A424-A UML MODEL TECHNICAL REPORT

Résumé / *Synopsis* :

In the frame of A424A, which merge A424-19 and NDBX, a new UML model of aeronautical data is created.
The objective of this document is to provide a description of the design of this new UML model. It explains how the UML model have been designed, which hypothesis were taken, what are the differences between A424-19 and this new UML model, and what are the open points and the possible improvements.

Mots Clés / *Key Words* : A424, A424-A, ULM modeling, NDB, NDBX

	Nom / *Name*	Date / *Date*	Signature / *Signature*
Rédigé par / *Written by*	RIFFLART P.F.	21/01/2011	ABSENT ETIENNE
Vérifié par / *Checked by*	BRUNEAUX P.	21/01/2011	
Approuvé par / *Approved by*	MORELL J.Y.	21/01/2011	

Client / *Customer* : EYAN2
Référence externe / *External reference* :
Cahier des Charges / *Statement of Work* : WD803257

Modifications / *Revisions*			
Edition / *Issue* Date / *Date*	**Réf. fiche de relecture / *Proof reading form ref.***	**Page / *Page***	**Objet des évolutions / *Reasons for revision***
Ed 1 21/01/2011	N/A	All	Creation

Diffusion / *Distribution*				
Société / *Company*	**Département / *Department***	**Nom / *Name***	**Papier / *Paper***	**Elec. / *Digital***
Diffusion interne / *Internal distribution*				
Aeroconseil		Archivage - Sharepoint		X
		MORELL J.Y.		X
		BRUNEAUX P.		X
		ABGRALL E.		X
		GRANGENOIS A.		X
		RIFFLART P.F.		X
Diffusion externe / *External distribution*				
AIRBUS	EYAN2	MARION G.		X
		RAKOTOARINDRIAKA B.		X

TABLE OF CONTENTS

TABLE OF FIGURES

REFERENCES

Titre / *Title*	Référence interne / *Internal reference*	Référence externe / *External Reference*
[1] ARINC424 issue 19		A424-19

ABRÉVIATIONS / ACRONYMS

ASCII	American Standard Code for Information Interchange
NDB	Navigation Data Base
UDP	User-defined Property
UML	Unified Modeling Language (http://www.uml.org/)
XML	eXtensible Markup Language (http://www.w3.org/XML/)

INTRODUCTION

During Melbourn A424 meeting in March 2009, the decision was taken to create an UML model for aeronautical data that merges A424-19 and UML model developed for NDBX. The resulting model, would then be a basis for A424A standard.
This new model was initiated by Aeroconseil that was already deeply involved in the NDBX modeling activities.
Jeppesen took the lead on the model generation in july 2009 and provided data types completed by September 2010.
The A424A ULM model was then completed by Aeroconseil and shared with the A424A WG members late December 2010..

The aim of this document is to present methodology used to design this new UML model, the assumptions used and the differences between A424-19 and this new UML model.
A final part presents open points and the possible improvements identified during the modeling.

1. ASSUMPTIONS

This A424-A model is based on ARINC 424 issue 19[1] and covers chapters 4 (records layout) and 5 (fields definitions).
Modeling is constituted by a set of class diagrams performed with Enterprise Architect 8.
This model is based on three sources:

- Jeppesen UML model. It was performed in 2009 and is based on A424-19. Only data types will be used for this new model.
- NDBX UML model performed by Airbus/AEROCONSEIL in 2008. Class diagrams pattern and some UDP will be reused. UDP are used to an automatic generation of A424 chapters 4 (UML diagrams) and 5 (data types).
- A424-19. All records names are reused, major data will be populated in UDP of UML model.

During the generation process of the model the assumptions taken during Melbourne meeting, March 2009, were taken into account.

As a reminder here are these assumptions:

- Class names in the UML model are derived from the 424 names
- XML tags derived from abbreviated names (whitespaces, slashes etc. replaced by underscore)
- Naming convention: ASCII for 132 character format, XML for XML (former NDBX)
- Full text of all fields is stored in invisible fields (UDPs). The visible attributes have a limitation of 255 characters. These are populated only for easy browsing of the model (automatically from UDPs)
- “Used on” could be populated automatically
- Content/Source split into two paragraphs
- XML units are metric (SI), see ICAO Annex 5
    - Problem: CRCs for RNP data block
    - Bearings in Radians
- Combined ASCII fields like 5.58 (rwy. mag. bearing) or 5.42 (waypoint type) will be split into distinct subchapters
- Section numbers like 5.58.1 used for new sections closely related to existing ones
- Section numbers like 5.1000 used for completely new XML content
- Blank / Reserved (spacing) / Reserved (Expansion) fields will not be modeled separately
- Keep references like 5.58, even if field descriptions are now in chapter 4

2. UML REMINDERS

This chapter explains philosophy used for UML modeling and naming convention.

