

Airline Industry Data Model

Technology Physical Model Guidelines

Date: 24 March 2020

Document Version: V 0.5

*For further information and updates please contact* *Jean-Christophe Cornu (*[*cornuj@iata.org*](mailto:cornuj@iata.org)*) at IATA Airline Industry Data Model Secretariat (*[*aidm@iata.org*](mailto:aidm@iata.org)*)*

Table of Contents

[1 Introduction 4](#_Toc13828349)

[1.1 Document Purpose and Intended Audience 4](#_Toc13828350)

[1.2 Document Context 4](#_Toc13828351)

[2 Modeling XML Message Schemas 4](#_Toc13828352)

[2.1 XML Message Schema Usage 4](#_Toc13828353)

[2.2 XML Message Schema Transformation from the Logical Model 5](#_Toc13828354)

[2.3 XML Message Schema Properties 7](#_Toc13828355)

[2.4 XML Message Schema Namespaces 9](#_Toc13828356)

[2.5 XML Message Schema Quality Assurance 9](#_Toc13828357)

[2.6 XML Message Schema Diagrams 9](#_Toc13828358)

[2.7 XML Message Schema Package Structure 9](#_Toc13828359)

[2.8 Forward Engineering / Message Schema XSD File Generation 9](#_Toc13828360)

[3 Modeling Business WSDLs 10](#_Toc13828361)

[4 Aspects yet to be refined 10](#_Toc13828362)

[Annexes 11](#_Toc13828363)

[Annex A: Glossary of this document 11](#_Toc13828364)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Description of change** |
| 0.2 | 29 Feb 2016 | Michael Thomas | First version : XML Message Schema Models only |
| 0.3 | 30-Aug 2017 | Michael Thomas | Refined element names generated in section 2.2  XML Message Schema Transformation from the LDM |
| 0.4 | 12-Jul-2019 | Graham Ferguson | Modified the process for Generating an XSD |
| 0.5 | 24 Mar 2020 | Jean-Christophe Cornu | Minor correction on chap 2.3 for the removal of the choice which is now defined in T3 |

# Introduction

## Document Purpose and Intended Audience

|  |  |  |
| --- | --- | --- |
| The purpose of this document is to describe how to develop the contents of the technology pillar physical layer (partition “I4”) of the airline industry data model, i.e. XML Message Schema Models and WSDL diagrams. |  |  |

The intended audience of this document is all individuals involved in developing the physical model:

* members of PSC (Passenger Services Conference) Standards work-groups developing or expanding the model as part of BRD development,

These individuals have a variety of profiles including Airline and IT supplier Business Analysts, Enterprise Architects, Data Modelers.

This document is owned by the Methodology Group operating under the Architecture, Technology & Strategy Board.

## Document Context

The Airline industry data model is to be published by IATA as a foundational layer for the development of airline messaging standards in XML or any other data format that may emerge in the future.

The data model is structured in 3 pillars (Business, Information, Technology), 4 layers (Contextual, Conceptual, Logical, Physical), and operational stakeholder views. A separate guideline document will exist for each of the 12 partitions defined by the pillar and layer.

The data model uses UML and as a tool Sparx Enterprise Architect (EA). The first 3 layers are platform-independent. The partition addressed by the present guidelines (T4) is part of the 4th layer which is platform-specific. In the long run, it might hold models for multiple platforms, depending on business needs. Presently, we are only targeting XML schemas and Web Service Definitions. The guidelines will therefore be specific to the XML platform, as well as to the EA tool.

We are generating two types of XML schemas:

1. XML Message Schemas: are the ones addressed by this document,
2. XML Object Library Schemas: those are out of scope of the present document because they are modeled and generated in partition I4.

XML Message schema models in T4 are almost entirely generated through EA “Model Transformation” from the Logical Message Model in T3. Ideally none – and in practice very few aspects are specified and maintained at the T4 level.

Upon completion, the XML Message schema models in T4 are used to generate the actual XML Message Schema XSD files using EA “Code Engineering”.

# Modeling XML Message Schemas

## XML Message Schema Usage

XML Message Schemas are the schemas standardizing airline industry data exchanges.

A schema is typically of one of three types: Notify, Request, or Response.

The AIDM approach and tools suggest and support development of lean, strongly typed message schemas:

* The XSD files have no “include” or “import” statements. All common types – and only those needed for the specific message – are copied into the message XSD.
* There is consequently one namespace per XML Message Schema (NS + TargetNS)
* Strong typing is achieved through restrictions defined in the T3 logical message model, then transformed into the XSD models.

