

Open Geospatial Consortium

Date: 2010-12-19

Reference number of this document: OGC 11-106r1

<http://www.opengis.net/doc/ows8-notam-refactor>

Category: Public Engineering Report

Editors: Rob Atkinson, James Groffen

OGC[®] OWS-8 Digital NOTAM Refactor

Copyright © 2011 Open Geospatial Consortium.

To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>.

Warning

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

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

Preface

Use Case modeling has been undertaken to establish the scope and business requirements for system and domain information modeling. INSPIRE has elucidated on this process (INSPIRE, 2007) and similar approaches have been used for domain models such as GeoSCIML (Sen and Duffy, 2005).

This document assumes formal domain modeling undertaken according to the ISO 19103 Conceptual Schema Language (ISO, 1999) and 19109 Rules for Application Schema (ISO, 2000).

A number of additional perspectives may be added from the experience of these and other domain modeling exercises.

This document is a deliverable for the OGC Web Services 8 (OWS-8) testbed activity. OWS testbeds are part of OGC's Interoperability Program, a global, hands-on and collaborative prototyping program designed to rapidly develop, test and deliver proven candidate standards or revisions to existing standards into OGC's Standards Program, where they are formalized for public release. In OGC's Interoperability Initiatives, international teams of technology providers work together to solve specific geoprocessing interoperability problems posed by the Initiative's sponsoring organizations. OGC Interoperability Initiatives include test beds, pilot projects, interoperability experiments and interoperability support services - all designed to encourage rapid development, testing, validation and adoption of OGC standards.

The OWS-8 sponsors are organizations seeking open standards for their interoperability requirements. After analyzing their requirements, the OGC Interoperability Team recommend to the sponsors that the content of the OWS-8 initiative be organized around the following threads:

- * Observation Fusion
- * Geosynchronization (Gsync)
- * Cross-Community Interoperability (CCI)
- * Aviation

More information about the OWS-8 testbed can be found at:

<http://www.opengeospatial.org/standards/requests/74>

OGC Document [11-139] “OWS-8 Summary Report” provides a summary of the OWS-8 testbed and is available for download:

https://portal.opengeospatial.org/files/?artifact_id=46176

License Agreement

Permission is hereby granted by the Open Geospatial Consortium, Inc. ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD.

THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications.

This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

Contents		Page
1	Introduction	1
1.1	Scope	1
1.2	Document contributor contact points	2
1.3	Revision history	2
1.4	Future Work	3
1.5	Forward	3
2	References	3
3	Conventions	4
3.1	Abbreviated terms	4
3.2	UML notation	4
4	Benefits of Formalising the Model	5
4.1	Reduce Ambiguity	5
4.2	Dependency Management	5
4.3	Governance	7
5	Refactor Activities	9
5.1	Starting Point	9
5.2	Repackaging	9
5.3	Linking to a Real Type	9
5.4	Generated Tagged Values	10
5.5	Namespace in Class Names	11
5.6	Referencing the Harmonized Model	11
5.7	Generate Package Dependencies Diagram	11
5.8	Adding Sequence Number Tagged Values	11
5.9	Documenting the Model	11
5.10	Moving Code Lists to it's own Package	12
5.11	Publish to the Model Registry	12
6	Nillable Attributes	14
6.1	Nillable Tagged Value	15
6.2	Voidable Stereotype	15
	Recommendation	15
7	Temporality	16
7.1	Modelling Approach	17
	Pattern 1: Special Feature Stereotype	18
7.2	Pattern 2: Tagged as Time-variant	19
7.3	Pattern 3: Timeslice Feature with Time-variant Properties	21
7.4	Temporality in ISO Standards	22
7.5	Summary	22
7.6	Work Performed	23
7.7	Usability of Time-variant Features	23

Figures	Page
Figure 1 – Example conformance result report.....	5
Figure 2 – Package dependency diagram – dependencies of application schemas.....	6
Figure 3 – Dependencies published in the Registry Browser for the Event Model.....	7
Figure 4 – Registry Browser with the Event Application Schema selected	8
Figure 5 – Events XML Schema in UML	10
Figure 6 – Refactored Event Model.....	13
Figure 7 – ISO 19136 - DynamicFeature Class Diagram.....	16
Figure 8 – Special Feature Stereotype - DynamicFeatureType	18
Figure 9 – Properties individually labeled as time-variant	19
Figure 10 – Timeslice feature with explicit properties.....	22

OGC® OWS-8 Digital NOTAM Refactor

1 Introduction

This engineering report details the benefits and application of using a domain modelling process refined to formalise the use of UML to describe domain models.

UML is a common tool for modelling. The flexibility of UML allows for multiple approaches to be taken when modelling, introducing diversity into the application of UML to domain models. The approach described in this document and the associated Domain Modelling Cookbook (OGC 11-107) is designed to meet the needs of defining ISO / OGC conformant domain models using UML in a way that does not introduce idiosyncrasy into the model and how it must be interpreted.

As there are multiple ways to interpret the UML, each model would also need to include instructions on how to interpret modelling details. While this is the goal of the standardisation of domain modelling, CSIRO have progressed on the standards by providing tools and templates that help eliminate unwarranted variation in modelling approach. These tools also provide benefits such as automating repetitive tasks, supporting conformance checking and generally encouraging a modelling approach that has been honed over many years of experience.

This document also makes recommendations on changes to the modelling approach of AIXM and documents the application of these recommendations to a subset of AIXM - Digital NOTAM Events. In some cases these recommendations have broader ramifications than AIXM alone. In these situations an approach is suggested on how to implement these recommendations.

1.1 Scope

AIXM is a GML Application Schema described in UML using the relevant ISO / OGC standards from the 19100 series. The Digital NOTAM Events Specification (DNES) is an extension of AIXM that can describe notices to airmen using the AIXM standard.

This document has been produced in conjunction with the Domain Modelling Cookbook - a practical guide to domain modelling following a series of best practices developed by the CSIRO and other OGC members.

Where possible, documentation of the refactor effort for Digital NOTAM to be compatible with these practices is incorporated into the domain modelling cookbook.

This document contains details not appropriate for the wider audience of the cookbook, as well as outcomes and recommendations specific to Digital NOTAM, and AIXM in general.

1.2 Document contributor contact points

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

Name	Organization
James Groffen (editor)	LISAsoft
Rob Atkinson (editor)	CSIRO
Johannes Echterhoff	iGSI

1.3 Revision history

Date	Release	Editor	Primary clauses modified	Description
8 AUG 2011	0.1	Rob Atkinson	throughout	Document framework
9 SEP 2011	0.2	Jim Groffen, Rob Atkinson, Johannes Echterhoff	Temporality	Temporality discussion added
12 SEP 2011	0.3	Jim Groffen	Refactor Activities, Nullability	Refactor activities initial draft, nullability discussion added
18 SEP 2011	0.4	Jim Groffen	Benefits, Refactor Activities	Domain Model discussion and AIXM / Digital NOTAM refactor activities added
28 AUG 2011	1.0.0	Jim Groffen	throughout	Review and reformat into OGC ER template.
30 AUG 2011	1.0.1	Jim Groffen	Future work, references, conventions	Minor corrections based on feedback.