2.1. UML CLASS DIAGRAMS

This part reviews UML concepts used in ARINC424A model and in this document.

2.1.1. UML Class view

This UML view describes a collection of elements (packages, classes …) and relationships between them in order to define model structure.

2.1.2. Package

A package main function is to organize UML model logically. It defines a space where names are unique. All packages and data inside this space may be reused in others packages.

2.1.3. Class and enumeration

This part defines class and enumerations.

Figure 1: class diagram

2.1.3.1. Class and object

A class is an abstract type defined by its attributes and methods in order to model objects which have these properties(attributes and methods).
To be used, a class shall be realized by an object. Its attributes will contain values.
A class may be abstract. It implies that this class shall not be realized by an object. An abstract class is used to merge common properties between different classes. See **Erreur ! Source du renvoi introuvable.**.

2.1.3.2. Enumeration

An enumeration is a specialized class which defines a set of values that it may have. It shall not be abstract and inheritance shall not be used.

2.1.4. Relationship, navigability and multiplicity

This part defines relations between classes and enumerations.

2.1.4.1. Relationship and navigability

Relationships between classes may be represented using several means. They can be specialized (inheritance, composition …), named, have a sense of applicability and inform of number of objects that this relation could accept.
The navigability defines the direction of applicability of a relation. For instance, in Figure 2: multiplicity, the Route DME class uses the NavaidChoiceRecord class.

2.1.4.2. Multiplicity

The multiplicity provides the number of object that can enter in the relation, it may be undefined (blank or 0..*), a range (1..5) or a fixed value (2). All side of a relationship could define a multiplicity. Example below illustrates it:

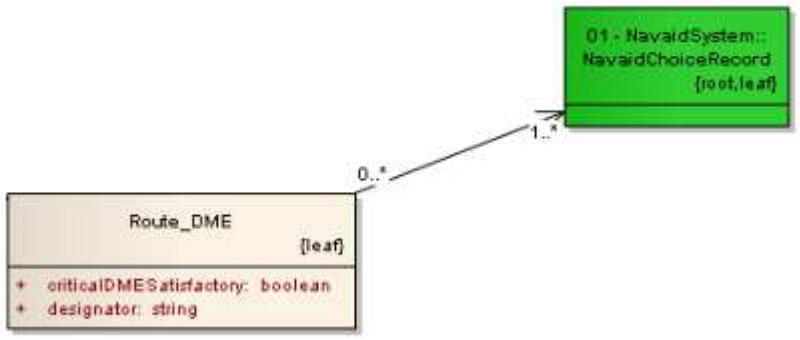

Figure 2: multiplicity

In this diagram, Route_DME references one or more NavaidChoiceRecord.

2.1.4.3. Inheritance

This type of relationship defines the connection between two classes. Child class inherits of all properties (attributes and methods) of the parent class. The diagram below illustrates this:

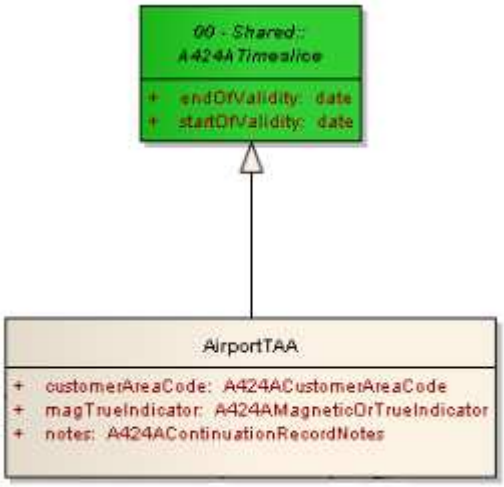

Figure 3: Inheritance diagram

AirportTAA class inherits from A424ATimeslice class. As a consequence, an AirportTAA have startOfValidity and endOfValidity attributes.