An XML Message Schema contains the artifact types listed in the below table.

|  |  |  |
| --- | --- | --- |
| **Meta-Class** | **Stereotype** | **Description** |
| Class | XSDtopLevelElement | Root element of the schema |
| Class | XSDcomplexType | Grouping of multiple elements |
| Class | XSDchoice | Grouping of alternative elements |
| Class attribute | XSDelement, XSDattribute | Part of a complex type |
| Class | XSDsimpleType |  |
| Enumeration | XSDenumeration | Defines a list of acceptable values. |
| Class attribute | CodelistEntry | Part of an Enumeration; one of the acceptable values |

## XML Message Schema Transformation from the Logical Model

This section is common to XML Message Schema Transformation (T4) and XML Library Schema Transformation (I4), from their respective logical model sources, because the majority of the guidelines are common.

By default, the guidelines will apply to both.

Paragraphs applying to XML Message Schemas only will be marked with “(MS only) “.

Paragraphs applying to XML Library Schemas only will be marked with “(LS only) “.

XML Schema models are automatically produced by transforming models from the logical level. Those schema models will subsequently have to be generated into the actual XSD files.

(MS only) An XML Message Schema model is to be produced for each Logical Message Model in T3.

(LS only) An XML Library Schema model is to be produced for each Information Domain package, and also for the “Common Library” package (holding all BDTs, ENUMs, PRIMs) of the integrated Logical Data Model in I3.

***Running the Transformation tool***

The AIDM uses a customized version of the EA Transformation tool. Transformation is a package-level operation, i.e. it transforms the elements in a package (ignoring any diagrams).

The target XML schema model is produced under T4 or I4, depending on whether the source is under T3 or I3, and with a package hierarchy analogous to the one of the sources in T3/I3.

*For example:* I3 Information Logical Models/Governance View/Logical Info Blueprint Specifics/Offerings

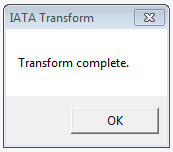
*is transformed into:* I4 Information Logical Models/Governance View/Logical Info Blueprint Specifics/Offerings

If the target model already exists, it will be updated. If the target hierarchy of packages does not yet exist, it will be automatically created by the transformation tool, entirely or partly as needed.

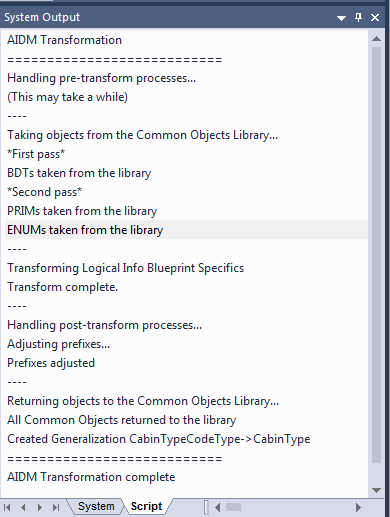
With regards to Common Library objects:

* the artifacts to be transformed are located in 3 distinct sub-packages (BDTS, ENUMs, PRIMs), 2 / respectively 1 level below the message schema package / respectively Common Library package which for technical reasons also needs to be transformed in one operation.
* The pattern of package hierarchies is :
  + I3 Information Logical Models / Governance View / Common Library Master / BDT Common
  + I3 Information Logical Models / Governance View / Common Library Master / BDT Common
  + “ / ENUM Common
  + “ / PRIM Common
  + T3 Technology Logical Models / Operational View / …. / Msg Schema Name / BDT Common
  + “ / etc.
* For all the artifacts to be transformed, the AIDM transformation tool temporarily moves the artifacts 2 / respectively 1 level up, transforms the package, then moves the artifacts back down into their specific packages. In case of “Common Library” transformation, all the artifacts will be moved. In case of Message model transformation only those BDTs/ENUMs/PRIMs needed by the ABIEs in the message are moved, so the message package temporarily contains the exact collection of artifacts need to produce the lean schema.

To transform from the logical model and create the XSD model:

1. Make sure you have locked the target package with all its contents (target XML model and diagram) if they already exist, or else the target parent package under which transformation should create the target package/hierarchy.
2. Make sure you have locked the source package with all its contents, in case of Message or Common Library package (not needed for Info Domain packages).
3. Select the source package (a Message, an Information Domain, or the Common Library) in the project browser. Right-click and select “Scripts” / “Transform to XSD”.
4. Wait until you get the completion message on your screen (shown to the right). For a Message model or an Info Domain, this can take 5 to 60 min depending on its size and on your workstation performance.   
   After having pressed “OK”, the XSD diagram created will automatically open.

Hints:

* The transformation will display progress information (as will other AIDM tools) in the “System Output” window. Hence it may be appropriate to have that window be part  
   of your usual EA desktop configuration. It can be opened from top menu “View”.