1.4 Future Work

This document could be improved by additional coverage of AIXM – only the Event extension, messaging and the DynamicRunway package has been considered in this document.

If any recommendations from this document are implemented, it would be appropriate to document the outcomes here.

1.5 Forward

This document is a deliverable of the OGC Web Services (OWS) Initiative - Phase 8 (OWS-8). It describes a recommended approach to domain modelling following ISO and OGC standards. The approach described supports GML application profiles defined using formal UML. Also described are tools to support the modelling process from conceptual design (UML) to physical implementation (XML Schema).

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

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

2 References

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

- Digital NOTAM Event Specification v1.0
- AIXM Temporality Concept
<http://www.aixm.aero/gallery/content/public/AIXM51/AIXM%20Temporality%201.0.pdf>
- INSPIRE, E., 2007. D2.6: Methodology for the development of data specifications. INSPIRE Drafting Team "Data Specifications"
- ISO, 1999. ISO 19103 Geographic information - Part 3: Conceptual schema language. International Organization for Standardization (ISO).

- ISO, 2000. ISO 19109.3 Geographic information - Rules for application schema. International Organization for Standardization (ISO).
- ISO, 2007. ISO 19136 Geographic information - Geographic Markup Language (GML). International Organization for Standardization (ISO).
- Sen, M., Duffy, T., 2005. GeoSciML: Development of a generic GeoScience Markup Language. *Comput. Geosci.*, 31(9): 1095-1103.
- OWS-8 Aviation Architecture ER (OGC 11-093)
- Domain Modelling Cookbook (OGC 11-107)
- SEEGRID Community Website - Nillable Voidable Discussion - <https://www.seegrid.csiro.au/wiki/CGIModel/NillableVoidableDiscussion>

3 Conventions

3.1 Abbreviated terms

AIXM	Aeronautical Information Exchange Model - A GML Application Profile that defines feature types and standardised encoding methods for aeronautical information.
DNES	Digital NOTAM Event Specification – This document covers the Event extension of AIXM, NOTAM and how the event extension is to support digitization of NOTAM.
GML	Geographic Markup Language - as a modeling language for geographic systems as well as an open interchange format for geographic transactions.
NOTAM	Notice to Airmen - A NOTAM is filed with an aviation authority to alert aircraft pilots of any hazards <i>en route</i> or at a specific location. The authority in turn provides a means of disseminating relevant NOTAMs to pilots.
UML	Unified Modelling Language - a standardized general-purpose modeling language that includes a set of graphics for visual representation.

3.2 UML notation

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

This document introduces the concept of formal UML notation that requires conformance of the UML model to an ISO / OGC standardized approach. Conformance can be checked with a provided tool – Solid Ground.

4 Benefits of Formalising the Model

As with any standardisation process, the effort of defining and conforming to the standard needs justification. While the Domain Modelling Cookbook (OGC 11-107) makes a case for domain modelling in general, a justification for expending the effort of formalising the model to use the common process discussed has to be made.

4.1 Reduce Ambiguity

The ideal would be completely unambiguous, self describing models. There would be no need for ancillary documentation beyond the model itself. This process incorporates multiple strategies for reducing ambiguity from the model.

- Automated conformance checking that provides context of the conformance problem and will if possible select the related element in the project browser.
- The conformance checking also generates warnings for model elements that have no notes defined.

	Type	Description
1	✘	The attribute: AIS_Publication.type' has an unknown data type: 'CodeAISPublicationType'. This may be cau...
2	✘	The attribute: AIS_Publication.series' has an unknown data type: 'aixm:CodeUpperAlphaType'. This may be ...
3	✘	The attribute: AIS_Publication.number' has an unknown data type: 'aixm:NoNumberType'. This may be caus...
4	✘	The attribute: AIS_Publication.year' has an unknown data type: 'aixm:DateYearType'. This may be caused b...
5	✘	The attribute: AIS_Publication.issued' has an unknown data type: 'aixm:DateTimeType'. This may be cause...
6	✘	The attribute: AIS_Publication.effectiveStart' has an unknown data type: 'aixm:DateTimeType'. This may be ...
7	✘	The attribute: AIS_Publication.effectiveEnd' has an unknown data type: 'aixm:DateTimeType'. This may be ...
8	✘	The attribute: AIS_Publication.content' has an unknown data type: 'aixm:XHTMLType'. This may be caused...
9	✘	The attribute: AIS_Publication.publisher' has an unknown data type: 'aixm:UnitPropertyType'. This may be c...
10	✘	Tagged value inlineOrByReference={inline, byReference} owned by attribute 'type' is not a correct type
11	✘	Tagged value isMetadata=BOOL owned by attribute 'type' is not a boolean type

Figure 1 – Example conformance result report

4.2 Dependency Management

Discrete packaging of classes that have well defined package dependencies is encouraged. These package dependencies are automatically modelled by the SolidGround tool, which will generate a UML Package diagram that includes dependency information that can be interpreted as normative.