2.1.4.4. Aggregation

Aggregation relationship defines that an aggregated class is a part of another class. The two classes can exist independently.The diagram shows and example:

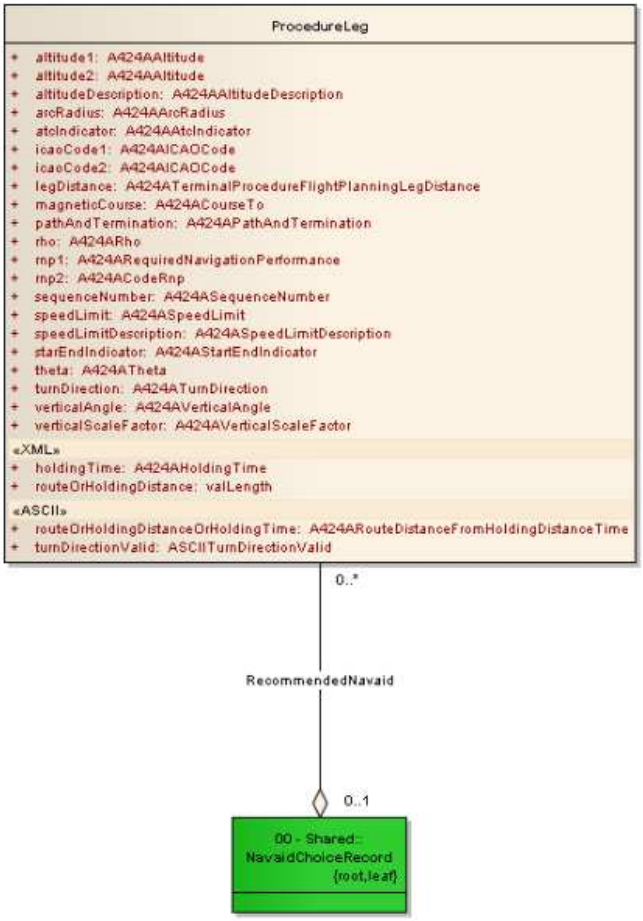

Figure 4: aggregation relationship diagram

A ProcedureLeg can contain at most one NavaidChoiceRecord (0..1).

2.1.4.5. Composition

This type of relationship is a particular type of aggregation. In a composition, when the aggregate class is removed, all composed classes are removed.

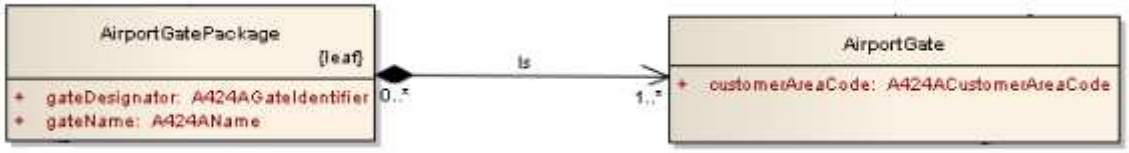

Figure 5: composition relationship

AirportGatePackage is composed by zero or more AirportGate. Moreover, this relationship is named “Is” to reinforce the fact that if an AirportGatePackage object is destroyed, all AirportGate objects will be destroyed too.

2.1.4.6. Association class

An association class is used when attributes have to be stored by the relationship itself. In A424-A UML model, association classes are used to store attributes and associated UDP of the association.