A “System Output” window with sample contents is shown to the right.

* The “System Output” window will indicate current stages such as “pre-transform”, “transform”, “post-transform”. During “transform”, an additional Transformation Progress” window will open automatically and display further details.
* For Message Schemas and for Common Library package transformation – and in the un-expected event that the tool encounters a severe error and stops – you need to check if any BDTs, ENUMs, PRIMs are located directly in the Message or Common Library package, and  
  if so move them back down into their respective sub-packages.
* In case of a Cloud connection error, the tool will bring up a small dialog with a message referring to the “cloud”. As most of these issues are intermittent, you can then typically press “OK”, and the tool will proceed and complete successfully.
* Any error message other than temporary cloud connection issues should be reported to the AIDM Administrator.

***Understanding what elements are created by the Transformation tool***

This paragraph is purely for information, in case the modeler wants to understand what objects are created.

The following table shows how the artifacts from the logical models are transformed.

|  |  |
| --- | --- |
| **Source Element Stereotype in T3/I3** | **Target Element Stereotype in T4/I4** |
| IATA\_ABIE | XSDcomplexType |
| IATA\_BBIE (within IATA\_ABIE) | IATA\_XSDelement or IATA\_XSDattribute (depending on tagged value IATA transform to XSD Attribute) |
| IATA\_ASBIE | (MS only) Directional Association |
| Generalization | (MS only) Generalization |
| IATA\_BDT | XSDelement or XSDattribute (depending on the complexity of the BDT, according to UN/CEFACT XML NDR) |
| IATA\_ENUM | XSDenumeration |
| IATA\_CodelistEntry (within IATA\_ENUM) | CodelistEntry |
| IATA\_PRIM | *(not transformed, however references to the PRIMs are translated into references to standard XSD data types)* |

Notes:

* (LS only) Contrary to message schemas, Information Domain Library Schemas will:
  + ignore all associations that exist in the source logical model; neither associations nor elements for cross-complex-type references are generated
  + not include the BDTs and ENUMs. Instead, they will rather have an “xs:include” statement referencing a schema file with all BDTs and ENUMs in the integrated data model. Indeed, generating “lean schemas” would have been less important for library object schemas, than it is for message schemas.
* (MS only) The translation of associations into XSDelements is done at code generation time. Associations and Generalizations are therefore carried over from Logical to XML model as such.
* Names of ABIEs, BBIEs, ASBIE Roles, BDTs, BDT SUPs, ENUMs are translated:
  + into Upper Camel case,
  + with any space, underscore, and hyphen characters being removed,
  + with abbreviations substituted to words as per list of abbreviations in guideline document “AIDM Guidelines - Standard Abbreviations and Acronyms.xlsx”,
  + with underscore added after any acronyms followed by any further words.

## XML Message Schema Properties

The vast majority of the properties of the artifacts in the XML Message Schema models must not be modified at this (T4) level. This applies to standard EA properties as well as to tagged values added for the AIDM such as references back to sources. All these attributes are maintained at the upper layers and derived from the logical model.

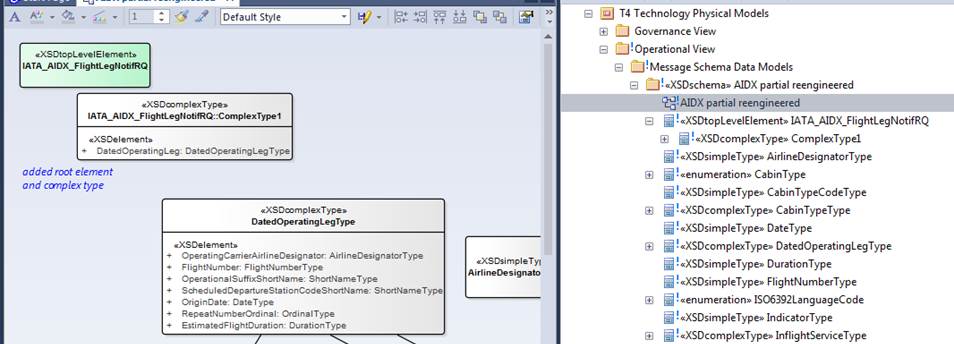
The only aspects to be modeled at this level are:

* Add an XSD Top Level Element,
* Tune the positions of particles within XSD complex types.

