# Open Geospatial Consortium

Approval Date: 2012-10-16

Publication Date: 2013-06-18

Reference number of this OGC® project document: OGC 12-027r2

Reference URL for this document: http://www.opengis.net/doc/dp/WFS-TE

Category: OGC® Discussion Paper

Editor: Timo Thomas

## OGC Web Feature Service (WFS) Temporality Extension

### Warning

This document is not an OGC Standard. This document is an OGC Discussion Paper and is therefore 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, an OGC Discussion Paper should not be referenced as required or mandatory technology in procurements.

| | |
|---|---|
| Document type: | OGC® Discussion Paper |
| Document subtype: | NA |
| Document stage: | Approved for public release |
| Document language: | English |

## Contents

1 Scope ........ 1

2 Compliance ........ 1
2.1 Conformance Classes ........ 1

3 References ........ 2

4 Conventions and Abbreviated Terms ........ 3

5 AIXM 5 Temporality Model ........ 4
5.1 Issues with GML DynamicFeature Model ........ 4
5.2 Time Slice Interpretation ........ 6
5.3 Sequences, Corrections and Cancelations ........ 7
5.4 Properties with Schedules ........ 7
5.5 Differences to the GML Dynamic Feature Model ........ 8
5.6 Conclusion ........ 9

6 Use Cases for a WFS Service Serving AIXM 5 Data ........ 10
6.1 Data Retrieval ........ 10
6.2 Data Storage ........ 12

7 Limitations of the WFS/FES 2.0 standards when applied to AIXM data ........ 12
7.1 GetFeature ........ 13
7.1.1 Ad Hoc Queries ........ 13
7.1.2 Stored Queries ........ 14
7.2 GetPropertyValue ........ 15
7.3 Summary ........ 16

8 Dynamic Feature Query ........ 16
8.1 Overview ........ 16
8.2 DynamicFeatureQuery ........ 17
8.3 DynamicFeatureFilter ........ 18
8.4 SnapshotGeneration ........ 20
8.5 TimeSliceProjection ........ 24
8.6 PropertyExclusion ........ 26

9 Indicating Support for Temporality Extension ........ 27

10 Compatibility with Existing WFS 2.0 Based Systems ........ 28

11 Alternative Approaches ........ 28
11.1 Web Processing Services ........ 28
11.2 XQuery and XSLT ........ 29

**12 Future Work ... 30**
**12.1 Formal Work ... 30**
**12.2 Advanced Filtering: valueFor() XPath function ... 30**

## i. Abstract

This OGC Discussion Paper (DP) provides a proposal for a temporality extension for the WFS 2.0 and FES 2.0 standard. It is based on the work of and experiences made in several OWS test beds, in particular OWS-7 and OWS-8, Aviation threads and discussions at the 2011 OGC TC meeting in Brussels, Belgium. This DP partially replaces and advances the document “OWS-8 Aviation: Guidance for Retrieving AIXM 5.1 data via an OGC WFS 2.0” [4].

## ii. Keywords

ogcdoc, ogc document, aviation, aixm, wfs, fe, temporality

## iii. Submission contact points

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

| CONTACT | COMPANY |
|---|---|
| Timo Thomas | COMSOFT GmbH |
| Johannes Echterhoff | iGSI GmbH |
| Jeroen Dries | Luciad NV |

## iv. Revision history

| Date | Release | Author | Paragraph modified | Description |
|---|---|---|---|---|
| 2012-10-16 | 1.0 | Timo Thomas, | all | Initial draft for general review. |

| | | | | |
|---|---|---|---|---|
| | | Johannes Echterhoff | | |
| | | | | |
| | | | | |

## v. Foreward

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights. However, to date, no such rights have been claimed or identified.

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 specification set forth in this document, and to provide supporting documentation.

## Introduction

The Aeronautical Information Exchange Model (AIXM) is designed to enable the management and distribution of Aeronautical Information Services (AIS) data in digital format. The newest version of this model, AIXM 5.1, is based on GML 3.2 and features an exhaustive temporality model loosely based on the GML Dynamic Feature Model.

Various interoperability test-beds at OGC, in particular OWS-7 and OWS-8, have applied OGC's WFS 2.0 and FES 2.0 standards on AIXM 5 data. Though it could be demonstrated that a basic interoperability is possible, it turned out that some key requirements could not be fulfilled. This paper summarizes the observations made and shows that these requirements are not specific to AIXM 5, but more generally apply to any data model featuring temporality. To overcome these shortcomings, a proposal is made for an extended type of WFS query: a dynamic feature query.

# 1 Scope

This document specifies the AIXM 5 temporality extensions of the WFS 2.0 and FES 2.0 standards. It identifies the required functionality and use cases, analyses the existing WFS and FES standards, documents their limitations and defines extensions to overcome them.

# 2 Compliance

## 2.1 Conformance Classes

The following tables specify the conformance classes defined by this specification. Table 1 specifies the conformance classes that cover the features of the Temporality Extension that extend FES 2.0 while Table 2 lists those classes that cover features extending WFS 2.0.

**Table 1 — FES Temporality Extension Conformance Classes**

| Conformance class name | Conformance class identifier | Operation and/or behavior |
|---|---|---|
| FES-TE-Core | http://www.opengis.net/spec/FES-TE/1.0/conf/Core | The server implements the DynamicFeatureFilter, SnapshotGeneration and TimesliceProjection defined by this specification as well as the conformance class that this conformance class depends upon (FES Ad hoc Query, see Figure 1).<br>The valueFor() function, schedule evaluation for time periods and property exclusion are not covered via this conformance class. |
| FES-TE-TimePeriodScheduleEvaluation | http://www.opengis.net/spec/FES-TE/1.0/conf/TimePeriodScheduleEvaluation | The server implements schedule evaluation for time periods defined by this specification as well as the conformance class that this conformance class depends upon (FES-TE-Core, see Figure 1). |
| FES-TE-PropertyExclusion | http://www.opengis.net/spec/FES-TE/1.0/conf/PropertyExclusion | The server implements property exclusion defined by this specification as well as the conformance class that this conformance class depends upon (FES-TE-Core, see Figure 1). |

**Table 2 — WFS Temporality Extension Conformance Classes**

| Conformance class name | Conformance class identifier | Operation and/or behavior |
|---|---|---|
| WFS-TE-Core | http://www.opengis.net/spec/WFS-TE/1.0/conf/Core | The server implements the DynamicFeatureQuery defined by this specification as well as the conformance classes that this conformance class depends upon (FES-TE-Core and Basic WFS, see Figure 2). |

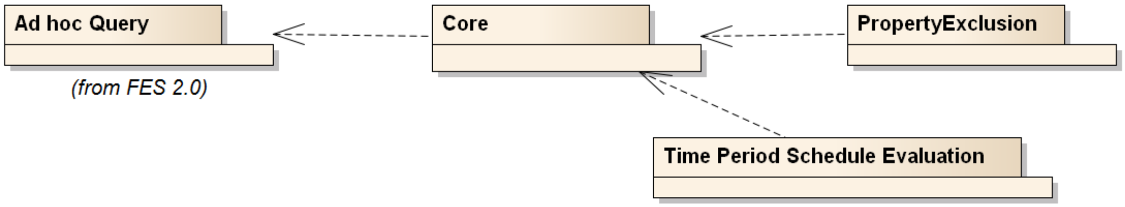


**Figure 1: FES Temporality Extension Conformance Classes**

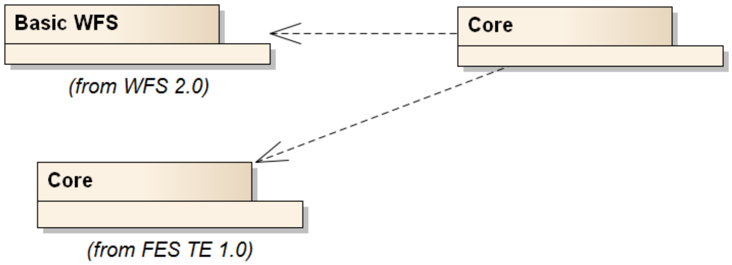


**Figure 2: WFS Temporality Extension Conformance Classes**

## 3 References

The following normative documents contain provisions that, through reference in this text, constitute provisions of 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.

[1] ISO/DIS 19142 and OGC 04-094, OpenGIS® Web Feature Service 2.0 Interface Standard (2010-11-02)

[2] ISO/DIS 19143 and OGC 09-026r1, OpenGIS® Filter Encoding 2.0 Encoding Standard (2010-11-22)

[3] OGC OGC 11-093r2, OGC® OWS-8 Aviation Architecture ER (2011-09-30)

[4] OGC 11-073r2, OGC® OWS-8 Aviation: Guidance for Retrieving AIXM 5.1 data via an OGC WFS 2.0 (2011-11-04)

[5] OGC 10-131r1, OGC® OWS-7 Aviation – AIXM Assessment Report (2010-08-18)

[6] OGC 05-007r7, OpenGIS® Web Processing Service (2007-06-08)

[7] XQuery 1.0: An XML Query Language (Second Edition), W3C Recommendation 14 December 2010, at http://www.w3.org/TR/2010/REC-xquery-20101214/

[8] OGC 11-171, Requirements for Aviation Metadata - OGC Best Practice (2011-11-10)

[9] OGC 11-172, Guidance on the Aviation Metadata Profile (2011-11-10)

[10] AIXM 5 Temporality Model 1.0 (2010-09-15)

[11] Lake R. et al., GML Geography Mark-Up Language, Wiley 2004

# 4 Conventions and Abbreviated Terms

| | |
|---|---|
| AIP | Aeronautical Information Publication |
| AIRAC | Aeronautical Information Regulation And Control |
| AIXM | Aeronautical Information Exchange Model |
| AIXM-TM | AIXM 5 Temporality Model |
| API | Application Program Interface |
| ER | Engineering Report |
| FES | Filter Encoding Specification (Version 2.0 – if not stated otherwise) |
| GML | Geography Markup Language |
| HTTP | Hypertext Transfer Protocol |
| ISO | International Organization for Standardization |
| NOTAM | Notice To Airmen |

| | |
|---|---|
| OGC | Open Geospatial Consortium |
| UML | Unified Modeling Language |
| URL | Uniform Resource Locator |
| W3C | World Wide Web Consortium |
| WFS | Web Feature Service (Version 2.0 – if not stated otherwise) |
| WPS | Web Processing Service |
| XML | eXtended Markup Language |
| XPath | XML Path Language |

# 5 AIXM 5 Temporality Model

AIXM 5 is based on GML 3.2. All features inherit from the type **gml:DynamicFeature** which is the base class of the GML Dynamic Feature Model (DFM). The properties of a feature are all time-variant and encoded in time slices. The only exceptions to this are global identifiers, names, metadata and a bounding box. A detailed comparison of the two models is given in 5.5.

The following section describes issues with the GML DynamicFeature Model. Then a brief introduction is given to the AIXM Temporality Model and to what makes it special compared to the Dynamic Feature Model in GML 3.2.

## 5.1 Issues with GML DynamicFeature Model

The GML 3.2.1 standard [OGC 07-036] defines the UML model and XML Schema encoding of the DynamicFeature, DynamicFeatureCollection and AbstractTimeSlice types. The UML model is provided in [OGC 07-036] D.3.11 while the XML Schema is defined in section 14.5 of that document.

The GML standard provides an example (see [OGC 07-036] section 14.5.7) in which a cyclone is modeled as a DynamicFeature, with its movement status captured in time slices that together constitute the history of the cyclone.