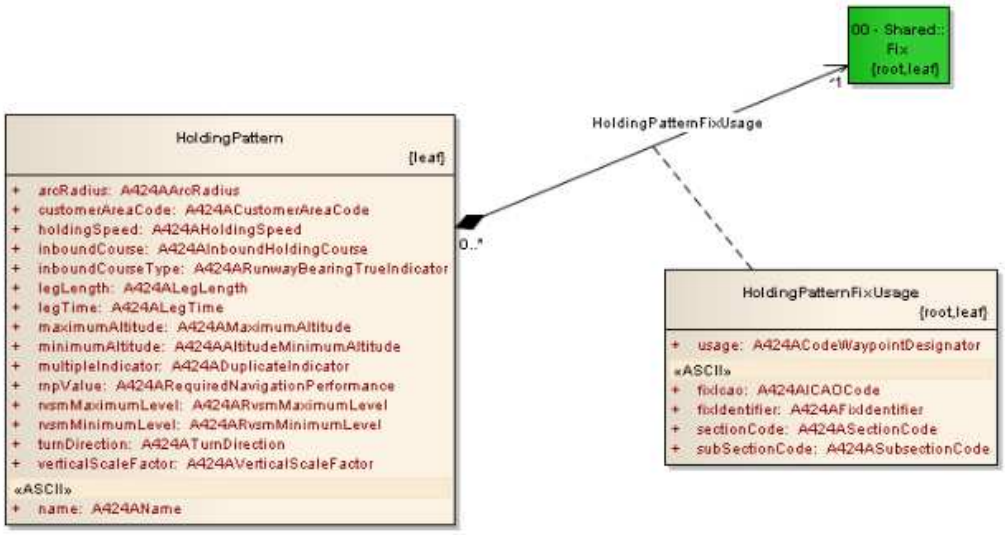

Figure 6: association class illustration

The class HoldingPatternFixUsage explains association between HoldingPattern class and Fix class. It has attribute usage that cannot be stored by Fix class: a fix can be used by another holding pattern and by holdingPattern class: this attribute is directly linked with the fix.

2.2. GENERAL PHILOSOPHY

In the frame of Melbourn meeting done in 2009, the aim of this model is to merge A424-19 data definition (chapter 4 and 5) and NDBX model design.

In 2009, Aeroconseil/Airbus have transmitted to Jeppesen an initial UML model. This model contains airport and runway records according to Melbourn meeting conclusions.
In 2010, a first release of this complete model has been provided by Jeppesen to the community.

This model is a first step and does not satisfy all requirements defined during Melbourn meeting. Indeed this first step is a simple UML representation of ARINC 424. The diagram shows the example of waypoint model:

Figure 7: Jeppesen UML Model

In this UML diagram, Waypoint primary records, Waypoint Continuation Record and Waypoint Flight Planning Continuation Records exist, as required, but NDBX design has not been introduced.
The “Figure 8: General record diagram design” shows a record designed using NDBX philosophy.

To introduce NDBX design in the model, data types from this model have been reused and records are redesigned according to NDBX model (2007).
In order to propose a model for next meeting in Febuary 2011, the new model will be a basic model (complete A424-19 data associated to NDBX design). Indeed because of the time constraint no optimization has been performed.

Each A424-19 records are modeled in UML using a set of UML classes in order to improve data quality, data upgradability, and efficiency.
NDBX modeling has been reused. For each record diagram, several classes are designed in 4 groups:

- General classes which extend Package class to model records. These classes are in yellow in the diagrams.
- Time sliced classes which extend TimeSlice class in order to model time slice record. They are in yellow color.
- A424 classes in order to model A424 mechanism such as record name, record description and notes in chapter 4. These classes are blue.
- External classes (i.e. another diagram explains this record) that are linked to this record. These classes are package classes. They are green in diagrams.