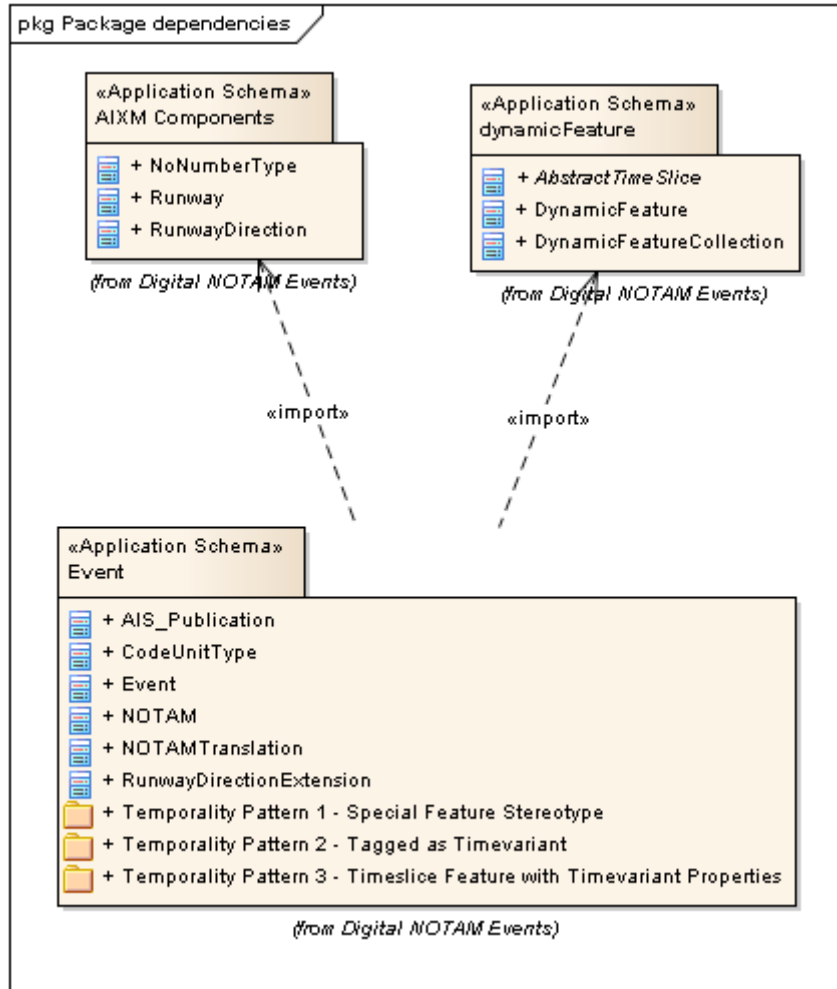


Figure 2 – Package dependency diagram – dependencies of application schemas

By enforcing no circular references between packages they can be safely, individually consumed by external parties. This is necessary for compliance with the ISO conceptual schema language. Tools such as the Model Registry Browser can use formally defined dependency information to publish the dependency information.

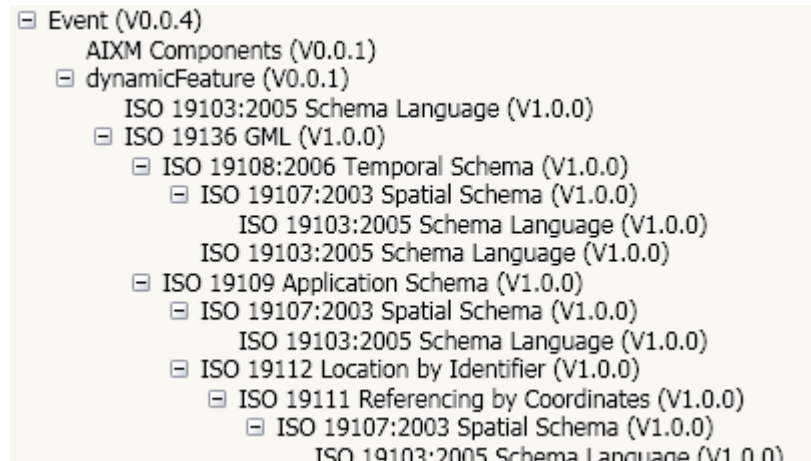


Figure 3 – Dependencies published in the Registry Browser for the Event Model

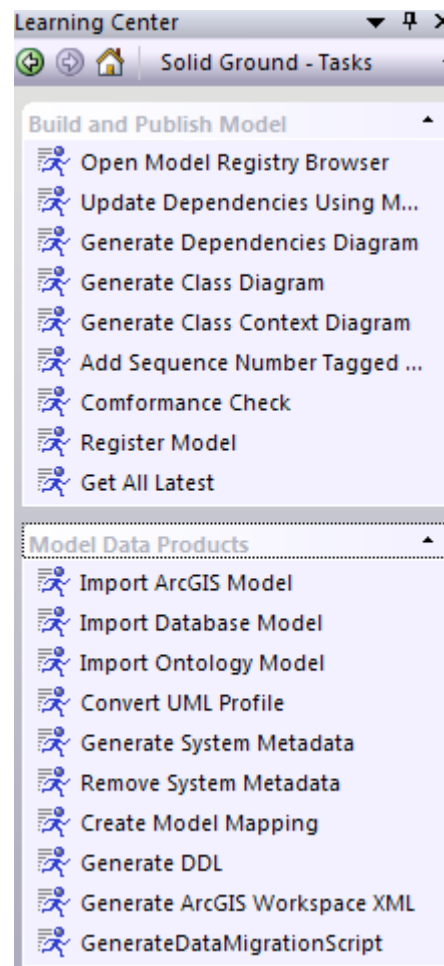
This dependency information can be taken advantage of when importing models, in this case all dependent models can be imported from the registry.

4.3 Governance

By formalising the model it can participate in a broader community of domain models. A register of domain models can be established for a community of practice, allowing discovery, import and publishing extended models.

The image to the right is of the tasks available in the Solid Ground plugin for Enterprise Architect. This plugin interacts directly with the model registry in a few ways;

- Open Model Registry Browser: authenticates to and opens the model registry browser window.
- Update Dependencies Using Model Registry: All packages the selected package depends on will be updated from



the model registry, picking up any changes that have been submitted to the registry.

- Register Model: initial registration or update of the selected model to the model registry.

The registry browser supports version tracking features. When a model is registered the version number recorded in the Application Schema package of the model being registered is used to identify a model version. This is used to notify custodians of models that are dependant of your of the change, including the nature of the change by following a common Major.Minor.Revision approach.