GML does not define modeling and management aspects of dynamic features in sufficient detail:

- Modeling/encoding:

- 
    - GML does not clearly specify the rules for modeling and encoding dynamic features.
        - Section 14.5.1 says that "… dynamic feature classes will normally be extended to suit particular applications."
        - Section 14.5.3 states that "this [the gml:dynamicProperties group] allows an application schema developer to include dynamic properties in a content model in a standard fashion." That the gml:dynamicProperties group allows the inclusion of dynamic properties in a content model causes confusion, as it may suggest that it is this group which needs to be extended by an Application Schema developer – although no UML type exists for this group.
        - Section 14.5.6 says: "A timeslice encapsulates the time-varying properties of a dynamic feature - it shall be extended to represent a time stamped projection of a specific feature."
    - These statements suggest that the DynamicFeature and AbstractTimeSlice types are extended when modeling a specific DynamicFeature type in an Application Schema by adding the same properties to both of them. This would then enable snapshots as well as the full feature history to be represented. However, this is not explicitly stated in the standard. It is also not explicitly stated that all of the dynamic feature properties would need to be optional to enable representation of snapshots and the complete feature history.
- Management:
    - Section 14.5.4 in GML states that "each time-stamped instance [of a dynamic feature] represents a 'snapshot' of a feature." This appears to refer to the gml:validTime, which is an optional property of a GML DynamicFeature type. However, the relationship to gml:validTime is not explicitly stated there. Section 14.5.5, though, states that "a gml:DynamicFeatureCollection is a feature collection that has a gml:validTime property (i.e. is a snapshot of the feature collection)" which provides a hint that this assumption is correct.
    - GML also does not clearly define what a snapshot is if the 'time-stamp' (presumably the gml:validTime) is not a gml:TimeInstant (or a gml:TimeNode). Again the reader is forced to make assumptions, though an obvious assumption would be that if a gml:TimePeriod or –Edge is given for a 'snapshot' that then the state of the dynamic feature is constant for that time.
    - GML does not define how to solve situations in which multiple time slices contain a value for a dynamic property for a given point in time. The AIXM-TM avoids this potential ambiguity through the definition of sequence and correction numbers for time slices.
    - Other application aspects that are covered in the AIXM-TM, such as deltas for complex or multi-occurring properties, or canceling a time slice are not defined in GML.

Apparently the GML standard only defines the basic model and encoding of the dynamic feature base type. Actual rules for modeling/encoding and management of specific dynamic features that are defined in an Application Schema are neither covered in GML 3.2.1 nor in the current OGC or ISO standards baseline.

### 5.2 Time Slice Interpretation

AIXM 5 distinguishes four types of time slices: BASELINEs, PERMDELTAs, TEMPDELTAs and SNAPSHOTs. The time slice type is encoded in a time slice property called 'interpretation'. From now on, the term *time slice interpretation* may be used as a synonym to *time slice type* in this document. The function of the time slice types are described in Table 3. BASELINEs and SNAPSHOTs are the direct result of PERMDELTAs and TEMPDELTAs, which means that they are only a different representation of the information carried in the delta time slices. Their generation is the result of a merging process in which overlapping deltas are ordered according to their sequence number (which is a property of all AIXM 5 time slices) to define a unique and unambiguous result.

**Table 3 — Time Slice Types in AIXM 5**

| Time Slice Type / Interpretation | Function |
|---|---|
| PERMDELTA | Contains all properties that change permanently; valid at a time instant. Example: a change in the length of a runway due to construction works. |
| BASELINE | Current state of the feature as the result of permanent changes; valid for a time period (= the sum of all PERMDELTAs relevant for the time period) |
| TEMPDELTA | Contains all properties that change temporarily; valid for a time period. Example: a closure of a runway due to icing. |
| SNAPSHOT | Current state of the feature as the result of all permanent and temporary changes; valid for a time instant[A]. (= the sum of all PERMDELTAs and TEMPDELTAs relevant for the time instant) |
| [A] The concept of snapshots is extended to time periods later in this document | |

SNAPSHOT time slices cannot be stored in a WFS and a WFS cannot retrieve them from a persistent storage. This is because in theory, an unlimited number of SNAPSHOT time slices exist as they are valid for a single point in time.

### 5.3 Sequences, Corrections and Cancelations

An AIXM 5 feature never forgets its history, which includes expected events on the state of a feature to happen in the future, that were known at some point in the past. This implies that a time slice is never deleted or updated. Every change is communicated through the insertion of new time slices, for which overwrite rules apply. For this purpose AIXM 5 introduced the concepts of *sequences*, *corrections* and *cancelations*. Every time slice of a feature has a unique identifier (key) which consists of the following properties:

- the interpretation (time slice type),
- the sequence number and
- the correction number

For every sequence (i.e. the set of time slices that share a sequence number), there is always only one *active* time slice: it is the one with the highest correction number. All time slices with a lower correction number (in a sequence) are considered to be outdated. If the active time slice is a cancelation, which is indicated by a specific value of the validTime property, the sequence is considered to be canceled[1]. This functionality makes possible to communicate both updates and deletions without losing the history. For simplification we define:

- a *sequence*: the set of time slices of a feature that share the same interpretation value and sequence number
- the *active time slice*: the time slice of a sequence with the highest correction number
- a *canceled sequence*: a sequence whose active time slice is a cancelation
- *canceled time slices*: the time slices of a canceled sequence
- *corrected time slices*: all time slices of a sequence except the active time slice

### 5.4 Properties with Schedules

In AIXM 5, a special property type exists for modeling periodic events: **PropertiesWithSchedules** [10]. It introduces an additional temporality aspect to properties which is overlaid onto the underlying temporality model. Good examples for schedules are the opening times of an airport (e.g. daily 8-18h) and the regular activation of airspaces for military operations (e.g. weekly on Saturday 8-16h). Schedules consist of a list of consecutive time sheets, where each time sheet contains a validity period and set of values.

[1] A canceled sequence is always closed and cannot be re-opened by the insertion of another time slice with a higher correction number.

### 5.5 Differences to the GML Dynamic Feature Model

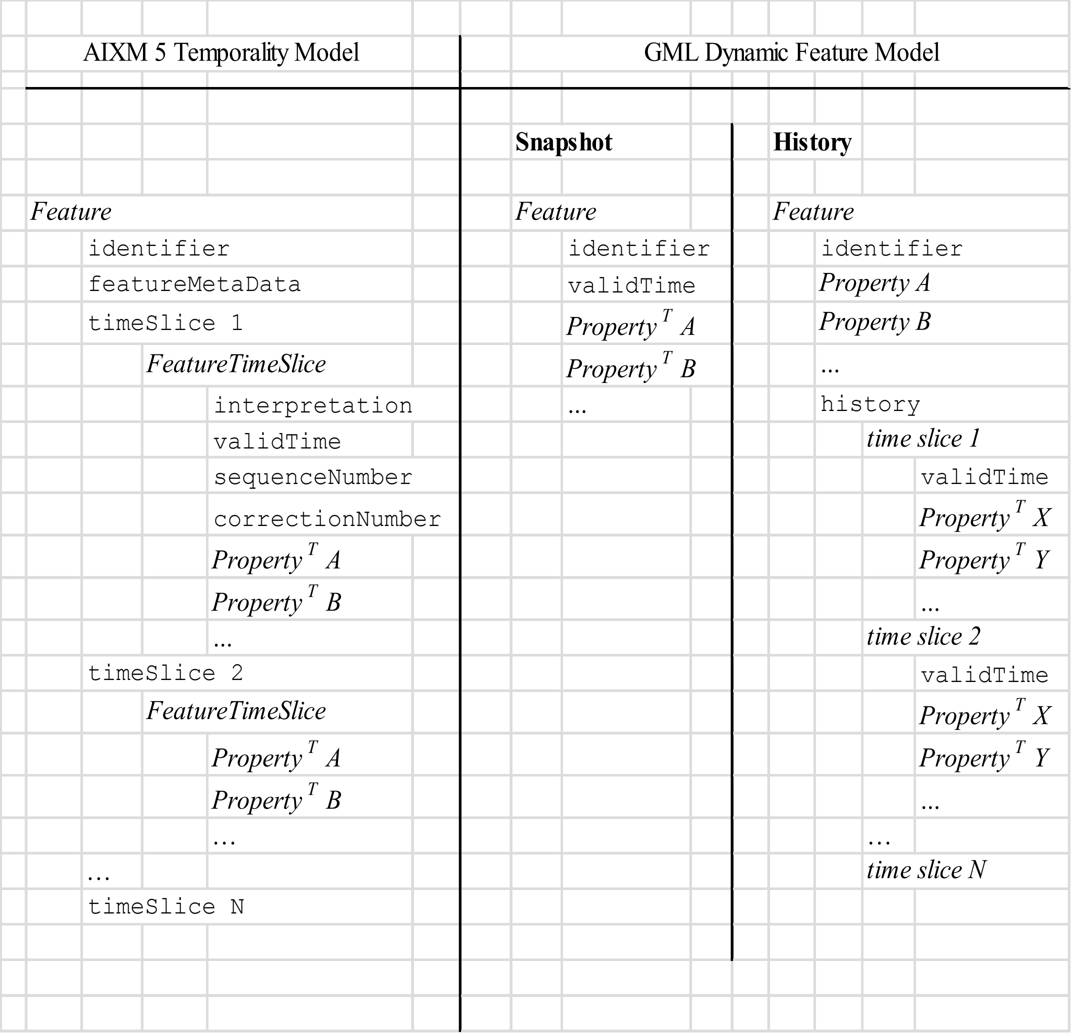
AIXM 5 is only loosely based on the GML Dynamic Feature Model. Important differences exist both

- on a conceptual level (see Table 4) as well as
- in the structure of the features (see Figure 3).

**Table 4 — Differences between AIXM and GML temporality models, conceptual level**

| Concept | AIXM | GML 3.2 |
|---|---|---|
| Base types | gml:DynamicFeatureType, but restricts the inheritance to only a few elements. gml:validTime and gml:history are not among them | gml:DynamicFeatureType |
| Types of temporary changes | Time slice types: SNAPSHOT (current state), PERMDELTA (permanent change), TEMPDELTA (temporary change), BASELINE (current permanent state, excluding temporary changes) | Snapshots[2] (current state) and history (list of changes in time-variant properties) |
| Corrections | Supported through sequence and correction numbers | No such concept |
| Cancelations | Supported through special interpretation of validTime | No such concept |
| Schedules | Supported through interpretation of special timeInterval properties | No such concept |

[2] What exactly a snapshot is is not clearly defined in GML 3.2.1. Here, we refer to the definition in [11] section 14.5, which is “A ‘snapshot’ is a representation of the state of a feature at a point in time or during a some interval”



| AIXM 5 Temporality Model | | | | GML Dynamic Feature Model | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | **Snapshot** | | **History** | | | |
| *Feature* | | | | *Feature* | | *Feature* | | | |
| | `identifier` | | | | `identifier` | | `identifier` | | |
| | `featureMetaData` | | | | `validTime` | | *Property A* | | |
| | `timeSlice 1` | | | | *Property*$^T$ *A* | | *Property B* | | |
| | | *FeatureTimeSlice* | | | *Property*$^T$ *B* | | ... | | |
| | | | `interpretation` | | ... | | `history` | | |
| | | | `validTime` | | | | | *time slice 1* | |
| | | | `sequenceNumber` | | | | | | `validTime` |
| | | | `correctionNumber` | | | | | | *Property*$^T$ *X* |
| | | | *Property*$^T$ *A* | | | | | | *Property*$^T$ *Y* |
| | | | *Property*$^T$ *B* | | | | | | ... |
| | | | ... | | | | | *time slice 2* | |
| | `timeSlice 2` | | | | | | | | `validTime` |
| | | *FeatureTimeSlice* | | | | | | | *Property*$^T$ *X* |
| | | | *Property*$^T$ *A* | | | | | | *Property*$^T$ *Y* |
| | | | *Property*$^T$ *B* | | | | | | ... |
| | | | … | | | | | … | |
| | … | | | | | | | *time slice N* | |
| | `timeSlice N` | | | | | | | | |

**Figure 3 — Comparison of the XML structure in the AIXM 5 Temporality Model and the GML Dynamic Feature Model**

Figure 3 explained: each item below "Feature" represents an XML element (including the Feature element itself). Because XML can be quite verbose, tags, attributes and namespaces are completely left out as they are of little value in this comparison. Elements in `type-writer notation` denote real existing elements, whereas elements in italic letters denote elements of a certain type or class. Feature properties fall into two classes: time-variant properties (*Property*$^T$) and time-invariant properties (*Property*). The latter only exist in the GML Dynamic Feature Model.