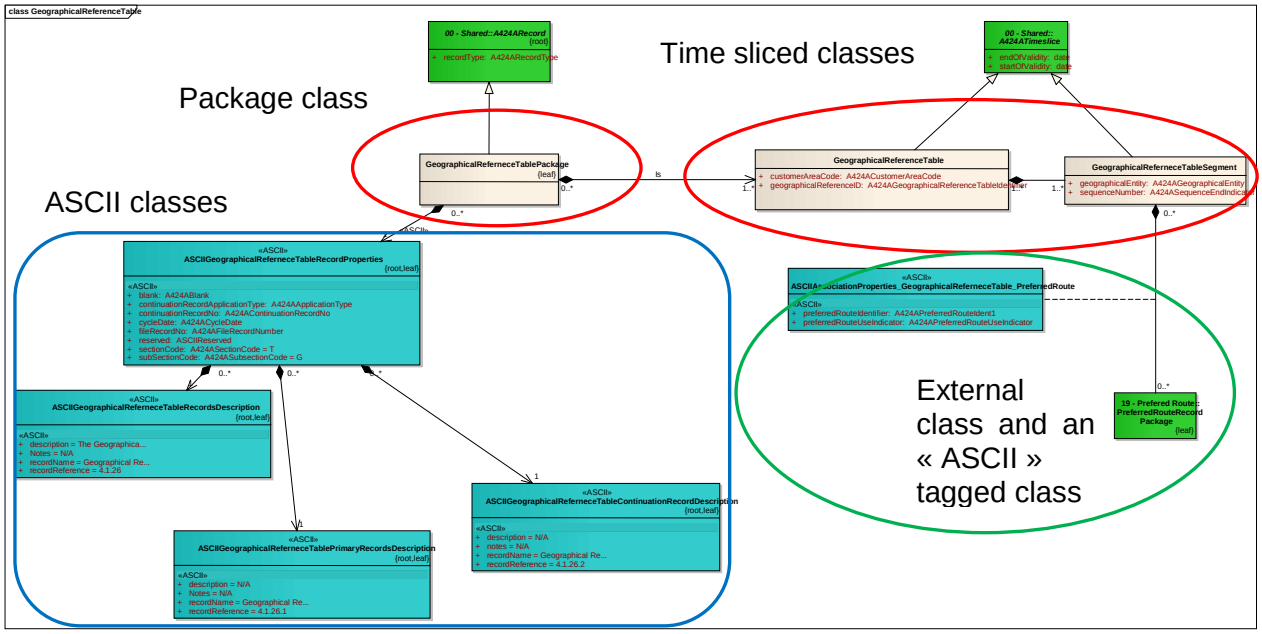

Figure 8: General record diagram design

When an association is only required for A424, an association class is used. This association class is in blue in diagrams and is tagged <ASCII> as others A424 only classes. It provides UPD values and extracts from external class only fields used by A424.

2.3. NAMING CONVENTION

Names used in this UML model result from previous NDBX and A424-19 naming convention.
A424-19 names are used as basics names and NDBX names are used for tuned names when a record required to be identified as a non A424-19 record (for instance the A424Record class).

3. A424 MODIFICATIONS FOR UML MODEL

This chapter describes all differences between UML model and A424-19. These differences have been introduced to improve 424-19 or to clarify diagrams.

3.1. CHOICE RECORDS

In NDBX model, a set of special diagrams has been introduced: Choice. The role of these diagrams is to clarify diagrams that use different records types as a choice. There are several choices:

- Fix choice that replace a waypoint, a runway, a navaid, a NDB or an airport when one of them is used as a fix.
- Procedure choice when a terminal procedure (SID, STAR or Approach) is required.
- Route choice that extends procedure choice by including Enroute Airways.
- Navaid Choice that provide choice between navaid (VHF, NDB, ILS and MLS) record and GLS record.
- Alternate Choice that purpose choice between Company route record, Airport/Heliport record and Alternate record.
-