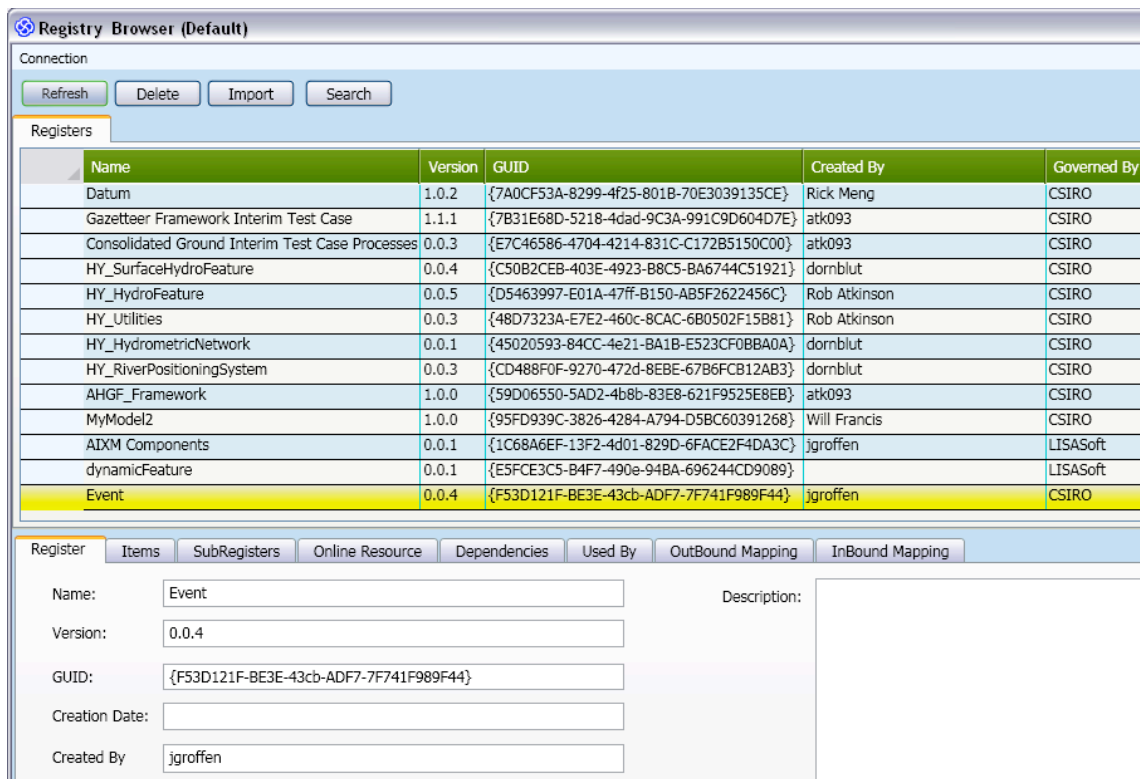


Figure 4 – Registry Browser with the Event Application Schema selected

5 Refactor Activities

5.1 Starting Point

The AIXM UML models were imported into Enterprise Architect from XMI files. A small subset centered on the Runway and RunwayDirection classes were cloned into a separate package (AIXM Components) for this refactor effort. This was done to ensure the current AIXM UML was considered in the refactor effort.

For the purposes of the Domain Modelling Cookbook, the refactor effort for Digital NOTAM Events started from the Event XSD files instead of the UML model. This allows the modelling cookbook to show an example of starting conceptual modelling by taking advantage of an existing source of concepts derived from an existing system, while the RunwayDirection subset from AIXM ensures the UML methodology applied for AIXM is still considered.

A variety of tasks were performed to take the UML generated by the Enterprise Architect XSD Import tool to a UML model that meets the ISO 19103 conformance target. These are detailed below.

5.2 Repackaging

A new Application Schema stereotyped package was created to house the refactored Event schema. This isolates our work from the XSD import and the existing AIXM UML. Dynamic features were also cloned from ISO 19136 as part of the temporality modelling discussed later in this document.

5.3 Linking to a Real Type

Many of the types referenced are not actual links to defined classes but simple textual references. When performing a conformance check with Solid Ground these are reported as an unknown data type.

There are two parts to this - mapping to a standard type name and then linking actual type model so that the model is internally consistent and can be processed.

Solid Ground can perform a mapping of types using a mapping file. A few mapping files are provided with Solid Ground, or you can create your own. An example is the Oracle to ISO mapping file that will switch from Oracle types to the appropriate ISO Harmonized Model type.

5.4 Generated Tagged Values

The majority of tagged values in both the AIXM model and especially in the imported XSD model was not expected and generated warnings or errors in conformance checking.

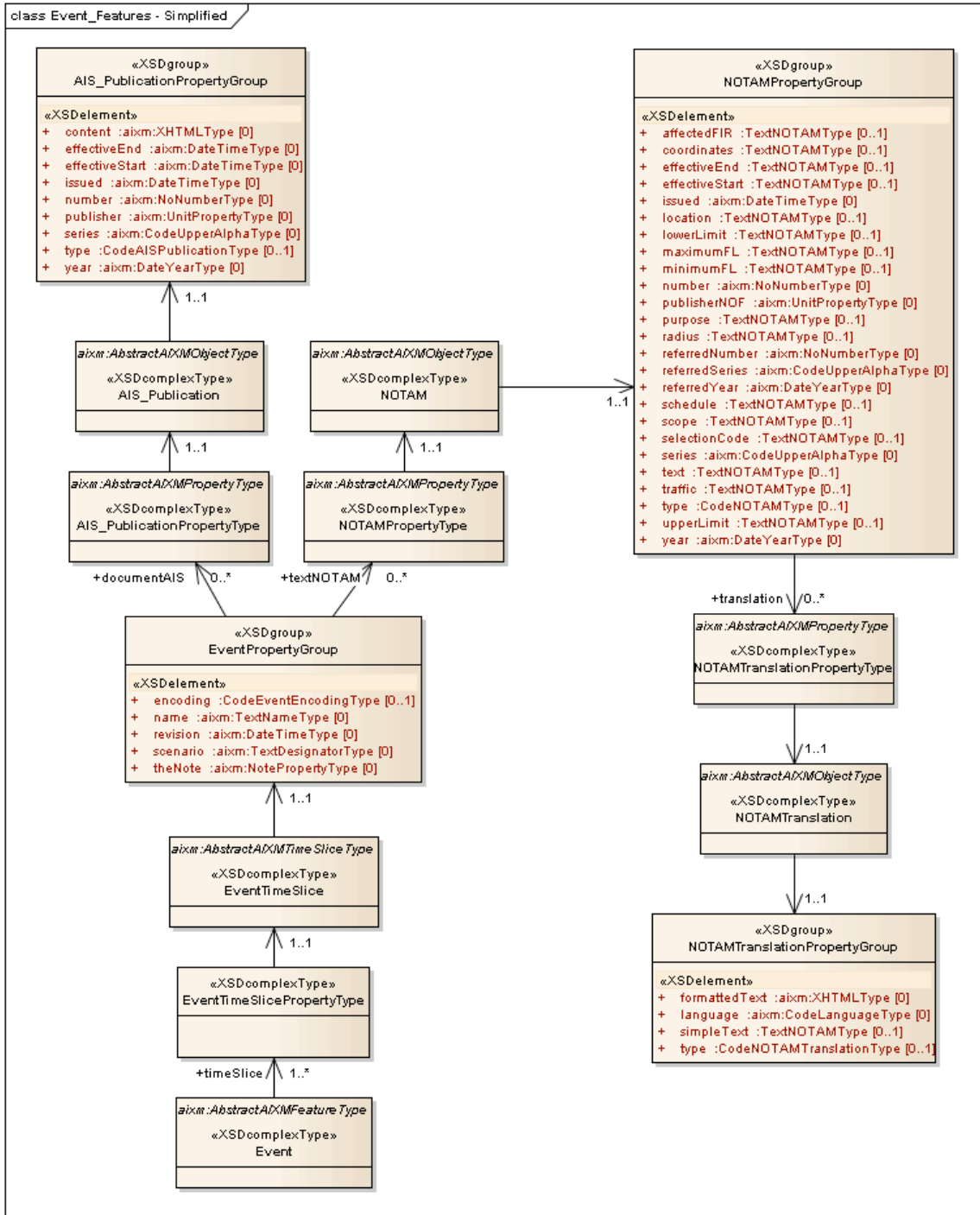


Figure 5 – Events XML Schema in UML

5.5 Namespace in Class Names

Imported XSD Elements embed the namespace into the type information. This is generally automatically resolved when ensuring all types are correctly linked.