### 5.6 Conclusion

Despite the AIXM Temporality Model being based on the dynamic feature model of GML through the common base type gml:DynamicFeature, in practice there is little in common. AIXM-TM introduces a number of important new concepts: the distinction between temporary and permanent changes, corrections, cancelations and properties with schedules. However, both models share the approach of modeling time-variant data by the introduction of time slices, which associate properties with a validity time. Most important of all is that the AIXM Temporality Model completely defines how dynamic features with time-varying property values are modeled, encoded, interpreted and managed.

The Temporality Extension specified in this document defines the necessary additions to FES 2.0 and WFS 2.0 that support specific data retrieval use cases required by the Aviation domain. The Temporality Extension thus enables better support for retrieval of AIXM data, but in general supports retrieval of dynamic feature data that follows the principles of the AIXM Temporality Model. As outlined in [OGC 11-093r2] the Temporality Model can become a standalone standard that extends the concepts of the General Feature Model (GFM) to support modeling, encoding, interpretation and management of dynamic features.

# 6 Use Cases for a WFS Service Serving AIXM 5 Data

## 6.1 Data Retrieval

The use-cases for the retrieval of AIXM 5 data are manifold and come from different areas of applications. Table 5 categorizes them, gives examples and derives the technical requirements for a query processor.

**Table 5 — Use cases for AIXM 5 data retrieval**

| Ref. No. | Use Case | Category | Example | Technical requirements |
|---|---|---|---|---|
| 1 | Retrieve the complete state of a feature at a point in time | Visualization, decision support | Retrieve the state of a runway at the time of arrival. | Filtering of features and generation of SNAPSHOT time slices. |
| 2 | Retrieve all features fulfilling a certain criteria at a point in time | Decision support | Retrieve all airports in a certain area that are operational at the time of arrival. | Filtering of features and generation and filtering of SNAPSHOT time slices. |
| 3 | Retrieve all time slices of a feature relevant for (i.e. affecting the state at) a point in time | Change-aware visualization | Retrieve all time slices of a runway relevant for the time of arrival. (This enables the client to process any TEMPDELTAs (e.g. digital NOTAMs) received at a later point in time). | Filtering of features and determination of the relevant time slices according to the AIXM-TM. |

| | | | | |
|---|---|---|---|---|
| 4 | Retrieve the BASELINE of a feature valid at a point in time | AIP publication | Retrieve the BASELINE of an airport at an AIRAC effective date. | Filtering of features and filtering and aggregating the contained time slices, ignoring corrected and canceled time slices[3]. |
| 5 | Retrieve all TEMPDELTAs of a feature fully or partly valid for a time period and matching a certain criteria | NOTAM communication | Retrieve all TEMPDELTAs affecting the operational status of a navigation aid until the time of arrival. | Filtering of features and filtering the contained time slices. |
| 6 | Retrieve all PERMDELTAs of a feature valid for a time period | Replication | Retrieve all PERMDELTAs of a feature valid now or in the future. | Filtering of features and filtering the contained time slices, ignoring corrected and canceled time slices. |
| 7 | Retrieve specific time slices of a feature by their unique id | Replication, technical verification | Retrieve the PERMDELTA time slice of feature X, sequence number Y, correction number Z. | Filtering of features and filtering the contained time slices. |
| 8 | Retrieve the full history of a feature | Backup | Retrieve all time slices of a feature (BASELINEs, PERMDELTAs and TEMPDELTAs) valid at any point in time. | Filtering of features. |

Four technical requirements of a query operation on AIXM 5 data follow from this analysis:

a) The ability to filter features and time slices.

b) A filter expression to easily identify corrected and canceled time slices[4].

c) A filter expression to easily identify the time slices that are relevant for a point in time.[5]

d) The ability to generate SNAPSHOT time slices and to filter them.

[3] For details on what BASELINEs are and how to compute them, see the AIXM-TM.

[4] Identifying corrected time slices is not trivial using FES 2.0. The maximum correction number of a sequence has to be calculated for this purpose, which would involve a complex XPath expression.

[5] This is a special case of a), with the option to omit possibly lengthy checks on validTime in the time slice filter (to select the time slices of interest). The difference to d) is that the response does not contain snapshot time slices but rather the actually stored ones.

These requirements will form the basis for the extensions introduced in chapter 8.

### 6.2 Data Storage

Because AIXM-TM does not allow the permanent deletion of information[6], there are only two use cases:

a) insert a feature

b) insert a time slice into a feature

This is already possible with the existing **Transaction** operation. If a feature is inserted, the **Insert** action has to be used. If a time slice is to be inserted, the **Update** action is necessary, because technically, the feature is changed (updated) by adding a new time slice to its list of properties.

EXAMPLE Insertion of a time slice into a feature of type T, with a gml:identifier I. Please note that the order of the time slices is irrelevant in AIXM.

```
<wfs:Transaction>
  <wfs:Update typeName=”T”>
    <wfs:Property>
      <wfs:ValueReference action=”insertAfter”>
        aixm:timeSlice[1]
      </wfs:ValueReference>
      <wfs:Value>
        <aixm:timeSlice>
          ...
        </aixm:timeSlice>
      </wfs:Value>
    </wfs:Property>
    <fes:Filter>
      <fes:PropertyIsEqualTo>
        <fes:ValueReference>gml:identifier</fes:ValueReference>
        <fes:Literal>I</fes:Literal>
      </fes:PropertyIsEqualTo>
    </fes:Filter>
  </wfs:Update>
</wfs:Transaction>
```

## 7 Limitations of the WFS/FES 2.0 standards when applied to AIXM data

In this chapter we will show that there is only little support for the identified requirements in the WFS 2.0 and FES 2.0 standards.

The WFS 2.0 interface provides two operations for retrieving data:

- **GetFeature:** returns a response containing a selection of zero or more features corresponding to the criteria defined in the request

[6] See 5.3 for details

-  **GetPropertyValue:** returns a response containing zero or more of the selected feature property values corresponding to query criteria defined in the request

There is also a **GetFeatureWithLock** operation, but this does not extend the power of the query language compared to **GetFeature** and is therefore ignored in the following. Both the **GetFeature** and **GetPropertyValue** operations support two types of query:

-  **wfs:Query:** these are ad hoc queries generated by a client to retrieve specified feature types or property values

-  **wfs:StoredQuery**: this is a pre-defined parameterized query that has been stored on the server for re-use by clients

In the following it is shown that neither of them fully supports the requirements identified in the previous section.

## 7.1 GetFeature

The **GetFeature** operation request is used to retrieve features using one or more ad hoc queries (**wfs:Query**) or stored queries (**wfs:StoredQuery**).

### 7.1.1 Ad Hoc Queries

The **wfs:Query** element consists of two parts: a selection and a projection part.

The selection part is a FES 2.0 filter expression and selects the features matching given criteria. An additional selection criterion is given by the **typeNames** attribute on the **wfs:Query** element: it is used to restrict the types of features to be returned.

The projection part is optional and may contain one or more projection clauses. In WFS 2.0, there is only one projection clause available: **wfs:PropertyName**. By default, non-mandatory properties of a feature are not returned in the response document unless they are referenced by a **wfs:PropertyName** element in the request. In WFS, the properties of a feature are made up of all first-level child elements of the feature. For AIXM 5 data these are the gml:identifier, gml:name, gml:description, gml:boundedBy, gml:featureMetadata and aixm:timeSlice elements. Except for the list of aixm:timeSlice elements, all of these are non-mandatory. Thus, the **PropertyName** clause enables the client to include gml:identifier, gml:name, gml:description, gml:boundedBy, gml:featureMetadata in the response which are excluded otherwise.

Obviously the PropertyName projection clause was designed for static feature data only, where all properties are encoded in the feature’s first-level child elements. In the DynamicFeatureQuery element introduced in 8.2, we also introduce PropertyExclusion, which replaces PropertyName and is capable of handling dynamic feature properties as well.

Please note that the **GetFeature** operation does not provide any means to request a modified property or a modified list of properties according to criteria. Hence, there is support neither for the generation of SNAPHOTs nor the filtering of time slices. Only the use case “Retrieve the full history of a feature“ is supported.

### 7.1.2 Stored Queries

From the specification:

> A stored query expression is a persistent, parameterized, identifiable query expression. A stored query can be repeatedly invoked using its identifier with different values bound to its parameters each time.

Stored queries can use any language to implement their behavior. The given parameters can be of any type. Hence, stored queries naturally have the power to support all AIXM 5 use cases[7]. Those queries could be predefined[8] for the server and tailored to the required use cases. Table 6 shows that 3 different queries would be required. That said, what are the reasons that this paper does not simply define stored queries but instead promotes the introduction of new types for WFS 2.0 and FES 2.0? They are here:

- Demanding predefined stored queries is not very different from introducing a new query type: generic WFS 2.0 clients cannot support any of them because they cannot know about the semantics of the query parameters or elements.

- Using predefined stored queries would require the definition of the overall semantics involved in the execution of these queries. Basically, this would result in the definition of a new query language or in the extension of an already existing one. The latter is what this document does. By extending the wfs:Query and adding functionality tailored to the AIXM-TM, queries with Temporality Extension functionality can be executed in both an ad hoc and stored query.

- The intended use of stored queries is to parameterize complex queries to improve readability, speed up server processing (by query pre-compilation on server side) and reduce the amount of text in the request document. Typical parameters are concrete values or objects like numerical values in a certain unit of measurement, a spatial object for comparison, etc. For temporal queries, the parameters include complex

[7] The specification says ‘A stored query expression may be used in a [,,,] GetFeature […] operation to identify a set of features to be operated upon’ ([1], section 7.9.3.1). Strictly speaking, this implies that no transformation of features can take place and hence no time slice selection or snapshot generation is possible. However, we are not sure if this was the intention of the standard and assume for the following reasoning as if the result of a stored query can be an arbitrary set of features, new ones or modified ones.

[8] Predefined in the same manner as the required ‘GetFeatureById’ stored query.

filters, which are used to define the query itself, and because of which the query cannot be pre-compiled.

**Table 6 — Required Query Functionality**

| Description | Parameters | Return Value |
|---|---|---|
| Query SNAPSHOTs | - a list of feature types<br>- a feature level filter (optional)<br>- a time instant or time period<br>- a time slice level filter (optional) | Collection of features containing generated (and optionally filtered) snapshot time slices |
| Query time slices | - a list of feature types<br>- a feature level filter (optional)<br>- a time instant or time period (optional)[9]<br>- a time slice level filter (optional)<br>- a marker (flag) whether to ignore corrected time slices or not<br>- a marker (flag) whether to ignore canceled time slices or not[10] | Collection of features containing all or a subset of time slices |

### 7.2 GetPropertyValue

From the specification:

> The GetPropertyValue operation allows the value of a feature property or part of the value of a complex feature property to be retrieved from the data store for a set of features identified using a query expression.

In other words, **GetPropertyValue** is a combination of a **GetFeature** request and an XPath expression. A given filter is used to select features by criteria (this is the regular **GetFeature** operation), and in a subsequent process the XPath expression is used to extract XML elements out of the result.

One of the requirements is to generate and filter SNAPSHOT time slices. This requirement cannot be met by the **GetPropertyValue** operation, because the XPath expression is only capable of identifying a feature property (or any of its sub-properties)[11] (for the discussion to use stored queries for this, see above). Another central requirement

[9] Avoids a lengthy check on the valid time in the time slice filter.

[10] The two markers are technically time slice filters, but cannot be realized with the subset of XPath syntax defined for FES.

[11] Whereas in theory, custom XPath functions could be introduced to transform a list of time slices into SNAPSHOT time slices, this would be clearly against the semantics of the operation as defined by the WFS specification, which are to retrieve “the value of a feature property or part of a complex feature property”. Snapshots, on the other hand, are not part of the persistent set of time slices of a dynamic feature, they are generated (or derived).

is the capability to filter the list of time slice properties contained in the feature. This is possible with **GetPropertyValue**. However, significant disadvantages are observed:

- The filter criteria for time slices is likely to be quite complex. Spatial and temporal criteria are required for almost every common use case. The existing FES operators cannot be used for this purpose, because they are not available as XPath functions. Further on, custom XPath functions have to be introduced to make the exclusion of corrected and canceled time slices possible.