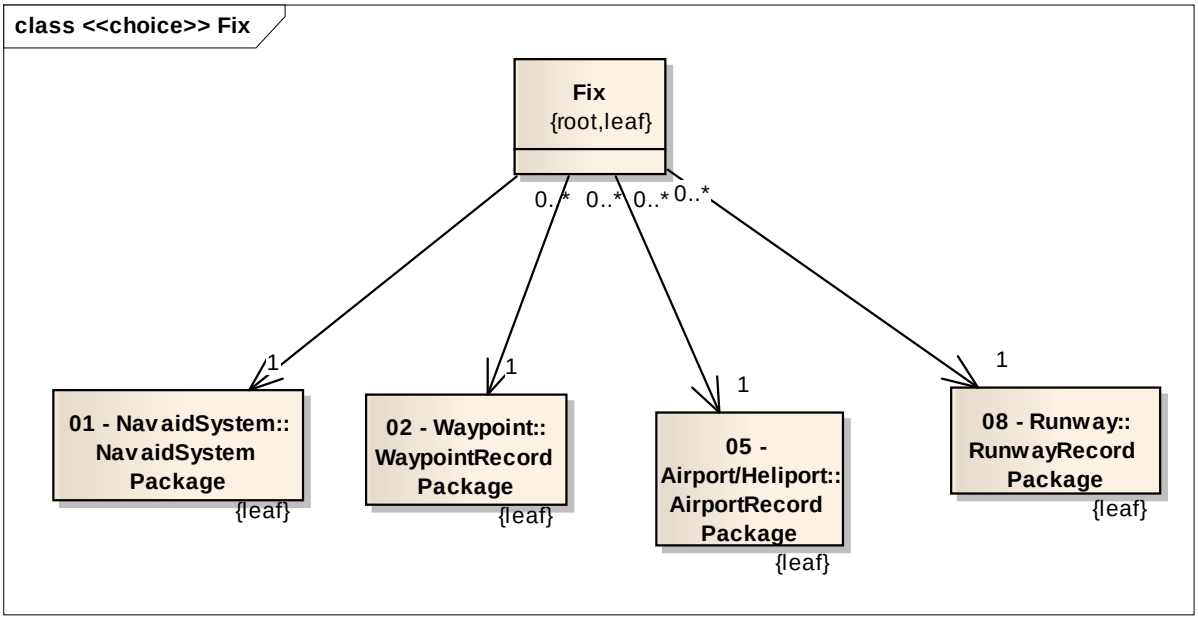

Figure 9: Choice record diagram

3.2. MINIMUM ALTITUDE ZONES

In A424, record 4.1.31 (Airport TAA), sectors bearing are represented using a field which contains two radials (5.146). In UML model, bearings are modeled in two attributes. As a consequence, all UDP associated to a sector bearings reference the same field.
Diagrams below explain this:

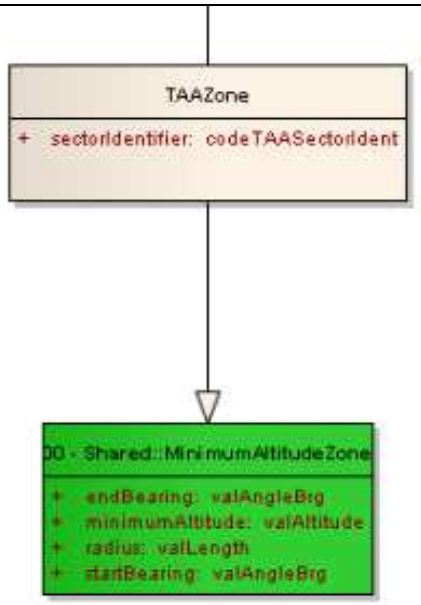

Figure 10: Minimum Altitudes in TAA

5.274 TAA Sector Radius

Definition/Description: The "Sector Radius" field in TAA records defines the start and end distances that define a TAA area. They are referenced to the TAA IAF Waypoint defined in that record. As TAA information is used towards that waypoint, the radius information is provided towards that waypoint. They enclose the sector defined in the record. The values are inclusive.

Source/Content: The Sector Radius information will be derived from official government source. Each TAA sector is made up of the start of sector radius and the end of sector radius. The values are provided in nautical miles. The first three digits define the radius for start of the sector, the second three digits the end of the sector, when flying towards the IAF Waypoint.

Used On:	Airport and Heliport TAA Primary Records
Length:	4 characters
Character Type:	Numeric
Examples:	3011, a Sector that starts at 30 nautical miles to the IAF Waypoint and ends at 11 nautical miles to the IAF Waypoint.
	0500, a Sector that starts at 5 nautical miles to the IAF Waypoints and ends at that IAF Waypoint

Figure 11: TAA Sector Radius 5.274

3.3. TERMINAL PROCEDURE ROUTE TYPE AND QUALIFIERS

In order to use choice record for terminal procedures, procedure route type, route qualifier 1 and qualifier 2 have been modified from A424.
In NDBX terminal procedure model, a SID references a SID route type, a SID qualifier 1 and qualifier 2, a STAR a STAR route type and its qualifiers and an approach references an approach route type and its qualifiers.
When terminal procedure choice record is used, an ASCII association class links this choice to the record. But which route type and qualifiers fields have to be chosen? There is no generic route type and qualifiers record.
In order to improve them, route type and qualifiers have been redesign in two steps as below:

- Each enumeration of route type and qualifiers has been encapsulated in a class (approachRouteType, SIDSTARRouteQualifier1 …).
- Abstract classes RouteType, RouteQualifier1 and RouteQualifier2 have been introduced. All other route types and qualifiers inherited from them and, for all procedures, these abstract classes are used.

The first step is necessary because UML language does not allow inheritance between an enumeration and another class. See Enumeration chapter.
This modification allows associating an approach route type to a SID procedure, but it may be checked during real NDB validation by a set of rules.
The diagram below illustrates route qualifier 1 redesign:

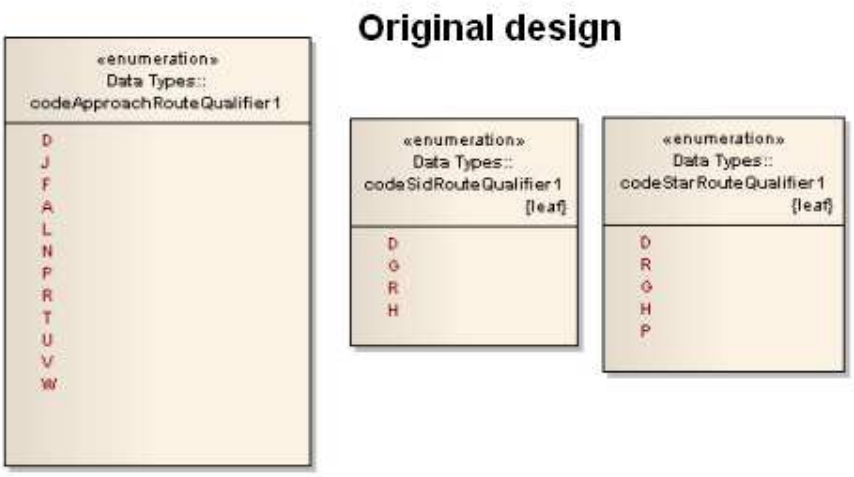

Figure 12: terminal procedure route qualifier 1 modifications, previous design

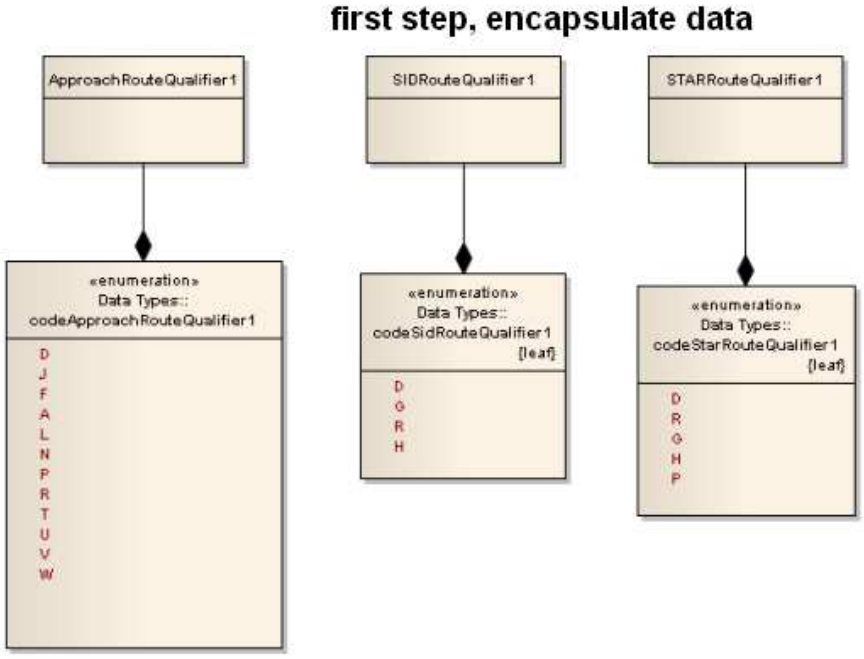

Figure 13: terminal procedure route qualifier 1 modification, first step

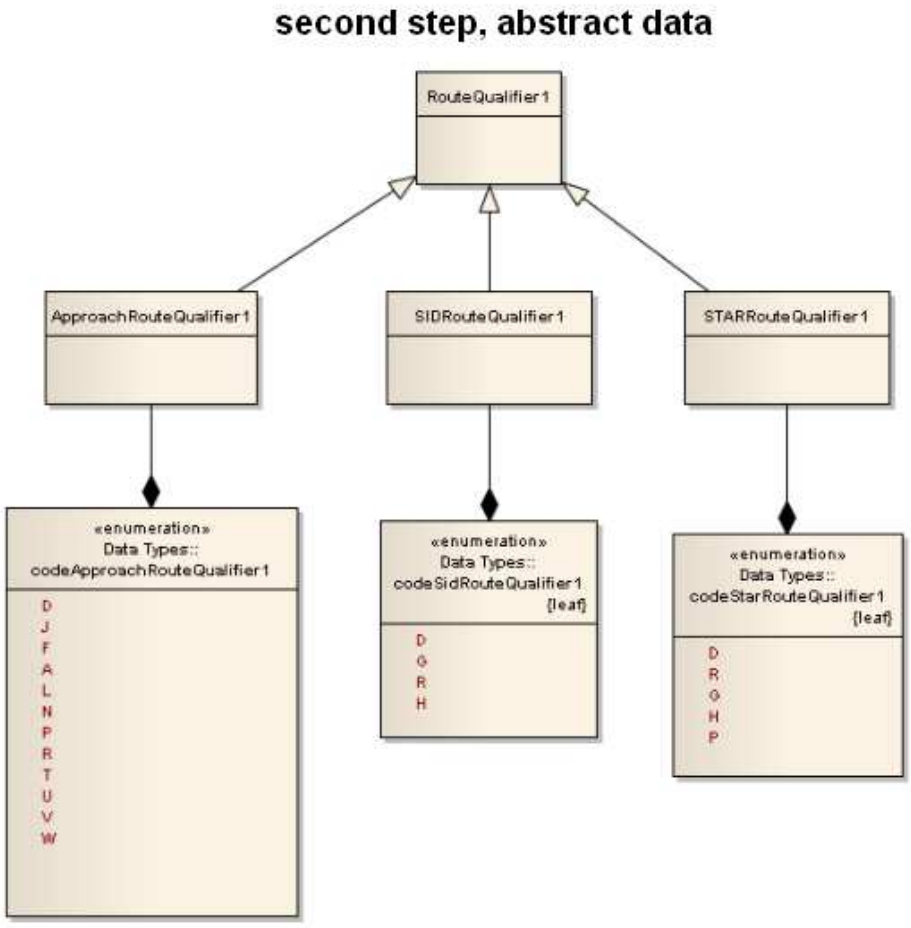

Figure 14: terminal procedure route qualifier 1 modification, second step

4. A424 MISPRINTS

During modeling, some potential editorial errors have been identified in chapter 4. They are listed below per record:

- VHF VAVAID Primary record (4.1.2.1) at comumn 122: correct reference is 5.298
- Airport SID/STAR/Approach Primary Extention Continuation Record (4.1.9.2) at column 107 to 118: why are there blanks followed by ICAO code? What is data linked to these ICAO codes?
- Airport TAA Primary record (4.1.31.1) at column 101 and 117: Sector minimum Altitude field seems to be forget, a blank, which have the same length exists at the end.
- Heliport TAA primary Record (4.2.6.1) at column 95 and 121: Sector minimum Altitude field seems to be forget, a blank, which have the same length exists at the end.
- Heliport Terminal Arrival Altitude Continuation Record (4.2.6.2) at column 30: is continuation number equal to primary record? This philosophy differs from all others records.

5. OPEN POINTS

In NDBX UML model, developed using Rose Rational®, UDP were used to indicate columns and reference of fields in records.
Enterprise Architect provides a similar system name “Tagged Value”. In the continuation of NDBX modeling, four NDBX UDP name have been reused.

Records are listed below:

UDP identifier	identified data	typical content
