Aeronautical Information Exchange Model (AIXM)

Data Modelling Conventions for AIXM 5

For all inquiries, please contact:
Brett BRUNK - brett.brunk@faa.gov
Eduard POROSNICU - eduard.porosnicu@eurocontrol.int

Edition No.	Issue Date	Author	Reason for Change
0.1	2006/06/22	Brett Brunk	First Edition

Contents

1 Scope

1.1 Introduction

This document presents the data modeling conventions used in AIXM 5. The document consists of two sections:

- A data modeling style guide including naming conventions and abbreviations
- A profile of the UML based on the ISO 19100 series geographic standards.

1.2 Background

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

1.3 References

1. Geographic Information – Spatial Schema. ISO 19107. First Edition, 2003-05-01
2. Geography Markup Language (GML). ISO/TC 211/WG 4/PT 19136 OGC GML RWG. Committee Draft. 2004-02-07.
3. UML 2.0 In a Nutshell. Dan Pilone. O'Reilly Media Inc. 2005.

2 Data model style guide

2.1 Modeling terminology

The following terminology is used when discussing components of the AIXM model.

Baseline	A component of the AIXM temporal model. The state of a feature and all the feature properties as a result of a permanent change. The Baseline state of a feature also exists when a feature is initially created. The Baseline state lasts until the next permanent 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". In AIXM lowerCamelCase is used for represent attributes and relationships.
Object	An aeronautical entity that exists within the context of another feature or object. Objects do not have any of the temporal characteristics. Generally objects are used to represent complex properties of a feature. For example, a VOR feature may contain a TimeTable object that describes the operating hours for the VOR
Permanent Delta	A component of the AIXM temporal model. A set of properties that have changed or will change permanently. The permanent delta will result in a new Baseline.
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.
Temporary Delta	A component of the AIXM temporal model. A set feature property values that are effective for a limited time. For examples the operatingStatus of a VOR may change from Operational to Offline for two days.
UpperCamelCase	A naming convention style in which words are concatenated and the first letter of each word is capitalized. For example "SignificantPoint." In AIXM UpperCamelCase is used for feature, objects and group names.
Version	A component of the AIXM temporal model. The state of a feature and all the feature properties during the time period between two changes.

In AICM we use terms such as feature, attributes and relationship to describe the aeronautical data model. These concepts are illustrated in the class diagram shown in Figure 1. This example shows a subset of the Aerodrome model by depicting the ILS (Instrument Landing System) and ILS Glide Path component features. The features have properties that describe and characterize the feature. For example, the ILSGlidePath has a frequency, slope and referenceDatumHeight properties. A relationship associates the ILS to the ILSGlidePath component. In this case the relationship shows that the ILS has ILSGlidePath.

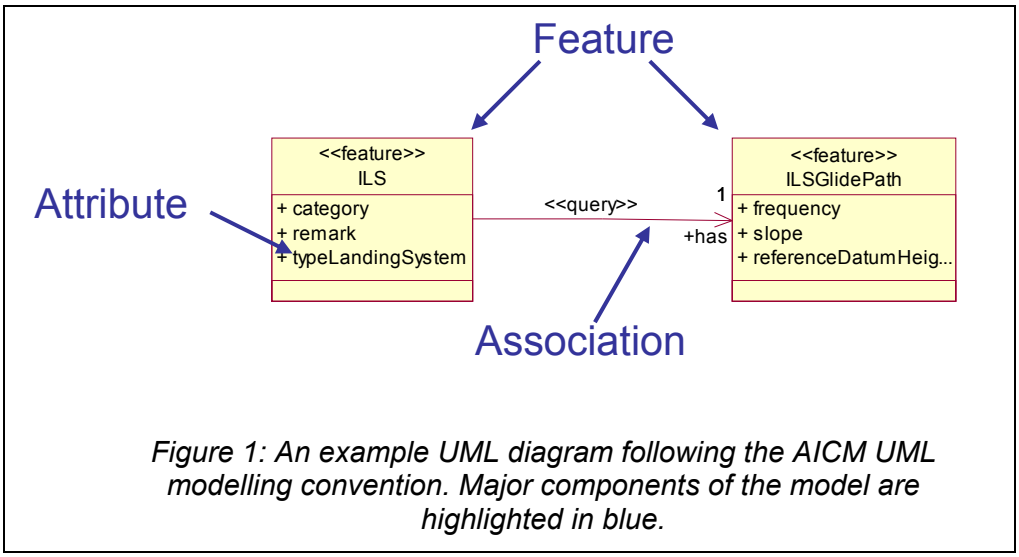

Figure 1: An example UML diagram following the AICM UML modelling convention. Major components of the model are highlighted in blue.

2.2 Feature and Property naming conventions

AIXM has the following general rules for class, property and relationship:

- All names must use UK English.
- All terms must be spelled out, except for the explicit set of AIXM abbreviations listed in Section 2.3.

2.2.1 Classes

Additional rules for classes are provided below:

- All classes should be stereotyped with one of the following:
  - <<feature>>, <<object>>, <<modelGroup>>, <<DataType>>, <<Enumeration>>, <<CodeList>>

- Instead of the traditional 3-character abbreviations used in prior versions of AIXM, AIXM 5 uses long names. This means features such as Rsg should be written as "RouteSegment"
- Names should be in UpperCamelCase.

2.2.2 Attributes

Additional rules for attributes are provided below:

- Meaningful names will be used.
- Names should be in lowerCamelCase.

In AIXM 4.x property names were prefixed with their type (e.g., val for value, txt for text, etc). This convention is discontinued in AIXM 5. Data type prefixes 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).