- The XPath expression has to be encoded in a single string. This string cannot be validated through an XML schema. The expression will be very hard to read for humans, especially if complex spatio-temporal functions are involved.

- The result of a **GetPropertyValue** operation is a collection of XML elements. In our case, these are timeSlice elements. A timeSlice element does not contain a reference to its parent feature. In addition, the GetPropertyValue operation response does not indicate which property values contained in the response belong to which features. Together this means that if property values from multiple features are contained in the result, the result is ambiguous.

### 7.3 Summary

This chapter showed that WFS 2.0 and FES 2.0 were designed to support static feature data, following the General Feature Model. They do not support the use cases for querying AIXM 5 data. Predefined stored queries can work around this limitation. However, in chapter 8 we will introduce extensions to FES 2.0 and WFS 2.0 as a general solution that works with ad hoc and stored queries.

## 8 Dynamic Feature Query

Following the requirements identified in Table 6 and in section 6.1, a new type of query is introduced here: a query for dynamic features. This query is tailored for retrieving data based on the AIXM 5 Temporality Model[12].

### 8.1 Overview

The following new elements are introduced:

- DynamicFeatureQuery (query expression)

[12] There was no experience with real-world domains applying the general GML dynamic feature model at the time of writing this document. However, AIXM-TM is general enough to form the base of any data model required to deal with temporality aspects, and not bound to the aeronautical world. It can become a domain independent concept, standardized by the relevant organizations. See also [OGC 11-093r2].

- DynamicFeatureFilter (selection clause)
- TimeSliceProjection (projection clause)
- SnapshotGeneration (projection clause)
- PropertyExclusion (projection clause)

The **DynamicFeatureQuery** is in the *substitution group* of the WFS *Query* type. Consequently, a dynamic feature query can be used in all query operations of a WFS: **GetFeature**, **GetPropertyValue**, **GetFeatureWithLock**, **LockFeature**, and also stored queries. A dynamic feature query shall target a dynamic feature type, for example an AIXM Airspace. The output of a dynamic feature query is a regular collection of dynamic features. The contained time slices, however,

- may represent a subset of the time slices of the dynamic feature or

- are generated in case of retrieving the complete state of the dynamic feature for a given point in time (snapshot query).

The contents depend on the projection clauses that are in place.

In the following, the new elements are explained in detail and their XML schema definition is presented. The complete schema can be found in Annex B, including the namespace bindings.

### 8.2 DynamicFeatureQuery

This is the basis for all dynamic feature queries at WFS. It is defined as follows:

```
<element name="DynamicFeatureQuery" type="wfs-
te:DynamicFeatureQueryType" substitutionGroup="wfs:Query">
</element>

<complexType name="DynamicFeatureQueryType">
   <complexContent>
      <extension base="wfs:QueryType"/>
   </complexContent>
</complexType>
```

The **DynamicFeatureQuery** extends from **wfs:QueryType** without any modification or extension. This is required to overcome and redefine constraints on applicable selection and projection clauses defined for **wfs:QueryType** in the underlying model (see OGC 09-025r1 figure 8).

The following child elements on **DynamicFeatureQuery** are allowed:

- Selection clauses: **fes-te:DynamicFeatureFilter** or **wfs:Filter**

- Projection clauses: **fes-te:TimeSliceProjection, fes-te:SnapshotGeneration, fes-te:PropertyExclusion**

**fes:Filter** may be used to select features without using time-based constraints, i.e. the state of a feature at a point or period in time is not needed. The WFS 2.0 specification fully applies. An example is feature selection using the **gml:identifier** (see A.1 for an example).

Please note that the **wfs:PropertyName** projection clause is not listed here and hence is not supported in a DynamicFeatureQuery. **fes-te:PropertyExclusion** replaces it, but with different overall semantics. Whereas in **wfs:Query** all optional properties are to be omitted by default, in a **wfs-te:DynamicFeatureQuery** all properties, mandatory and optional, are to be returned, except they are explicitly excluded by an **fes-te:PropertyExclusion** element.

### 8.3 DynamicFeatureFilter

The **DynamicFeatureFilter** is a selection clause capable of evaluating filter constraints for a given point in time or a time period. It consists of an optional **fes-te:timeIndicator** and a regular **fes:Filter**. It is defined as follows:

```
<element name="DynamicFeatureFilter" type="fes-
te:DynamicFeatureFilterType"
substitutionGroup="fes:AbstractSelectionClause">
</element>

<complexType name="DynamicFeatureFilterType">
  <complexContent>
    <extension base="fes:AbstractSelectionClauseType">
      <sequence>
        <element name="timeIndicator" minOccurs="0">
          <complexType>
            <sequence>
              <element ref="gml:AbstractTimeGeometricPrimitive"/>
            </sequence>
          </complexType>
        </element>
        <element name="featureFilter">
          <complexType>
            <sequence>
              <element ref="fes:Filter"/>
            </sequence>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

If a **timeIndicator** is present, its child element **gml:TimeInstant** or **gml:TimePeriod** defines the point in time or the time period, respectively, when the given **featureFilter** is to be evaluated. In other words, the state of the feature at the given point in time or time period is calculated according to the AIXM-TM and the filter constraints are applied on the result[13]. For **TimeInstant**s, properties with schedules are automatically evaluated before the filtering is done. For **TimePeriod**s, this is only supported if the according conformance class is supported (see 2.1).

If a **gml:TimePeriod** element is used as a **timeIndicator,** the filter is evaluated for all possible states of the feature, i.e. for all points in the feature lifetime that are not before and not after the given time period. In this case, multiple values per property are possible, for both ordinary properties and evaluated properties with schedules. These are handled in the usual manner for a collection of values, and the **matchAction** attribute applies (see [2], section 7.7.2.1.2). If the referenced property has a cardinality greater than one, i.e. is multi-occurring, technically a collection of collection of values is generated. In this case, all collections are merged into one in an arbitrary order[14].

[13] This equals an implicit application of the “evaluateDuring” function introduced in [4].

[14] For proper handling of collections of collections a new attribute such as ‘matchActionTime’ with similar semantics of matchAction would have to be introduced on every element that supports matchAction. However, in the sake of simplicity and the lack of use cases, this is not done.

Please note that there is no **matchAction** attribute for the operators **fes:PropertyIsLike** and **fes:PropertyIsNil**. For maximum expressiveness, this should be changed in a future version of WFS. Until this is the case, **matchAction** "any" shall apply here.

The reason for the **timeIndicator** being optional is that the valueFor() custom XPath function may act as replacement for it in a future version of the temporality extension (see 12.2). If no **timeIndicator** is present, the **DynamicFeatureFilter** behaves like a normal **fes:Filter**, where **timeSlice** properties are handled like ordinary first level properties.

Please note that the **featureFilter** is a filter on feature level. If no projection clause is in place, the complete feature including the full history of time slices is returned.

### 8.4 SnapshotGeneration

The **SnapshotGeneration** element is a projection clause that generates snapshot time slices according to AIXM-TM for the selected features. It is defined as follows:

```
<element name="SnapshotGeneration" type="fes-te:SnapshotGenerationType"
substitutionGroup="fes:AbstractProjectionClause">
</element>

