beginPosition>
     <gml:endPosition indeterminatePosition="unknown"/>
    </gml:TimePeriod>
   </gml:validTime>
   <aixm:txtName>PORTUGAL FIR</aixm:txtName>
   <aixm:lowerLimit>1000</aixm:lowerLimit>
   <aixm:upperLimit>4000</aixm:upperLimit>
   <gml:dataSource xlink:href="SomeResourceIdentifier"/>
  </aixm:Status>
 </aixm:effectivity>
</aixm:Airspace>
```

11.3.2 Combined Snapshot and TimeSlice Model

Both Snapshots and TimeSlices can exist within the same GML feature instance. This could enable a feature instance document to specify the static state of a feature using the gml:Snapshot model along with a set of dynamic property changes using the gml:TimeSlice model.

11.3.3 Observation model

The gml:Observation model is another approach for representing changes to features. In contrast to the direct temporal representation of the dynamic feature content model, the Observation Model provides a way to indirectly specify changes to features. The observation model was originally intended to model measurements taken by sensors. The observation model includes:

- Properties about the observation event
- Reference to the feature being observed
- Property values observed with the observation event.

The observation model is illustrated below:

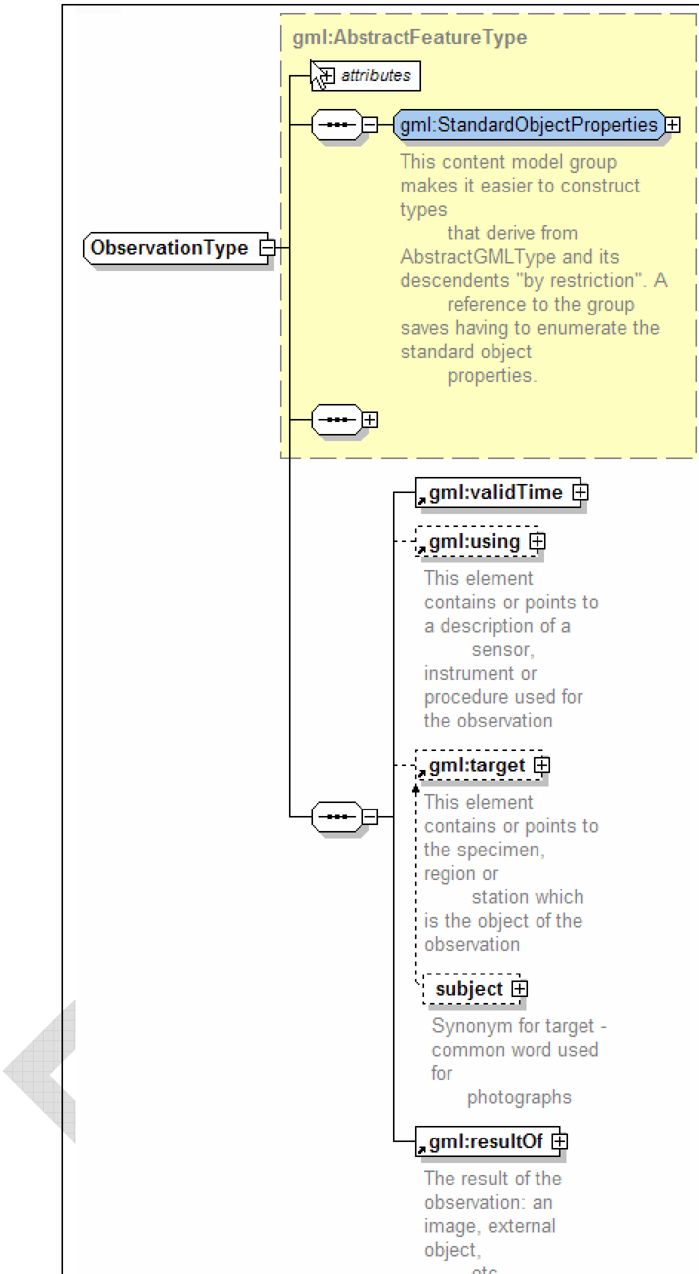


Figure 34: GML Schema model for Observations

In a sense the observation provides a way to encode information *about* a temporal event on a feature rather than reporting state changes of a feature. The level of indirection associated with the Observation model might be useful for applications like NOTAM that describe temporal events that affect one or more features.