5.6 Referencing the Harmonized Model

Ensure that the correct type references the ISO Harmonized Model classes and not a local cloned copy or unconnected text.

5.7 Generate Package Dependencies Diagram

The package dependency diagram is automatically generated by the Solid Ground tools to document the dependencies between the packages defined within the model Application Schema you are defining, identifying any circular references or inconsistencies / unexpected dependencies.

Currently this tool does not remove any existing automatically generated package dependency diagram. This is a recommendation for Solid Ground that has originated from this refactor effort.

5.8 Adding Sequence Number Tagged Values

Attributes and operations in UML are not considered to have any order that can be inferred from class definitions. To allow the conceptual model to capture rules on the order of attributes, sequence number tags are used. UML elements can include additional name / value pairs called tags. Solid Ground will automatically apply sequence numbers which may be manually adjusted if required.

5.9 Documenting the Model

AIXM has been meticulous model documentation, generally all attributes, classes and packages include appropriate comments in the notes for the element. Solid Ground includes a tool to simplify documentation by providing an Excel import / export that takes all elements into an Excel file with a notes column. This column can be filled out and imported back to the document. The generated spreadsheet includes the following fields for convenience:

- Identifier
- Package
- Parent
- Name
- Type

- Stereotype
- Description

5.10 Moving Code Lists to it's own Package

Code lists commonly need more detailed / specific extension when used external to the application schema they are defined in. Isolating the code list into its own package allows code lists to be easily maintained / extended in isolation from the rest of the model. They are also easier to make use of when they are available in isolation.

At the moment a change to one value in a code list, or extending that list would require all of AIXM to be versioned.

5.11 Publish to the Model Registry

Beyond simplifying collaboration with others, publishing to the model registry performs checks the model is in a state that can be minimally supported in the Model Registry. This mainly centres on ensuring the dependencies are appropriately available in the registry and are not circular.

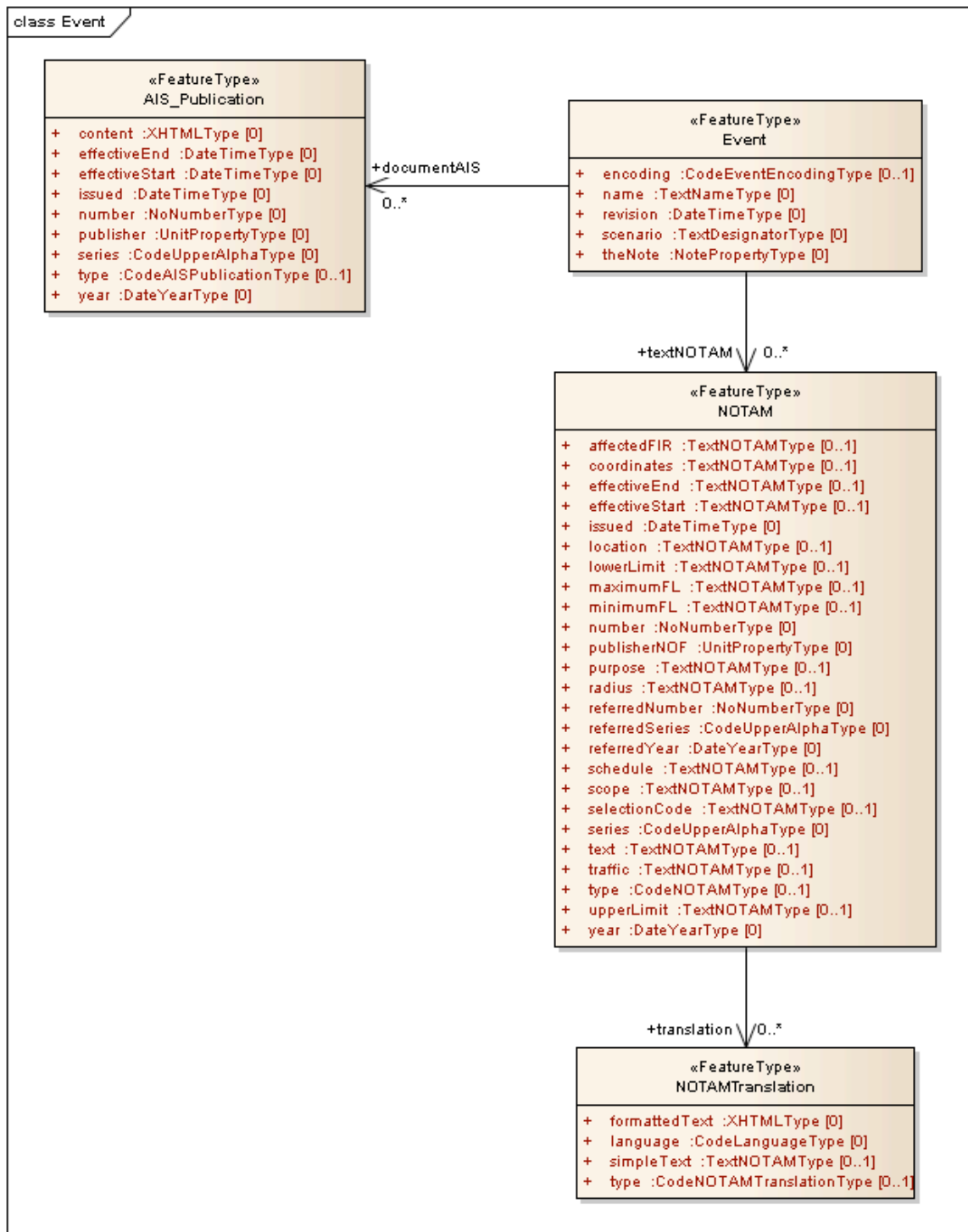


Figure 6 – Refactored Event Model

6 Nillable Attributes

Nillability of values is an important requirement of AIXM. An example is in the temporality model where a delta will omit mention of an attribute implying the value does not change, or the value is nilled which has a different implication.

Recent discussion on this topic has been captured on the SEEGrid Community Website - <https://www.seegrid.csiro.au/wiki/CGIModel/NillableVoidableDiscussion>.

Considerations on what approach should be taken for modelling nillability include:

- The approach used should be able to be represented in UML diagrams, and interpreted by tools that may process the model.
- Nillability refers to a conceptual detail, not an implementation detail. Implementation details should not be represented in conceptual models.
- Different domain modelling activities need to interpret a nil value in the same way. GeoSciML interprets a nil value as "the characteristic is not present in the spatial object, but is present in the real world.". INSPIRE uses the voidable stereotype for the same purpose.
- Modelling overhead. The approach for modelling nillability should not introduce an unfeasible overhead to modelling, such as the need to manually perform a frequently repeated task.
- What can be nillable? Class attributes? Association roles?