<complexType name="SnapshotGenerationType">
  <complexContent>
    <extension base="fes:AbstractProjectionClauseType">
      <sequence>
        <element name="snapshotTime">
          <complexType>
            <sequence>
              <element ref="gml:AbstractTimeGeometricPrimitive"/>
            </sequence>
          </complexType>
        </element>
      </sequence>
      <attribute name="evaluateSchedule" type="boolean" use="optional"
default="false">
      </attribute>
    </extension>
  </complexContent>
</complexType>
```

If this projection clause is present in a request, time slices with SNAPSHOT interpretation are generated for the selected features for the given time. They replace the list (history) of time slices. If **snapshotTime** is a **gml:TimeInstant**, the **gml:validTime** properties of all snapshots are identical **gml:TimeInstant**s. If **snapshotTime** is a **gml:TimePeriod**, an extended snapshot definition is applied, as follows: if no property changes in the given time period, a single time slice is returned with the according

**gml:validTime** (time period). If there are property changes, the server shall return a list of consecutive time slices with disjoint **gml:validTime**s that span the given time period. If the requested time period lies partly outside the life time of the feature, the returned time period is adjusted accordingly. If the feature is not defined at the given time instant or time period, it is removed from the returned feature collection[15] – see [3 – OGC 11-093r2], section 10.1.3.4.1, for further details.

The optional attribute **evaluateSchedules** triggers the evaluation of properties with schedules for the given time. That means that the collection of properties is replaced by a single property that contains the values valid at the given time with the **timeInterval** property removed (see A.9 for an example). If no attribute is given, “false” as a default value applies.

Schedule evaluation for time periods requires the service to drill down into the details of each schedule. Consider the following figures.

[15] An obvious alternative, returning an „empty“ feature without time slices, unfortunately is not schema-valid

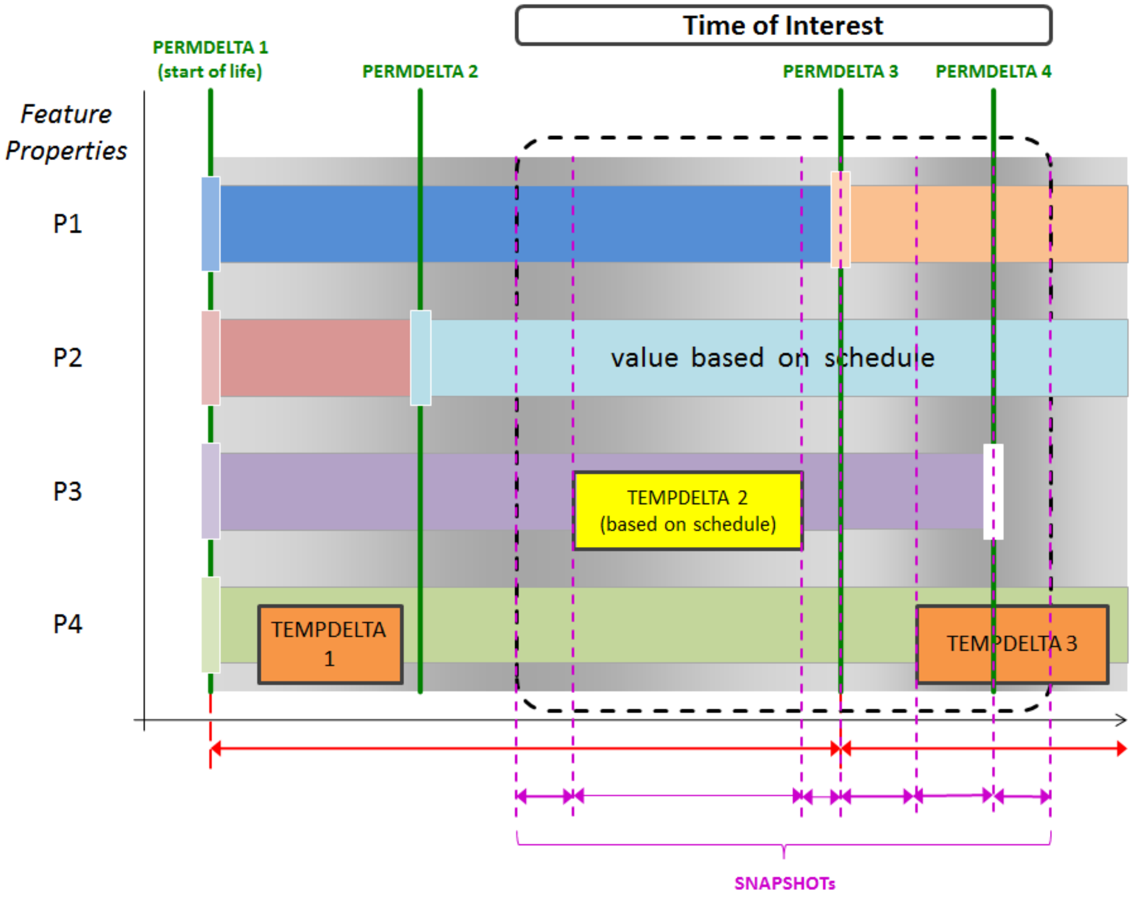


**Figure 4: Creating snapshots for a time period of interest without evaluation of schedules**

Figure 4 shows how a service would create snapshots of a dynamic feature for a time period of interest, without evaluation of schedules. Scheduled values are considered as ordinary, static properties with a fixed value throughout the valid time of the respective time slice. Six snapshots are created to cover the time of interest, based upon the following sequence of value changes:

1. TEMPDELTA 2 (P3, start of temporary change),
2. TEMPDELTA 2 (P3, end of temporary change),
3. PERMDELTA 3 (P1),
4. TEMPDELTA 3 (P4, start of temporary change),
5. PERMDELTA 4 (P3)

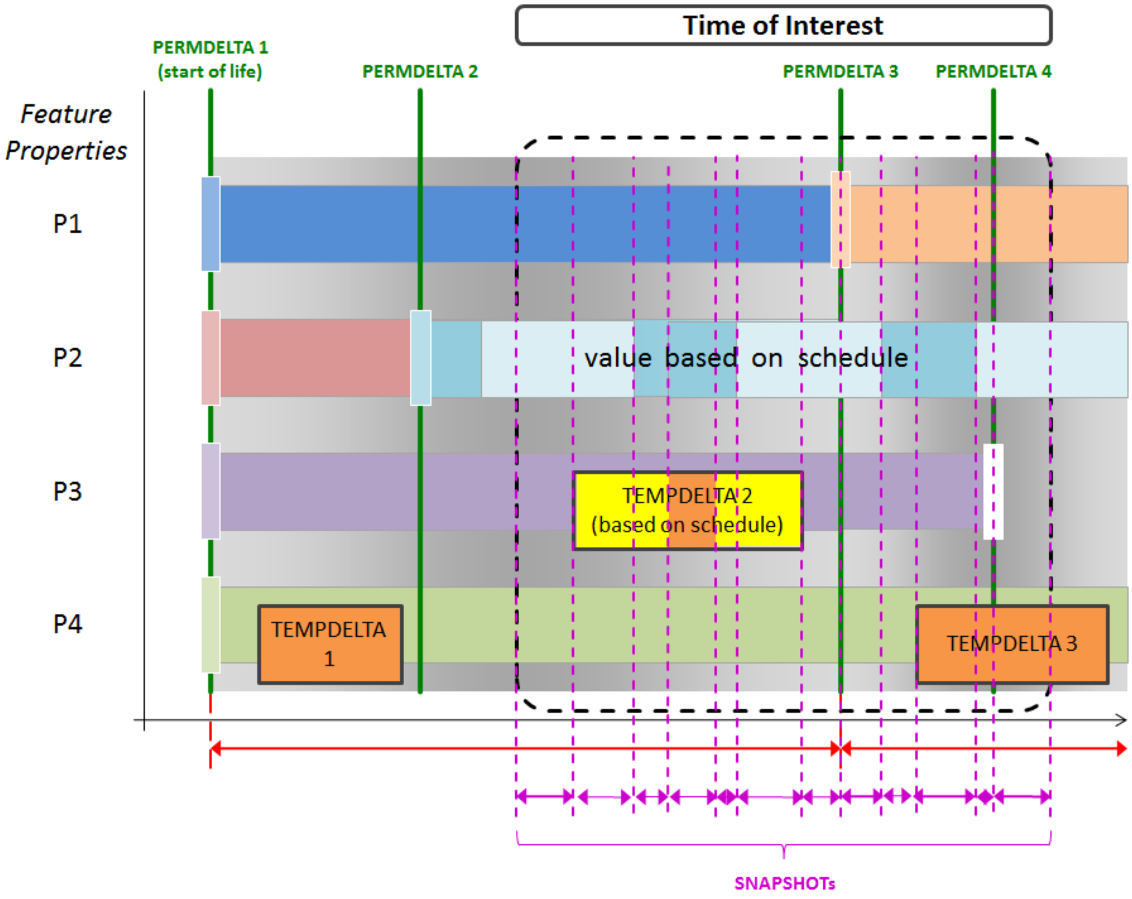


**Figure 5: Creating snapshots for a time period of interest with evaluation of schedules**

The creation of dynamic feature snapshots for a time period of interest with evaluation of schedules is shown in Figure 5. Here scheduled values are fully taken into account. Twelve snapshots are created to cover the time of interest, based upon the following sequence of value changes:

1. TEMPDELTA 2 (P3, start of temporary change),
2. Scheduled value change (P2),
3. Scheduled value change (P3),
4. Scheduled value change (P3),
5. Scheduled value change (P2),
6. TEMPDELTA 2 (P3, end of temporary change),
7. PERMDELTA 3 (P1),
8. Scheduled value change (P2),
9. TEMPDELTA 3 (P4),
10. Scheduled value change (P2),
11. PERMDELTA 4 (P3)

As schedule evaluation for time periods might result in a great number of time slices for frequently changing properties and long time periods, support for this is considered an advanced feature of the service. If the service does not implement the TimePeriodScheduleEvaluation capability[16], it shall report an exception in that case.

At most one **SnapshotGeneration** projection clause may be present on a **DynamicFeatureQuery** element. The **SnapshotGeneration** and **TimeSliceProjection** elements mutually exclude each other.

### 8.5 TimeSliceProjection

The **TimeSliceProjection** element is a projection clause that enables the client to filter the returned list of time slices using a **fes:Filter** definition for *relevant* time slices. It is defined as follows:

[16] The exact specification of this functionality is left for future work on this extension.

```
<element name="TimeSliceProjection" type="fes-
te:TimeSliceProjectionType"
substitutionGroup="fes:AbstractProjectionClause">
</element>

<complexType name="TimeSliceProjectionType">
  <complexContent>
    <extension base="fes:AbstractProjectionClauseType">
      <sequence>
        <element name="relevantTime" minOccurs="0">
          <complexType>
            <sequence>
              <element ref="gml:AbstractTimeGeometricPrimitive"/>
            </sequence>
          </complexType>
        </element>
        <element name="timeSliceFilter" minOccurs="0">
          <complexType>
            <sequence>
              <element ref="fes:Filter"/>
            </sequence>
          </complexType>
        </element>
      </sequence>
      <attribute name="includeCorrected" type="boolean" use="optional"
default="false">
      </attribute>
      <attribute name="includeCanceled" type="boolean" use="optional"
default="false">
      </attribute>
    </extension>
  </complexContent>
</complexType>
```

The context node for the **timeSliceFilter** element is the concrete time slice of the feature. This simplifies XPath expressions used in **fes:ValueReference** elements.

EXAMPLE This reduces an XPath of

```
aixm:timeSlice/*/aixm:someProperty
```

to

```
aixm:someProperty
```

The optional **relevantTime** element serves as a shortcut to an otherwise lengthy check on the **gml:validTime** of the time slices. It selects all time slices that are relevant for (not before and not after) a given time period or time instant, i.e. all time slices that define the state for that time.

By default, *canceled* and *corrected* time slices, as defined in 5.3, are not part of the returned set of time slices. This can be changed by setting the attributes **includeCanceled** and **includeCorrected** to true, respectively. In the special case of setting

**includeCorrected** to true, and setting **includeCanceled** to false, canceled time slices shall not be returned.

At most one **TimeSliceProjection** projection clause may be present on a **DynamicFeatureQuery** element. The **SnapshotGeneration** and **TimeSliceProjection** elements mutually exclude each other.

### 8.6 PropertyExclusion

This element serves as a replacement for the existing **wfs:PropertyName** projection clause, but has sort of an opposite meaning. Due to their time-variant nature, technically almost all properties of a dynamic feature following the AIXM-TM are optional from an XML schema point of view. Hence, an inclusion of optional properties as defined by the **wfs:PropertyName** projection clause is not sensible here. By definition, all available properties are returned by a **DynamicFeatureQuery** (see 8.2). The **PropertyExclusion** projection clause enables the client to exclude properties.

Each property of a dynamic feature – both on the feature and timeslice level – can be uniquely identified by its scoped name. In an XML encoding, this translates into a QName.

This projection clause is useful for use cases where only a subset of the properties of a feature (or time slice) need to be retrieved from a WFS. Additional information can always be skipped by the client, but it is generally more efficient not to receive it at all for saving bandwidth and processing time. This is especially true if the unwanted data is big in size compared to the data actually used, and if it is received very often. One example for this kind of data is metadata: there is time slice metadata, feature metadata and even message metadata defined in AIXM 5. For quite a lot of use cases, such as machine-to-machine communication in an operational scenario, metadata may not be processed at all.

XML schema fragment:

```
<element name="PropertyExclusion" type="fes-te:PropertyExclusionType"
substitutionGroup="fes:AbstractProjectionClause">
</element>

<complexType name="PropertyExclusionType">
  <complexContent>
    <extension base="fes:AbstractProjectionClauseType">
      <sequence>
        <element name="propertyName" type="QName"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Multiple **PropertyExclusion** elements may occur in a **DynamicFeatureQuery** to exclude multiple elements. See A.10 for an example.

It is the responsibility of the client not to exclude elements that render the resulting feature schema-invalid.

## 9 Indicating Support for Temporality Extension

To indicate the level of support for the Temporality Extension an OGC Web Service such as a WFS instance lists the supported Temporality Extension conformance classes in its Capabilities document. More precisely, it shall list the identifier of the supported Temporality Extension conformance classes as an ows:Profile value in the ServiceIdentification section of the Capabilities document – see following example.

**Listing 1: WFS Capabilities listing supported Temporality Extension conformance classes**

```
<WFS_Capabilities version="2.0.0" xmlns="http://www.opengis.net/wfs/2.0"
xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:fes="http://www.opengis.net/fes/2.0"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <ows:ServiceIdentification>
        <ows:Title>OGC Member WFS</ows:Title>
        <ows:Abstract>Web Feature Service</ows:Abstract>
        <ows:Keywords>
            <ows:Keyword>Aviation</ows:Keyword>
        </ows:Keywords>
        <ows:ServiceType>WFS</ows:ServiceType>
        <ows:ServiceTypeVersion>2.0.0</ows:ServiceTypeVersion>
        <ows:Profile>http://www.opengis.net/spec/WFS-TE/1.0/conf/Core</ows:Profile>
        <ows:Profile>http://www.opengis.net/spec/FES-
TE/1.0/conf/TimePeriodScheduleEvaluation</ows:Profile>
        <ows:Fees>NONE</ows:Fees>
        <ows:AccessConstraints>NONE</ows:AccessConstraints>
    </ows:ServiceIdentification>
</WFS_Capabilities>
```

Note that conformance classes that are implied via a direct or indirect dependency of listed conformance classes do not need to be listed in the Capabilities. For example, FES-TE-Core is not listed here because all of the other Temporality Extension conformance classes depend on it.

In this example, the service supports the minimum set of Temporality Extension functionality (as indicated by the WFS-TE-Core conformance class). In addition the FES-TE-TimePeriodScheduleEvaluation conformance class is supported. However, the FES-TE-PropertyExclusion is not supported.

## 10 Compatibility with Existing WFS 2.0 Based Systems

The proposed extension is established as a new XML schema which inherits from the existing WFS and FES 2.0 XML Schema. Thus, the existing WFS 2.0 standard is preserved and support for dynamic feature queries is optional and depends upon the implementation of the Temporality Extension conformance class(es).

To enable existing, non-temporality aware WFS 2.0 clients to query AIXM 5 data in a basic way, a proxy WFS could be installed which sits in-between the client and the temporality-enabled WFS and translates incoming requests to a snapshot query parameterized with the current time.

## 11 Alternative Approaches

Introducing a new query type is not the only way to get the required functionality of a temporality-aware query interface. In the process of finding the best solution, several approaches were evaluated. In chapter 7 the existing WFS 2.0 standard was evaluated in how far it is matching the requirements. In this chapter we evaluate alternative approaches of extending the standard or utilizing other existing standards. For each approach advantages and disadvantages over the proposed dynamic feature query are discussed.

### 11.1 Web Processing Services

The Web Processing Service (WPS) is an OGC standard for processing geospatial data. A WPS provides client access across a network to pre-programmed calculations and/or computation models that operate on spatially referenced data. Data inputs can be legitimate calls to OGC web services, such as WFS.

It is possible to implement a WPS that meets all of the identified requirements. The general workflow is depicted in Figure 6.