The XSD Top Level Element can be added to the XML message schema transformed from the logical level performing the following 3 steps:

1. From XML diagram toolbox: Create “Element”: drag from toolbox, fill in name (give the same name as the name of the schema), clear “type”, select nested complex type. This will create graphically superposed top-level element and complex type (“complexType1”).  Move them side by side in the diagram in order to see both.  Make sure in the Project Browser that the complex type is still embedded within/underneath the top-level element.
2. Create xsdElement inside complexType1 with Type pointing to the first ABIE’s complex type to include in the schema (and name typically equal to the name of the ABIE i.e. name of the complex type without “type” at the end).
3. Move Top Level Element to first place in project browser. Top Level Element will be the only xsdElement – amongst all the complex and simple types – at the first level inside the XSD.

Here below is an example with top level element added (the two entities in the upper left corner):



The position of particles within XSD complex types – e.g. the sequence of all XSD elements, XSD attributes, and associations to child complex types within the XSD complex types – can be specified through setting the “Position” tagged value to consecutive numbers starting with 1, without missing any number:

* within the XSD elements and XSD attributes (select Properties / UML / Tagged Values,
* within directional associations (select Properties / Tags / tags for “connector source”).

By default, the AIDM transformation tool will have filled in the Position tagged value for XSD elements and XSD attributes based on their order in the logical model in the project browser. However, the Position tagged value of associations will be empty and would be added in random order after the elements and attributes. If associations should be included in a specific order, then their Position tagged value needs to be filled in accordingly. If associations should be included in between elements or attributes, then element/attribute Position tagged values need to be renumbered to make space, and association Position tagged values must be set to values in between.

Note: when repeating “Model Transformation” after some changes have been made to the source (logical) model, EA will “synchronize” i.e. merge changes made in the source and changes already made in the target models.

## XML Message Schema Namespaces

Each XML Message Schema will have its specific namespace (NS + TargetNS).

XMLNS and Target Namespace are properties of the package holding the schema model, and will be defined there.

## XML Message Schema Quality Assurance

* All parts of the model created through model transformation other than the items listed as being maintained in section 2.3 must remain unchanged at this level.

## XML Message Schema Diagrams

Model Transformation will automatically create a diagram with all artifacts.

While this diagram is not needed for generating the XSD file, it will be natural to add the top-level elements through, and to it.

## XML Message Schema Package Structure

The package structure under T4 / Operational View will reflect the exact same hierarchy as under T3 / Operational View, and is automatically created by the transformation tool if not created manually.

The diagram and all XSD elements and types will be located in one single package, under a package hierarchy pattern of: T4 / Operational View / Value Chain Primary Activity Name / Message Name.

## Forward Engineering / Message Schema XSD File Generation

To generate the XSD file:

1. select the XSDschema package in the Project Browser and run the script Generate XSD.
2. A file browser window will appear; select the file you want to generate if one already exists, or if this is the first time select the desired directory and create a new file by entering the desired name including the suffix “XSD”. If you hit cancel at this point the script will be aborted and no changes will be made.

|  |  |
| --- | --- |
|  | *Note that if the file already exists a copy of the exiting file will be made and given a suffix of “.bak”. if a file with the same name exists an additional “.bak” suffix will be added until a unique name is achieved. This enables the modeller to compare previous files to verify any changes, or to return to a previous XSD file should there be a system failure.* |

1. The script will generate the XSD and sort the contents alphabetically it finds a tagged value named “sortXSDFile” set to “Y” or “y” against the XSDschema package. If no tagged value exists or it does not have a value of “Y” or “y” the schema in the file will remain as ordered by EA.

# Modeling Business WSDLs

\*\*\* section yet to be completed \*\*\*

# Aspects yet to be refined

1. **Naming Conventions for XML Schema Namespaces and Target Namespaces**
2. **Abbreviations** to be automatically applied by the transformation tool when going from PIM model to XSD tags (not yet decided / implemented in the tool).

# Annexes

## Annex A: Glossary of this document

|  |  |
| --- | --- |
| Term | Description |
| Acronym | An initialism formed from the initial letters of other words and often pronounced as a word (e.g. ICAO, ISO). |
| ABIE | Aggregated Business Information Entity (LDM artifact, see I3 guidelines) |
| AIDM | Airline Industry Data Model |
| ASBIE | Association Business Information Entity (LDM artifact, see I3 guidelines) |
| BBIE | Basic Business Information Entities (LDM artifact, see I3 guidelines) |
| BDT | Business Data Type (LDM artifact, see I3 guidelines) |
| BIE | Business Information Entity : includes all of ABIE, BBIE, ASBIE |
| LDM | Logical Data Model |
| PRIM | Primitive Data Type (logical level artifact, see I3 guidelines) |
| Tagged Value | “Custom” properties added to the standard set of information stored with each artifact in EA, either through adding a certain “profile” (e.g. UPCC) or defined and added by the data model team. |