This issue has seen discussion in INSPIRE. interesting points made in this discussion included:

- The practical need for a special rule should be established before investing in the effort.
- Different interpretations of no value need different representations, but leads to more verbose encoding.
- Limited value in supporting nillable if modellers / implementers are not going to use it or even be aware of it.

There is contention in how to model nillability in UML, currently with no clear recommendation on best practice. This issue is being considered in revisions to ISO 19103 / 19109. The two lead options in current use are discussed below.

6.1 Nillable Tagged Value

A tagged value “nillable” on attributes that can be nilled.

Benefits:

- Common approach to providing interpretation hints.
- Tagged values can have an automatic default value.

Drawbacks:

- Not usually represented in UML diagrams.
- Not integral to UML models - tagged values are often ignored by tools unless one is specifically considered.
- Common approach to providing encoding hints. May lead to misinterpretation that voidable is a conceptual level profile of implementation detail, not an encoding detail.

6.2 Voidable Stereotype

Apply a stereotype that would be defined in ISO 19103 to nillable elements.

Benefits:

- Represented in UML diagrams
- More compatible with INSPIRE
- Stereotypes may be able to apply tagged values, a potential solution that gains the benefits of both proposals without some of the drawbacks.
- You can stereotype associations / association roles.

Drawbacks:

- A proliferation of the use of stereotypes leads to more special cases that need to be catered for by modellers, tool implementers etc.
- Handling reuse of concepts across domain boundaries leads to the need to handle interpretation of stereotypes that a domain may not have had need of.
- An inheritable meta-class may not be the most appropriate way to model what is an implementation profile issue.

Recommendation

It is important for the custodians of AIXM to be aware of the effort in resolving nillability in the ISO standards. Implementation of any change should wait until this has been resolved. As a major international domain, aviation has a considerable stake in the outcomes of this discussion and should consider the options for nillability, providing feedback to the group based on their experiences.

7 Temporality

AIXM has a critical need to provide metadata information on features that describe how the feature changes over time, as detailed in the AIXM Temporality Concept document. The need for time-variant information is common to a variety of domains, and the functionality provided by the AIXM approach is of value to these other domains.

These temporality concerns are relevant to many domains. In the context of the ISO 19100 series of standards, previous attempts at standardising the approach to time-variant information include DynamicFeature in ISO 19136 - Geographic information - Geography Markup Language (GML). While this package has benefited from considered collaboration and evaluation in the past, the work is incomplete and it can only be considered a placeholder for a complete dynamic feature implementation.

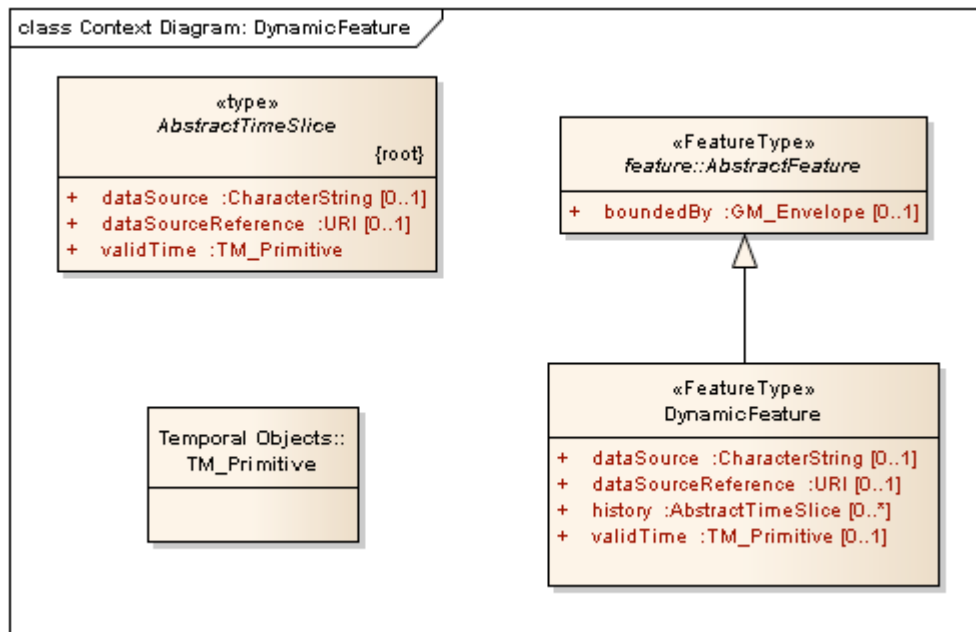


Figure 7 – ISO 19136 - DynamicFeature Class Diagram

AIXM have progressed the temporality features of the DynamicFeatures package within the domain of AIXM to meet aviation needs. Some of the common concepts include time slices and valid time, while some of the features added to meet the needs of AIXM include sequence numbers, correction numbers, deltas and schedules.

7.1 Modelling Approach

Currently the temporality model is not represented in the domain modelling of AIXM, instead it is incorporated during the conversion from the domain model to the physical model and implemented in AIXM encoding. It is also important to ensure that physical model encoding and implementation detail is not captured in the domain model.

Three options for introducing temporality into the model have been considered and are represented here. All three patterns allow baseline features to be fully modeled as well as deltas to be produced. Each option has various benefits and drawbacks, using RunwayDirection as an example time-variant feature type. These three patterns are compared for clarity and to compare how easily they can be processed.

Pattern 1: Special Feature Stereotype

In this pattern the time-variant nature of the class is inferred from a special feature stereotype.