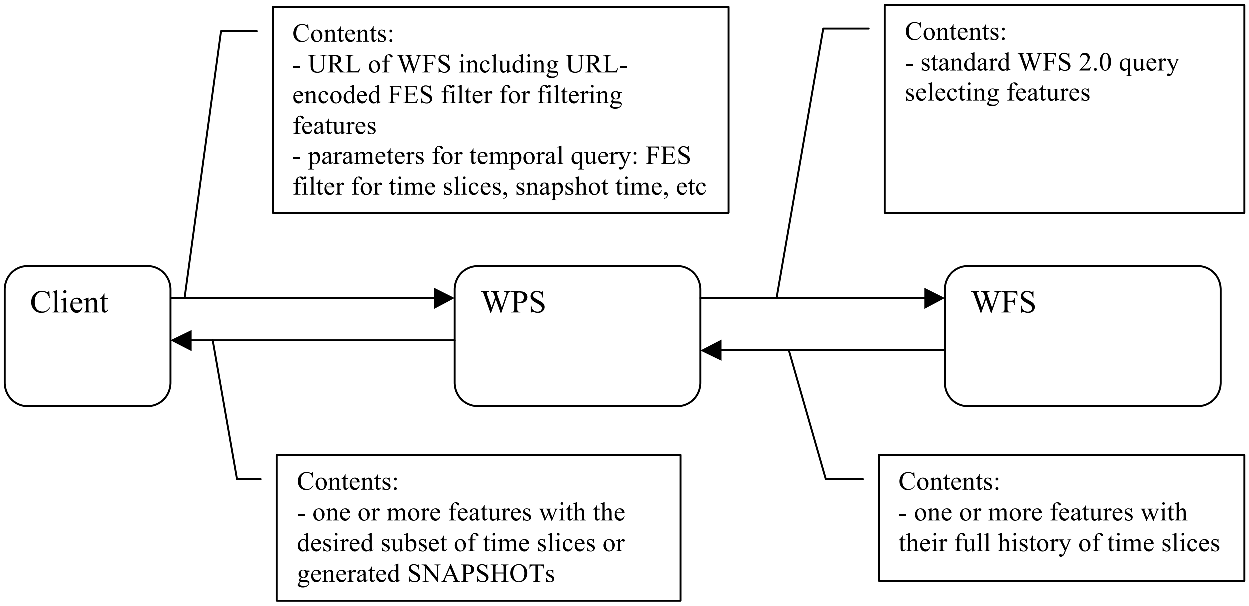

Contents:
- URL of WFS including URL-encoded FES filter for filtering features
- parameters for temporal query: FES filter for time slices, snapshot time, etc
Contents:
- standard WFS 2.0 query selecting features
Client
WPS
WFS
Contents:
- one or more features with the desired subset of time slices or generated SNAPSHOTs
Contents:
- one or more features with their full history of time slices

**Figure 6 - WPS/WFS Workflow for Temporal Queries**

Advantages

- Works with the existing WFS and WPS standards.

Disadvantages

- For each query, the full history of time slices of the selected features has to be transmitted between the WFS and the WPS. This is inefficient and will not be acceptable for productive, long-running systems.

- Still, though standard WFS and WPS are used, generic clients will not be able to make use of the temporal queries efficiently. In order to make use of and understand dynamic feature data clients still need to understand and support dynamic feature data, i.e. clients have to be temporality-aware.

Summary

The disadvantages clearly outweigh the advantages.

### 11.2 XQuery and XSLT

XQuery is a powerful query and functional language designed to query collections of XML data. XSLT is a transformation language designed for XML. Both are not OGC standards and do not define an interface for a web service. Hence, they cannot replace a WFS. However, the power of their query languages qualifies them for a possible extension of a WFS improving its filtering capabilities. For the sake of completeness we will briefly evaluate their application on AIXM 5 data.

XQuery and XSLT have the power to express all required queries including the generation of SNAPSHOT time slices. However,

- There are no spatio-temporal operators available. The powerful FES filter operators have to be redefined using custom XPath functions

- There is no feature model as they are based on pure XML. There is no abstraction of the temporality models used in AIXM 5 or GML DFM.

The result is that even simple queries dealing with temporality aspects will require complex query expressions that are hard to build and read.

## 12 Future Work

This discussion paper proposes an extension to the WFS 2.0 and FES 2.0 specification, and as such it is work in progress. Detailed reviews by stakeholders and testing are needed to bring it forward and finalize it. During that process, changes to individual elements are likely. For this reason some work items are not finished here by intention, others are features that were considered being too experimental or too advanced for the first version of this document.

### 12.1 Formal Work

Future work includes formal work in the following areas:

- the analysis of edge cases, error reporting, and robustness
- complete definition of how schedules are to be evaluated for output if requested for a time period (see 8.4)

### 12.2 Advanced Filtering: valueFor() XPath function

A custom XPath function for evaluating a property at a given time would further enhance the power of dynamic feature queries. It could be used in **fes:ValueReference** elements and would enable the client to combine constraints on time slice property values valid at different points in time. As such, it may replace the **timeIndicator** of the **DynamicFeatureFilter** (see 8.3). An example is given in A.2.

Related standards like the OGC Symbology Encoding may also profit by this function as it enables them to apply more powerful XPath expressions.

It can be defined as follows:

- Context node: AIXM 5 feature node containing the full history of time slices
- Parameters (Time Instant version):
  1. dateTime[17]
  2. XPath pointing to a time slice property (XPath subset as defined in [2])
- Parameters (Time Period version):
  1. dateTime (start time of the time period, inclusive[18])
  2. dateTime (end time of the time period, inclusive)
  3. XPath pointing to a time slice property (XPath subset as defined in [2])
- Return value: set of all nodes pointed to by the XPath expression and valid at the given time instant or period

Please note that the FES 2.0 specification puts constraints on the allowed XPath expressions of a **ValueReference**. Only a small subset of the XPath language is allowed. Hence, to be able to introduce the snapshot function there, the subset needs to be extended.The practical use of this function has further to be investigated before it can be included in this extension.

[17] XML Schema primitive datatype

[18] If needed, additional functions or parameters can be introduced to support open time intervals in a future version

# Annex A

# Examples

## A.1 Data Retrieval Use Case 1 (see 6.1)

Request:

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- DynamicFeatureQuery example for snapshot generation.
  Use case: Retrieve the complete state of a feature at a point in time
  Category: Visualisation, decision support
  Example: Retrieve the state of a runway at the time of arrival. -->

<wfs:GetFeature
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:wfs-te="http://www.opengis.net/wfs-te/1.0"
  xmlns:fes-te="http://www.opengis.net/fes-te/1.0"
  xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:aixm="http://www.aixm.aero/schema/5.1"
  service="WFS" version="2.0.0">
  <wfs-te:DynamicFeatureQuery typeNames="aixm:Runway">
    <fes-te:SnapshotGeneration>
      <fes-te:snapshotTime>
        <gml:TimeInstant gml:id="gmlID1">
          <gml:timePosition>2011-07-12T09:11:01.857Z</gml:timePosition>
        </gml:TimeInstant>
      </fes-te:snapshotTime>
    </fes-te:SnapshotGeneration>
    <fes:Filter>
      <fes:PropertyIsEqualTo>
        <fes:ValueReference>gml:identifier</fes:ValueReference>
        <fes:Literal>0083defb-b42e-4417-9be2-7aba2db2674d</fes:Literal>
      </fes:PropertyIsEqualTo>
    </fes:Filter>
  </wfs-te:DynamicFeatureQuery>
</wfs:GetFeature>
```

## A.2 Data Retrieval Use Case 2 (see 6.1)

Request using **timeIndicator**:

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- DynamicFeatureQuery example for snapshot generation with a time slice filter
  Use case: Retrieve all features fulfilling a certain criteria at a point in time
  Category: Decision support
  Example: Retrieve all airports in certain area that are operational at the time of
arrival
-->

<wfs:GetFeature xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:wfs="http://www.opengis.net/wfs/2.0"
```

```
  xmlns:wfs-te="http://www.opengis.net/wfs-te/1.0"
xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:fes-te="http://www.opengis.net/fes-te/1.0"
xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:aixm="http://www.aixm.aero/schema/5.1"
  service="WFS" version="2.0.0">
  <wfs-te:DynamicFeatureQuery typeNames="aixm:AirportHeliport">
    <fes-te:SnapshotGeneration>
      <fes-te:snapshotTime>
        <gml:TimeInstant gml:id="gmlID1">
          <gml:timePosition>2011-07-12T09:11:01.857Z</gml:timePosition>
        </gml:TimeInstant>
      </fes-te:snapshotTime>
    </fes-te:SnapshotGeneration>
    <fes-te:DynamicFeatureFilter>
      <fes-te:timeIndicator>
        <gml:TimeInstant gml:id="gmlID2">
          <gml:timePosition>2011-07-12T09:11:01.857Z</gml:timePosition>
        </gml:TimeInstant>
      </fes-te:timeIndicator>
      <fes-te:featureFilter>
        <fes:Filter>
          <fes:And>
            <fes:DWithin>
              <fes:ValueReference>aixm:timeSlice/*/aixm:ARP</fes:ValueReference>
              <gml:Point srsName="urn:ogc:def:crs:EPSG::4326" gml:id="gmlID3">
                <gml:pos>50.9 4.34</gml:pos>
              </gml:Point>
              <fes:Distance uom="M">100000</fes:Distance>
            </fes:DWithin>
            <fes:PropertyIsEqualTo>

<fes:ValueReference>aixm:timeSlice/*/aixm:availability/aixm:operationalStatus</fes:ValueR
eference>
              <!-- the schedule is automatically evaluated at the given time -->
              <fes:Literal>NORMAL</fes:Literal>
            </fes:PropertyIsEqualTo>
          </fes:And>
        </fes:Filter>
      </fes-te:featureFilter>
    </fes-te:DynamicFeatureFilter>
  </wfs-te:DynamicFeatureQuery>
</wfs:GetFeature>
```

### Alternative Request using the valueFor() XPath function (Future Work, see 12.2)

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- DynamicFeatureQuery example for snapshot generation with a time slice filter
  Use case: Retrieve all features fulfilling a certain criteria at a point in time
  Category: Decision support
  Example: Retrieve all airports in certain area that are operational at the time of
arrival
-->

<wfs:GetFeature xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:wfs-te="http://www.opengis.net/wfs-te/1.0"
xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:fes-te="http://www.opengis.net/fes-te/1.0"
xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:aixm="http://www.aixm.aero/schema/5.1"
  service="WFS" version="2.0.0">
  <wfs-te:DynamicFeatureQuery typeNames="aixm:AirportHeliport">
    <fes-te:SnapshotGeneration>
      <fes-te:snapshotTime>
        <gml:TimeInstant gml:id="gmlID1">
          <gml:timePosition>2011-07-12T09:11:01.857Z</gml:timePosition>
        </gml:TimeInstant>
      </fes-te:snapshotTime>
    </fes-te:SnapshotGeneration>
    <fes-te:DynamicFeatureFilter>
      <fes-te:featureFilter>
        <fes:Filter>
          <fes:And>
            <fes:DWithin>
              <fes:ValueReference>aixm:valueFor('2011-07-
12T09:11:01.857Z','aixm:ARP')</fes:ValueReference>
              <gml:Point srsName="urn:ogc:def:crs:EPSG::4326" gml:id="gmlID3">
                <gml:pos>50.9 4.34</gml:pos>
              </gml:Point>
              <fes:Distance uom="M">100000</fes:Distance>
            </fes:DWithin>
            <fes:PropertyIsEqualTo>
              <fes:ValueReference>aixm:valueFor('2011-07-
12T09:11:01.857Z','aixm:availability/aixm:operationalStatus')</fes:ValueReference>
              <!-- the schedule is automatically evaluated at the given time -->
              <fes:Literal>NORMAL</fes:Literal>
            </fes:PropertyIsEqualTo>
          </fes:And>
        </fes:Filter>
      </fes-te:featureFilter>
    </fes-te:DynamicFeatureFilter>
  </wfs-te:DynamicFeatureQuery>
</wfs:GetFeature>
```

## A.3 Data Retrieval Use Case 3 (see 6.1)

Request:

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- DynamicFeatureQuery example for retrieving relevant time slices
  Use case: Retrieve all time slices of a feature relevant for (i.e. affecting the state
at) a point in time
  Category: Change-aware visualisation
  Example: Retrieve all time slices relevant for the time of arrival of a runway.
-->

<wfs:GetFeature
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:wfs-te="http://www.opengis.net/wfs-te/1.0"
  xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:fes-te="http://www.opengis.net/fes-te/1.0"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:aixm="http://www.aixm.aero/schema/5.1"
  service="WFS" version="2.0.0">
  <wfs-te:DynamicFeatureQuery typeNames="aixm:Runway">
    <fes-te:TimeSliceProjection>
      <fes-te:relevantTime>
        <gml:TimeInstant gml:id="gmlID1">
          <gml:timePosition>2011-07-12T09:11:01.857Z</gml:timePosition>
        </gml:TimeInstant>
      </fes-te:relevantTime>
    </fes-te:TimeSliceProjection>
    <fes:Filter>
      <fes:PropertyIsEqualTo>
        <fes:ValueReference>gml:identifier</fes:ValueReference>
        <fes:Literal>0083defb-b42e-4417-9be2-7aba2db2674d</fes:Literal>
      </fes:PropertyIsEqualTo>
    </fes:Filter>
  </wfs-te:DynamicFeatureQuery>
</wfs:GetFeature>
```