RationaRose$UDP:ASCII_START_COLUMN	Column start	4.1.12.1@2, 4.1.12.2@5
RationaRose$UDP:ASCII_END_COLUMN	Column end	4.1.12.1@4, 4.1.12.2@6
RationaRose$UDP:ARINC424_FORMAT_REFERENCE	Reference	5.235
RationaRose$UDP:ASCII_RECORD_REFERENCE	Associated record	4.1.12.1, 4.1.12.2

These UPD inherited from NDBX, have been selected because they are directly linked to A424-19 chapters 4 and 5. This short UDP list can be extended.

An other point is that EntrepriseArchitect (EA) limits UDP size at 128 characters. This limitation impacts record used frequently such as A424Record for field RecordType (5.2). To by-pass this limitation, EA allows to associate notes to UDP. When an UDP overpasses this limitation, all UDP content is copied in its note and only the note will be completely populated.

6. IMPROVEMENTS

Futures improvements are listed below:

- UDP of TAA sector bearings model may be improved.
- General model improvement and optimization.

Jeppesen data types do not contain all data of chapter 5 such as definition/description and source/content.
As a consequence, in future usage of this model, all fields associated to data types have to be populated.

CONCLUSION

This UML model is based on A424-19 chapters 4 and 5. It uses Jeppesen UML data types generated in 2010 and philosophy of NDBX UML model for record design and choice pattern. During design, a first optimization have been done, based on choice pattern. This pattern has been applied to some other records in order to clarify diagrams.
Futur work on this model consists of completing model data and optimizing it.

ANNEXES / APPENDICES

A. UDP MODEL IN HTML

HTML - A424A UML model - 08 12 2010.zip