2.2.3 Associations

Associations conventions are:

- Meaningful role names will be used.
- All relationships will have unidirectional navigability.
- As a minimum the role name for the class being navigated to must be provided.
- Role names are in lowerCamelCase.
- Relationships between features should be stereotyped <<query>>.

2.3 Symbols and abbreviations

To improve understanding, the AIXM 4.x convention of using abbreviations and 3-letter codes for aeronautical feature names has been abandoned. Instead we use descriptive naming in Upper and Lower Camel Case for features and properties. In most cases abbreviations have been removed from the model. The table below lists abbreviations that may be used in the AIXM 5 UML model.

Abbreviation	Meaning
CRC	Cyclic Redundancy Check
DME	Distance Measurement Equipment
FATO	Final Approach and Take Off Area
IAP	Instrument Approach Procedure
IATA	International Air Transport Association
ICAO	International Civil Aviation Organization
ID	Identification
IFR	Instrument Flight Rules
NDB	Non-Directional Beacon
SID	Standard Instrument Departure
STAR	Standard Instrument Arrival
TACAN	UHF Tactical Air Navigation Aid
TLOF	Touch Down Lift Off Area
VASIS	Visual Approach Slope Indicator System
VFR	Visual Flight Rules
VOR	VHF Omnidirectional Radio Range

3 Profile of Unified Modeling Language (UML)

UML is a visual language for describing relationships, behavior and high level ideas. It was originally intended for software engineering, but today UML is used for business process modeling, data modeling, requirements modeling and other activities [3].

UML offers a number of advantages as a modeling language. It is the de facto modeling standard with a well defined model structure and semantics. It has good industry support in the form of UML modeling tools and tools that can convert UML into useful products like XML schemas, databases and computer programs.

The UML conventions used in AIXM 5 are based on ISO 19107 and GML. At a later date we intend to evaluate using ISO19103 as the basis for the AIXM 5 UML model.

3.1 Class attributes, operations and associations

Class attributes and associations are modeled in compliance with the UML notation guide. Specifically attributes and associations have the following form [3]:

Visibility / stereotype name : type multiplicity

For AICM 5 the following values are used:

- Visibility – Public
- / - not used
- Stereotype – not used
- Name – name of the attribute in lowerCamelCase
- Type – attribute type
- Multiplicity – if not specified assume optional, [0..1]

3.1.1 Attributes

Attribute multiplicity is always optional, [0..1]. Other values for multiplicity should be handled by explicitly modeling the relationship using associations.

Associations can be encoded as attributes of a class. In the AIXM 5 model, only associations to <<DataTypes>>, <<CodeList>> and <<enumeration>> relationships are encoded as class attributes.

3.1.2 Operations

Since AIXM 5 is a conceptual data model for information exchange, UML operations are not supported.

3.1.3 Associations

Only strong aggregation (e.g., composition) is modelled in AIXM. Composition implies strong ownership and coincident lifetime of the parts by the whole [3]. When the whole is removed the part is also removed. For example, TimeTables are composed of TimeSheets. When the TimeTable is deleted, the TimeSheets also are deleted. Composition is modelled with a filled diamond symbol at the end of the association line that connects to the composed (parent) class.

Association navigability is indicated by arrows on association lines between two classes. In AIXM 5 every relationship must have navigability specified.

Associations must be used to represent relationships between <<features>>, <<objects>> and <<modelGroups>>.

3.2 Stereotypes

Classes in the AIXM UML diagrams are stereotyped to extend the UML model to describe additional information about the classes. Most of the stereotypes used in the AIXM UML have the same meaning as described in ISO 19107, these are:

- <<Abstract>> - represented in UML by italicized class name. Indicates that the class cannot be instantiated directly [1].
- <<DataTypes>> - encapsulate data and must be strongly aggregated into another class [1].
- <<Union>> - a type that consists of one and only one of several alternatives [1]
- <<CodeList>> - similar to an enumeration used to indicate a list of possible values that can be expanded [1].

In addition to this set of ISO 19107 stereotypes, AIXM includes stereotypes used in the GML specification:

- <<modelGroup>> - represents a set of properties that are common to more than one AIXM object or feature. The set of common properties is defined in a model group that must be strongly aggregated into another class [2].

Finally AIXM defines four specialized stereotypes:

- <<feature>> - denotes classes that represent features.
- <<object>> - denotes classes that represent objects.
- <<enumeration>> - indicates a list of values that cannot be expanded without changes to the AIXM data model.
- <<query>> – Indicates a relationship implemented as a query in AIXM. Query relationships are dynamic. A <<query>> relationship is specified by selecting a target feature by query. Executing the query returns a result set which shall contain exactly 1 target feature. The <<query>> stereotype must always be used when describing relationships between <<features>>.
- <<static>> - Indicates that the target class is embedded in the static portion of the <<feature>>. The <<static>> stereotype may only be used to characterized relationships from a <<feature>> to an <<object>> or <<modelGroup>>. The normal situation is for the target class to be included as part of the feature Time Slice.

3.3 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 encoded:

- 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 Aerodrome.

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 2.

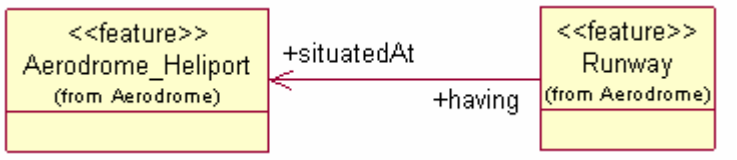

Figure 2: Illustrating relationship navigability in UML.

3.4 Collection Types

To be completed: Describe conventions used to encode ordered and unordered arrays.