## A.4 Data Retrieval Use Case 4 (see 6.1)

Request:

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- DynamicFeatureQuery example for retrieving BASELINE time slices
  Use case: Retrieve the BASELINE of a feature valid at a point in time
  Category: AIP publication
  Example: Retrieve the BASELINE of an airport at an AIRAC effective date.
-->

<wfs:GetFeature xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:wfs-te="http://www.opengis.net/wfs-te/1.0"
xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:fes-te="http://www.opengis.net/fes-te/1.0"
xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:aixm="http://www.aixm.aero/schema/5.1"
  service="WFS" version="2.0.0">
  <wfs-te:DynamicFeatureQuery typeNames="aixm:AirportHeliport">
    <fes-te:TimeSliceProjection>
      <fes-te:relevantTime>
        <gml:TimeInstant gml:id="gmlID1">
          <gml:timePosition>2010-04-08T00:00:00.000Z</gml:timePosition>
        </gml:TimeInstant>
      </fes-te:relevantTime>
      <fes-te:timeSliceFilter>
        <fes:Filter>
          <fes:PropertyIsEqualTo>
            <fes:ValueReference>aixm:interpretation</fes:ValueReference>
            <fes:Literal>BASELINE</fes:Literal>
          </fes:PropertyIsEqualTo>
        </fes:Filter>
      </fes-te:timeSliceFilter>
    </fes-te:TimeSliceProjection>
    <fes:Filter>
      <fes:PropertyIsEqualTo>
        <fes:ValueReference>gml:identifier</fes:ValueReference>
        <fes:Literal>0083defb-b42e-4417-9be2-7aba2db2674d</fes:Literal>
      </fes:PropertyIsEqualTo>
    </fes:Filter>
  </wfs-te:DynamicFeatureQuery>
</wfs:GetFeature>
```

## A.5 Data Retrieval Use Case 5 (see 6.1)

Request:

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- DynamicFeatureQuery example for retrieving and filtering TEMPDELTA time slice
  Use case: Retrieve all TEMPDELTAs of a feature fully or partly valid for a time period
and matching a certain criteria
  Category: NOTAM communication
  Example: Retrieve all TEMPDELTAs affecting the operational status of a navigation aid
until the time of arrival.
-->

<wfs:GetFeature xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:wfs-te="http://www.opengis.net/wfs-te/1.0"
xmlns:fes="http://www.opengis.net/fes/2.0"
```

```
  xmlns:fes-te="http://www.opengis.net/fes-te/1.0"
xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:aixm="http://www.aixm.aero/schema/5.1"
  service="WFS" version="2.0.0">
  <wfs-te:DynamicFeatureQuery typeNames="aixm:Navaid">
    <fes-te:TimeSliceProjection>
      <fes-te:timeSliceFilter>
        <fes:Filter>
          <fes:And>
            <fes:PropertyIsEqualTo>
              <fes:ValueReference>aixm:interpretation</fes:ValueReference>
              <fes:Literal>TEMPDELTA</fes:Literal>
            </fes:PropertyIsEqualTo>
            <fes:AnyInteracts>
              <fes:ValueReference>gml:validTime</fes:ValueReference>
              <gml:TimePeriod gml:id="gmlID1">
                <gml:beginPosition>2010-04-08T00:00:00.000Z</gml:beginPosition>
                <gml:endPosition>2010-05-08T00:00:00.000Z</gml:endPosition>
              </gml:TimePeriod>
            </fes:AnyInteracts>
            <fes:Not>
              <fes:PropertyIsNull>
                <fes:ValueReference>aixm:availability</fes:ValueReference>
              </fes:PropertyIsNull>
            </fes:Not>
          </fes:And>
        </fes:Filter>
      </fes-te:timeSliceFilter>
    </fes-te:TimeSliceProjection>
    <fes-te:DynamicFeatureFilter>
      <fes-te:featureFilter>
        <fes:Filter>
          <fes:PropertyIsEqualTo>
            <fes:ValueReference>gml:identifier</fes:ValueReference>
            <fes:Literal>0083defb-b42e-4417-9be2-7aba2db2674d</fes:Literal>
          </fes:PropertyIsEqualTo>
        </fes:Filter>
      </fes-te:featureFilter>
    </fes-te:DynamicFeatureFilter>
  </wfs-te:DynamicFeatureQuery>
</wfs:GetFeature>
```

## A.6 Data Retrieval Use Case 6 (see 6.1)

Request:

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- DynamicFeatureQuery example for retrieving and filtering PERMDELTA time slices
  Use case: Retrieve all PERMDELTAs of a feature valid for a time period
  Category: Replication
  Example: Retrieve all PERMDELTAs of a feature valid now or in the future.
-->

<wfs:GetFeature xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:wfs-te="http://www.opengis.net/wfs-te/1.0"
xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:fes-te="http://www.opengis.net/fes-te/1.0"
xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:aixm="http://www.aixm.aero/schema/5.1"
  service="WFS" version="2.0.0">
  <wfs-te:DynamicFeatureQuery typeNames="aixm:Navaid">
    <fes-te:TimeSliceProjection>
      <fes-te:timeSliceFilter>
```

```
        <fes:Filter>
          <fes:And>
            <fes:PropertyIsEqualTo>
              <fes:ValueReference>aixm:interpretation</fes:ValueReference>
              <fes:Literal>PERMDELTA</fes:Literal>
            </fes:PropertyIsEqualTo>
            <fes:AnyInteracts>
              <fes:ValueReference>gml:validTime</fes:ValueReference>
              <gml:TimePeriod gml:id="gmlID1">
                <gml:beginPosition>2010-04-08T00:00:00.000Z</gml:beginPosition>
                <gml:endPosition/>
              </gml:TimePeriod>
            </fes:AnyInteracts>
          </fes:And>
        </fes:Filter>
      </fes-te:timeSliceFilter>
    </fes-te:TimeSliceProjection>
    <fes-te:DynamicFeatureFilter>
      <fes-te:featureFilter>
        <fes:Filter>
          <fes:PropertyIsEqualTo>
            <fes:ValueReference>gml:identifier</fes:ValueReference>
            <fes:Literal>0083defb-b42e-4417-9be2-7aba2db2674d</fes:Literal>
          </fes:PropertyIsEqualTo>
        </fes:Filter>
      </fes-te:featureFilter>
    </fes-te:DynamicFeatureFilter>
  </wfs-te:DynamicFeatureQuery>
</wfs:GetFeature>
```

## A.7 Data Retrieval Use Case 7 (see 6.1)

Request:

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- DynamicFeatureQuery example for retrieving and filtering PERMDELTA time slice
  Use case: Retrieve specific time slices of a feature by their unique id
  Category: Replication, technical verification
  Example: Retrieve the PERMDELTA time slice of a feature given the sequence number and
correction number
-->

<wfs:GetFeature xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:wfs-te="http://www.opengis.net/wfs-te/1.0"
xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:fes-te="http://www.opengis.net/fes-te/1.0"
xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:aixm="http://www.aixm.aero/schema/5.1"
  service="WFS" version="2.0.0">
  <wfs-te:DynamicFeatureQuery typeNames="aixm:Navaid">
    <fes-te:TimeSliceProjection includeCanceled="true" includeCorrected="true">
      <fes-te:timeSliceFilter>
        <fes:Filter>
          <fes:And>
            <fes:PropertyIsEqualTo>
              <fes:ValueReference>aixm:interpretation</fes:ValueReference>
              <fes:Literal>PERMDELTA</fes:Literal>
            </fes:PropertyIsEqualTo>
            <fes:PropertyIsEqualTo>
              <fes:ValueReference>aixm:sequenceNumber</fes:ValueReference>
              <fes:Literal>2</fes:Literal>
            </fes:PropertyIsEqualTo>
            <fes:PropertyIsEqualTo>
              <fes:ValueReference>aixm:correctionNumber</fes:ValueReference>
              <fes:Literal>1</fes:Literal>
            </fes:PropertyIsEqualTo>
          </fes:And>
        </fes:Filter>
      </fes-te:timeSliceFilter>
    </fes-te:TimeSliceProjection>
    <fes-te:DynamicFeatureFilter>
      <fes-te:featureFilter>
        <fes:Filter>
          <fes:PropertyIsEqualTo>
            <fes:ValueReference>gml:identifier</fes:ValueReference>
            <fes:Literal>0083defb-b42e-4417-9be2-7aba2db2674d</fes:Literal>
          </fes:PropertyIsEqualTo>
        </fes:Filter>
      </fes-te:featureFilter>
    </fes-te:DynamicFeatureFilter>
  </wfs-te:DynamicFeatureQuery>
</wfs:GetFeature>
```

## A.8 Data Retrieval Use Case 8 (see 6.1)

Request:

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Example for retrieving all time slices of a feature (standard WFS 2.0 query, no
DynamicFeatureQuery necessary)
  Use case: Retrieve the full history of a feature
  Category: Backup
  Example: Retrieve all time slices of a feature (BASELINEs, PERMDELTAs and TEMPDELTAs)
valid at any point in time.
-->

<wfs:GetFeature
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:aixm="http://www.aixm.aero/schema/5.1"
  service="WFS" version="2.0.0">
  <wfs:Query typeNames="aixm:Navaid">
    <fes:Filter>
      <fes:PropertyIsEqualTo>
        <fes:ValueReference>gml:identifier</fes:ValueReference>
        <fes:Literal>0083defb-b42e-4417-9be2-7aba2db2674d</fes:Literal>
      </fes:PropertyIsEqualTo>
    </fes:Filter>
  </wfs:Query>
</wfs:GetFeature>
```

## A.9 Evaluation of Properties with Schedule  (see 8.4)

Data store contents:

```
<aixm:Airspace xmlns:aixm="http://www.aixm.aero/schema/5.1"
xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:xlink="http://www.w3.org/1999/xlink"
gml:id="tmp">
  <gml:identifier codeSpace="http://www.example.org">0083defb-b42e-4417-9be2-
7aba2db2674d</gml:identifier>
  <aixm:timeSlice>
    <aixm:AirspaceTimeSlice gml:id="Airspace1_TD2">
```