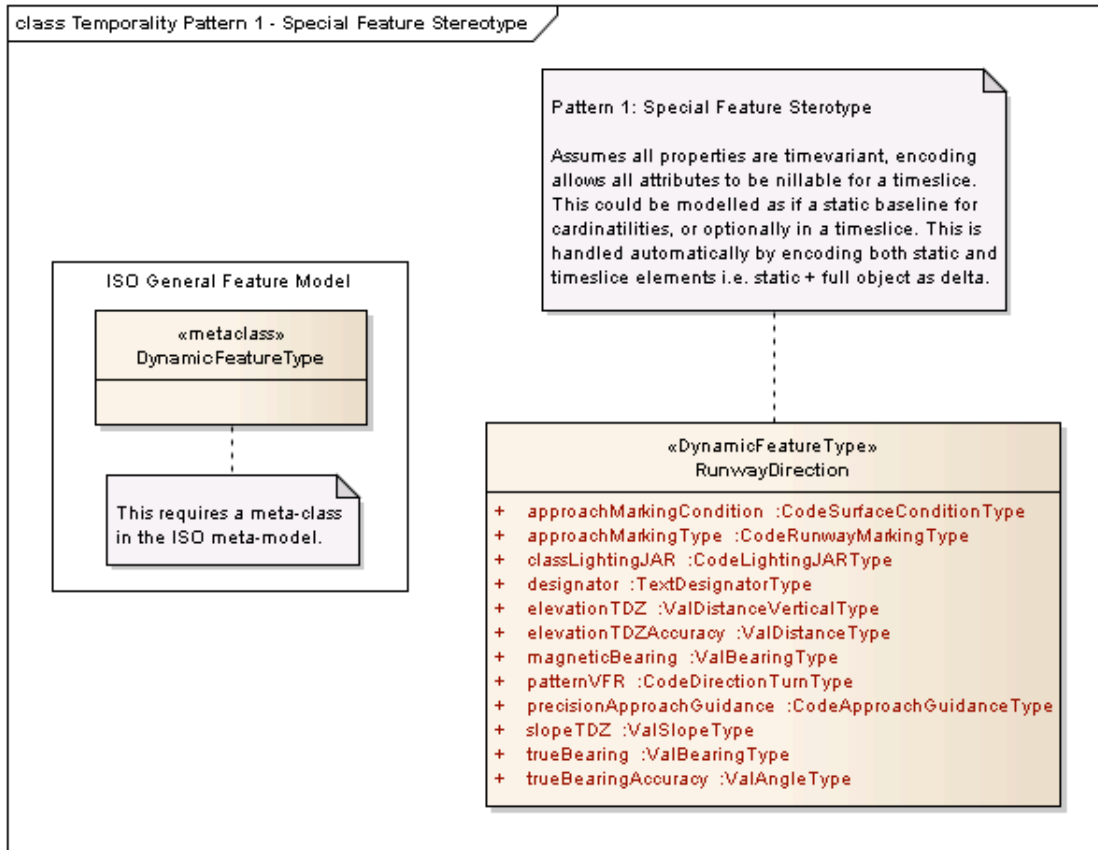


Figure 8 – Special Feature Stereotype - DynamicFeatureType

Benefits: Simplifies modelling. Nillability of attributes can be dealt with automatically for deltas.

Drawbacks: All attributes are always time-variant.

Conclusion: Not flexible enough to meet varied needs.

7.2 Pattern 2: Tagged as Time-variant

In this pattern a stereotype is applied to each attribute that is time-variant.

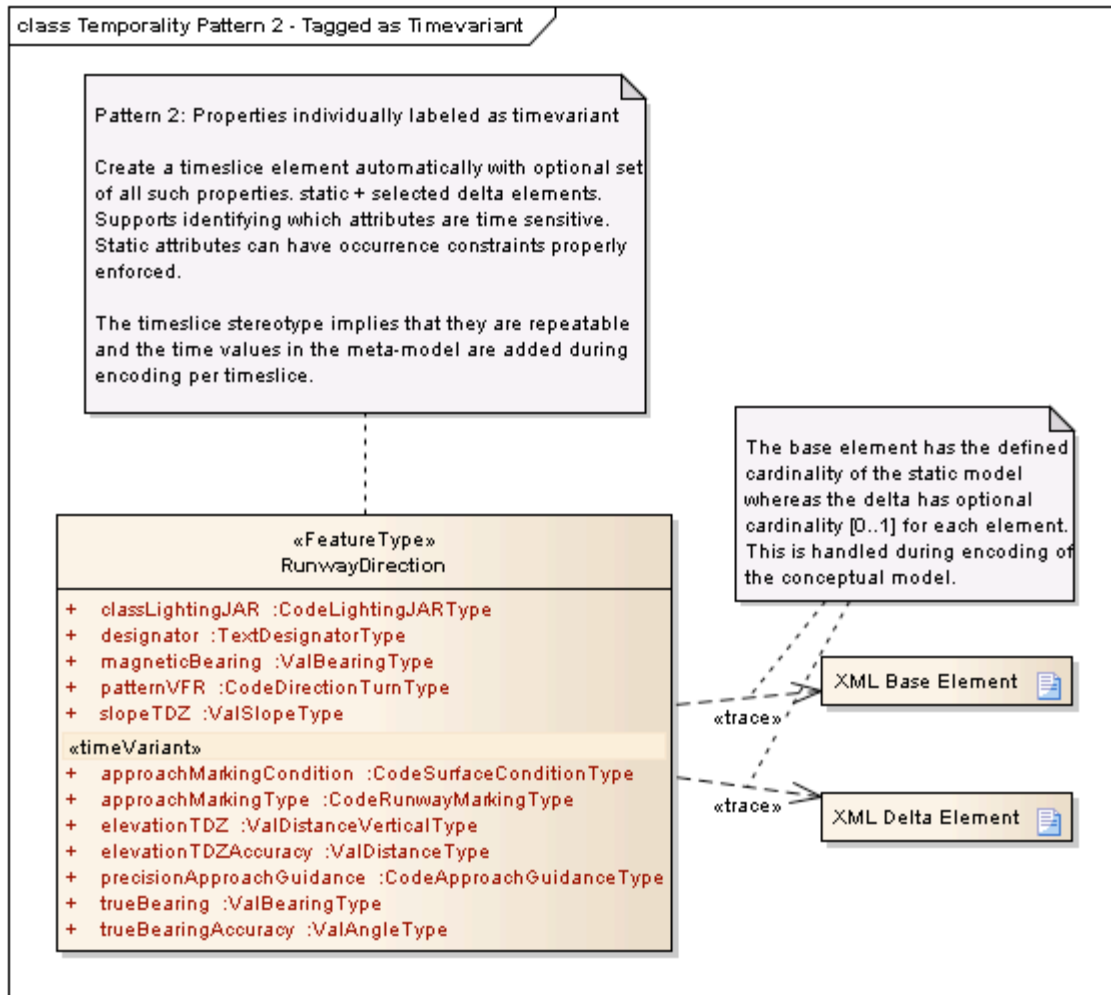


Figure 9 – Properties individually labeled as time-variant

Benefits: Accurate modeling at the attribute level. Simplifies modeling and interpretation of the models time sensitivity. Refactor of AIXM and other existing models would not be overly costly.

Drawbacks: Implementation in the ISO standards requires considerable effort and will ideally impact a variety of ISO documents (19103, 19109, and 19136).

Conclusion: This pattern is the ideal solution and it is recommended that this approach is considered for the ISO standards. Achieving this would take considerable co-ordination and effort though, so an interim solution is also to be considered.

7.3 Pattern 3: Timeslice Feature with Time-variant Properties

This pattern involves manually modeling the portion of a feature that is time-variant. This approach will need to include modeling of the timeslice and dynamic feature type stereotypes within the AIXM model.

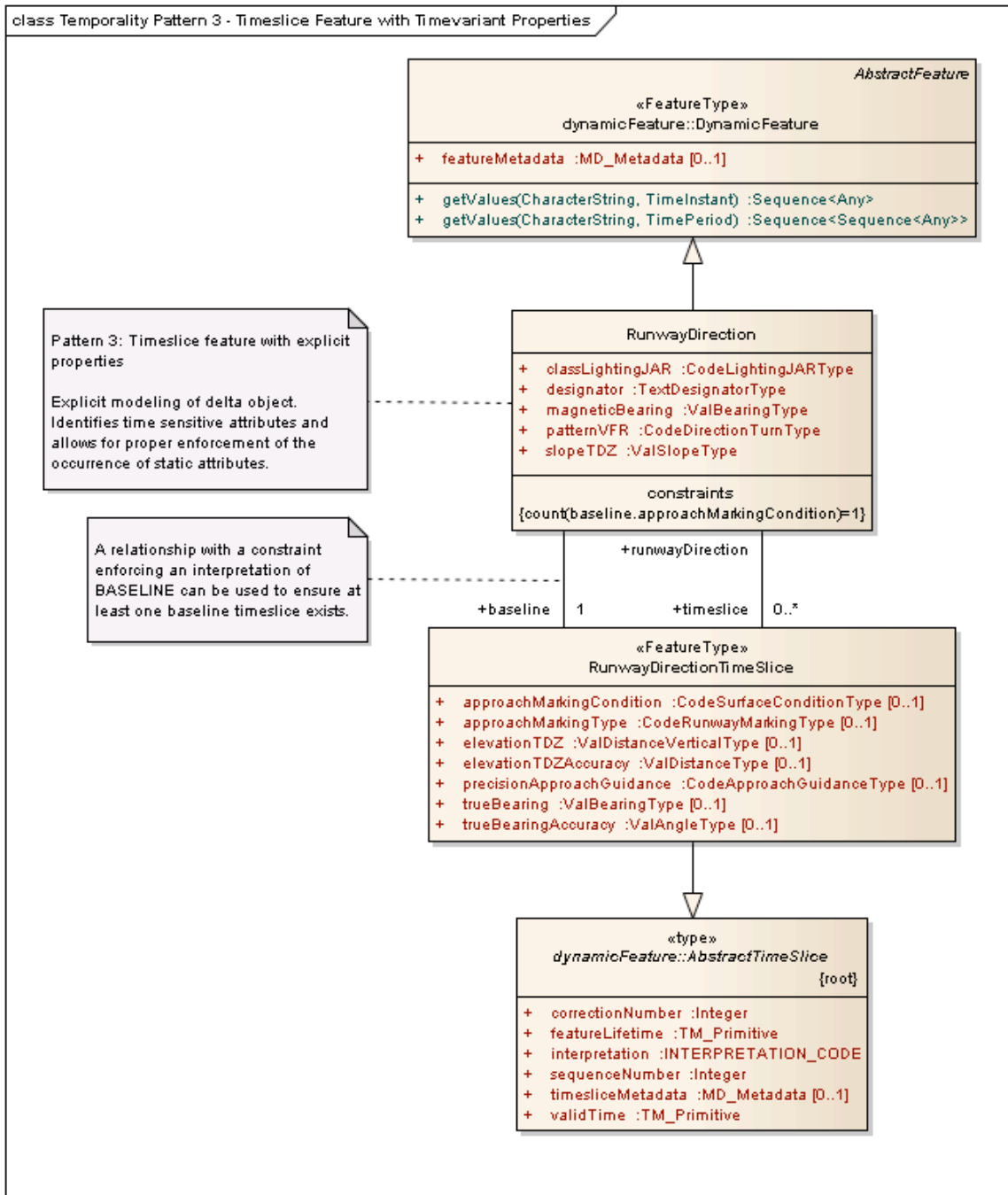


Figure 10 – Timeslice feature with explicit properties

Benefits: Accurate modeling at the attribute level. Uses existing idiom, and will work with existing tooling.

Drawbacks: Manual modeling of timeslice features required. Different domains may do inconsistently (we have had to modify the ISO 19136 base class to meet our needs.)

Conclusion: This pattern is a feasible interim step while the ISO harmonized model does not include the proposed dynamic feature improvements.

7.4 Temporality in ISO Standards

Introducing the stereotype for time-sensitive features would also affect ISO 19103, which defines the profile of UML for defining geographic information. Ideally this stereotype for dynamic features would be included here.

As a recommendation for ISO 19103, this standard should explicitly note that the set of stereotypes can be extended, it is not a closed set.

19108 is about defining and describing time, so it is not as appropriate as 19109 for describing how to make use of the stereotypes introduced to 19103 and what they mean.

How to make use of dynamic features and what they mean needs to be described as well. Ideally this would be as a separate addition to ISO 19109, e.g. 19109 Part 2. As it is a distinct extension, temporality can be safely ignored by those not interested.

The DynamicFeature package in ISO 19136 needs to be progressed to support the requirements of AIXM temporality that are generally applicable to time-variant geographic information, based on the AIXM temporality work.

OWS-8 Aviation Architecture ER (OGC 11-093) provides a detailed proposal for refactoring the AIXM Temporality Model into a standalone standard. Revisions and extensions to the OGC/ISO standards are suggested, including suggestions for the Filter Encoding (FES) and Web Feature Service (WFS) specifications to support temporally aware operations.

7.5 Summary

- Represent temporality detail in the conceptual domain model that captures conceptual detail, for instance RunwayDirections do have time-variant attributes.
- Drive encoding of temporality into the physical model based on the conceptual representation of time-variant features, along with a conceptual-to-physical model mapping step.

- Progress the introduction of these dynamic feature recommendations to the ISO standards as described above.
- Implement the updated DynamicFeature package when it is ready to replace the AIXM Temporality implementation.

7.6 Work Performed

During OWS-8, the DynamicFeature package was cloned into the digital NOTAM Events refactor project and adjusted to incorporate features of the AIXM temporality model. The appropriate sample classes in the refactored Events package that are time-variant (Event and RunwayDirection) were adjusted to make appropriate use of the updated DynamicFeature package.

This involved a separation of the conceptual model for Events from the encoding method for AIXM so that no encoding implementation details are inappropriately present in the conceptual model. Encoding implementation details are handled as part of the conceptual model to physical model mapping, keeping the conceptual model free of implementation specifics, but remaining an accurate representation of the concepts.

7.7 Usability of Time-variant Features

The usability of interacting with time-variant features once encoded is an issue that needs to be addressed. Currently it is difficult to easily perform operations (selects and queries) on time-variant data. Consider a WFS that provides time-variant information as an example. Performing requests that are aware of the temporal nature of the data is not trivial given the current encoding. To this end it would be beneficial to model the operations that a dynamic feature will support, allowing implementers to know what they need to do to appropriately interact with dynamic features. These operations could then be implemented where appropriate, OWS-8 Aviation Architecture ER (OGC 11-093) provides detailed suggestions for how these operations could be implemented in FES and WFS.