```
      <gml:validTime>
        <gml:TimePeriod gml:id="Airspace1_TD2_TP">
          <gml:beginPosition>2011-07-01T00:00:00.000Z</gml:beginPosition>
          <gml:endPosition>2011-07-15T23:59:59Z</gml:endPosition>
        </gml:TimePeriod>
      </gml:validTime>
      <aixm:interpretation>BASELINE</aixm:interpretation>
      <aixm:sequenceNumber>1</aixm:sequenceNumber>
      <aixm:activation>
        <aixm:AirspaceActivation gml:id="Airspace1_TD2_AACT1">
          <aixm:timeInterval>
            <aixm:Timesheet gml:id="Airspace1_TD2_TS1">
              <aixm:timeReference>UTC+2</aixm:timeReference>
              <aixm:day>ANY</aixm:day>
              <aixm:startTime>00:00</aixm:startTime>
              <aixm:endTime>05:59</aixm:endTime>
            </aixm:Timesheet>
          </aixm:timeInterval>
          <aixm:activity>MILOPS</aixm:activity>
          <aixm:status>INACTIVE</aixm:status>
          <!-- other AirspaceActivation properties omitted for brevity -->
        </aixm:AirspaceActivation>
      </aixm:activation>
      <aixm:activation>
        <aixm:AirspaceActivation gml:id="Airspace1_TD2_AACT2">
          <aixm:timeInterval>
            <aixm:Timesheet gml:id="Airspace1_TD2_TS2">
              <aixm:timeReference>UTC+2</aixm:timeReference>
              <aixm:day>ANY</aixm:day>
              <aixm:startTime>06:00</aixm:startTime>
              <aixm:endTime>17:59</aixm:endTime>
            </aixm:Timesheet>
          </aixm:timeInterval>
          <aixm:activity>MILOPS</aixm:activity>
          <aixm:status>ACTIVE</aixm:status>
          <!-- other AirspaceActivation properties omitted for brevity -->
        </aixm:AirspaceActivation>
      </aixm:activation>
      <aixm:activation>
        <aixm:AirspaceActivation gml:id="Airspace1_TD2_AACT3">
          <aixm:timeInterval>
            <aixm:Timesheet gml:id="Airspace1_TD2_TS3">
              <aixm:timeReference>UTC+2</aixm:timeReference>
              <aixm:day>ANY</aixm:day>
              <aixm:startTime>18:00</aixm:startTime>
              <aixm:endTime>23:59</aixm:endTime>
            </aixm:Timesheet>
          </aixm:timeInterval>
          <aixm:activity>MILOPS</aixm:activity>
          <aixm:status>INACTIVE</aixm:status>
          <!-- other AirspaceActivation properties omitted for brevity -->
        </aixm:AirspaceActivation>
      </aixm:activation>
    </aixm:AirspaceTimeSlice>
  </aixm:timeSlice>
</aixm:Airspace>
```

Request:

```
<wfs:GetFeature xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:wfs-te="http://www.opengis.net/wfs-te/1.0"
xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:fes-te="http://www.opengis.net/fes-te/1.0"
xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:aixm="http://www.aixm.aero/schema/5.1"
  service="WFS" version="2.0.0">
  <wfs-te:DynamicFeatureQuery typeNames="aixm:Airspace">
    <fes-te:SnapshotGeneration>
      <fes-te:snapshotTime>
        <gml:TimeInstant gml:id="gmlID1">
          <gml:timePosition>2011-07-01T08:00:00.000Z</gml:timePosition>
        </gml:TimeInstant>
      </fes-te:snapshotTime>
    </fes-te:SnapshotGeneration>
    <fes-te:DynamicFeatureFilter>
      <fes-te:featureFilter>
        <fes:Filter>
          <fes:PropertyIsEqualTo>
            <fes:ValueReference>gml:identifier</fes:ValueReference>
            <fes:Literal>0083defb-b42e-4417-9be2-7aba2db2674d</fes:Literal>
          </fes:PropertyIsEqualTo>
        </fes:Filter>
      </fes-te:featureFilter>
    </fes-te:DynamicFeatureFilter>
  </wfs-te:DynamicFeatureQuery>
</wfs:GetFeature>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:FeatureCollection
  timeStamp="2008-09-07T19:00:00" numberReturned="1" numberMatched="1"
  xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <wfs:member>
    <aixm:Airspace
      xmlns:aixm="http://www.aixm.aero/schema/5.1"
      xmlns:gml="http://www.opengis.net/gml/3.2"
      gml:id="id1">
      <gml:identifier codeSpace="http://www.example.org">0083defb-b42e-4417-9be2-
7aba2db2674d</gml:identifier>
      <aixm:timeSlice>
        <aixm:AirspaceTimeSlice gml:id="Airspace1_TD2">
          <gml:validTime>
            <gml:TimeInstant gml:id="i1">
              <gml:timePosition>2011-07-01T08:00:00.000Z</gml:timePosition>
            </gml:TimeInstant>
          </gml:validTime>
          <aixm:interpretation>SNAPSHOT</aixm:interpretation>
          <aixm:activation>
            <aixm:AirspaceActivation gml:id="Airspace1_TD2_AACT2">
              <!-- no aixm:timeInterval -->
              <aixm:activity>MILOPS</aixm:activity>
              <aixm:status>ACTIVE</aixm:status>
            </aixm:AirspaceActivation>
          </aixm:activation>
        </aixm:AirspaceTimeSlice>
      </aixm:timeSlice>
    </aixm:Airspace>
  </wfs:member>
</wfs:FeatureCollection>
```

## A.10 Property exclusion (see 8.6)

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!-- DynamicFeatureQuery example for property exclusion.
  Use case: Retrieve the complete state of a feature at a point in time but exclude
metadata to save bandwidth
  Category: Visualisation, decision support
  Example: Retrieve the state of a runway at the time of arrival without metadata. -->

<wfs:GetFeature xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:wfs-te="http://www.opengis.net/wfs-te/1.0" xmlns:fes-
te="http://www.opengis.net/fes-te/1.0"
  xmlns:fes="http://www.opengis.net/fes/2.0" xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:aixm="http://www.aixm.aero/schema/5.1"
  service="WFS" version="2.0.0">
  <wfs-te:DynamicFeatureQuery typeNames="aixm:Runway">
    <fes-te:SnapshotGeneration>
      <fes-te:snapshotTime>
        <gml:TimeInstant gml:id="gmlID1">
          <gml:timePosition>2011-07-12T09:11:01.857Z</gml:timePosition>
        </gml:TimeInstant>
      </fes-te:snapshotTime>
    </fes-te:SnapshotGeneration>
    <fes-te:PropertyExclusion>
      <fes-te:propertyName>aixm:featureMetadata</fes-te:propertyName>
    </fes-te:PropertyExclusion>
    <fes-te:PropertyExclusion>
      <fes-te:propertyName>aixm:timeSliceMetadata</fes-te:propertyName>
    </fes-te:PropertyExclusion>
    <fes:Filter>
      <fes:PropertyIsEqualTo>
        <fes:ValueReference>gml:identifier</fes:ValueReference>
        <fes:Literal>0083defb-b42e-4417-9be2-7aba2db2674d</fes:Literal>
      </fes:PropertyIsEqualTo>
    </fes:Filter>
  </wfs-te:DynamicFeatureQuery>
</wfs:GetFeature>
```

# Annex B

# XML Schema

## FES Temporality Extension Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:fes-te="http://www.opengis.net/fes-te/1.0"
targetNamespace="http://www.opengis.net/fes-te/1.0" elementFormDefault="qualified"
  attributeFormDefault="unqualified" version="1.0">
  <import namespace="http://www.opengis.net/gml/3.2"
schemaLocation="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
  <import namespace="http://www.opengis.net/fes/2.0"
schemaLocation="http://schemas.opengis.net/filter/2.0/filterAll.xsd"/>
  <!-- ================================================= -->
  <element name="DynamicFeatureFilter" type="fes-te:DynamicFeatureFilterType"
substitutionGroup="fes:AbstractSelectionClause">
    <annotation>
      <documentation>For the selection of dynamic features that follow the Temporality
Model.</documentation>
    </annotation>
  </element>
  <complexType name="DynamicFeatureFilterType">
    <complexContent>
      <extension base="fes:AbstractSelectionClauseType">
        <sequence>
          <element name="timeIndicator" minOccurs="0">
            <annotation>
              <documentation>Defines the time instant or period for which the service
needs to evaluate the value(s) of dynamic feature properties that are identified in value
references of filter operators contained in the featureFilter. Note: the time given in
getValues XPath function calls included in value references of filter operators in the
featureFilter override the timeIndicator value for that filter operator.</documentation>
            </annotation>
            <complexType>
              <sequence>
                <element ref="gml:AbstractTimeGeometricPrimitive"/>
              </sequence>
            </complexType>
          </element>
          <element name="featureFilter">
            <annotation>
              <documentation>Defines criteria for a dynamic feature to be selected.
Filter context is the feature root element. Default matchAction for all filter operators
(except logical ones) is: Any (TBD).</documentation>
            </annotation>
            <complexType>
              <sequence>
                <element ref="fes:Filter"/>
              </sequence>
            </complexType>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <complexType name="DynamicFeatureFilterPropertyType">
    <sequence minOccurs="0">
```

```
      <element ref="fes-te:DynamicFeatureFilter"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <!-- ================================================= -->
  <element name="PropertyExclusion" type="fes-te:PropertyExclusionType"
substitutionGroup="fes:AbstractProjectionClause">
    <annotation>
      <documentation>For the more fine grained projection of dynamic feature properties,
including properties of timeSlices.</documentation>
    </annotation>
  </element>
  <complexType name="PropertyExclusionType">
    <complexContent>
      <extension base="fes:AbstractProjectionClauseType">
        <sequence>
          <element name="propertyName" type="QName"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <complexType name="PropertyExclusionPropertyType">
    <sequence minOccurs="0">
      <element ref="fes-te:PropertyExclusion"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <!-- ================================================= -->
  <element name="SnapshotGeneration" type="fes-te:SnapshotGenerationType"
substitutionGroup="fes:AbstractProjectionClause">
    <annotation>
      <documentation>For the generation of snapshots of the dynamic features contained in
the selection.</documentation>
    </annotation>
  </element>
  <complexType name="SnapshotGenerationType">
    <complexContent>
      <extension base="fes:AbstractProjectionClauseType">
        <sequence>
          <element name="snapshotTime">
            <annotation>
              <documentation>Defines the point in time or time interval for which
snapshot(s) are created.</documentation>
            </annotation>
            <complexType>
              <sequence>
                <element ref="gml:AbstractTimeGeometricPrimitive"/>
              </sequence>
            </complexType>
          </element>
        </sequence>
        <attribute name="evaluateSchedule" type="boolean" use="optional" default="false">
          <annotation>
            <documentation>Defines whether properties with schedule are processed to only
contain the value that applies at the snapshot valid time instance (true, default option)
or whether the schedule information is kept as is (false).</documentation>
          </annotation>
        </attribute>
      </extension>
    </complexContent>
  </complexType>
  <complexType name="SnapshotGenerationPropertyType">
    <sequence minOccurs="0">
      <element ref="fes-te:SnapshotGeneration"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <!-- ================================================= -->
  <element name="TimeSliceProjection" type="fes-te:TimeSliceProjectionType"
substitutionGroup="fes:AbstractProjectionClause">
    <annotation>
      <documentation>For the projection of timeSlices of the dynamic features contained
in the selection. Note: if a feature does not have any timeSlice associated with it after
the projection, it shall be omitted in the result.</documentation>
    </annotation>
  </element>
  <complexType name="TimeSliceProjectionType">
    <complexContent>
      <extension base="fes:AbstractProjectionClauseType">
        <sequence>
          <element name="relevantTime" minOccurs="0">
            <annotation>
              <documentation>Project selection to timeSlices with validTime not being
before and not after given relevant time. In addition, all correction and cancelation
timeSlices for them shall be included in the projection set as well.</documentation>
            </annotation>
            <complexType>
              <sequence>
                <element ref="gml:AbstractTimeGeometricPrimitive"/>
              </sequence>
            </complexType>
          </element>
          <element name="timeSliceFilter" minOccurs="0">
            <annotation>
              <documentation>Project selection to timeSlices that satisfy the filter
criteria. Filter context is the timeSlice root element.</documentation>
            </annotation>
            <complexType>
              <sequence>
                <element ref="fes:Filter"/>
              </sequence>
            </complexType>
          </element>
        </sequence>
        <attribute name="includeCorrected" type="boolean" use="optional" default="false">
          <annotation>
            <documentation>True if corrected timeSlices of a selected feature are to be
included in the result, else false (default).</documentation>
          </annotation>
        </attribute>
        <attribute name="includeCanceled" type="boolean" use="optional" default="false">
          <annotation>
            <documentation>True if cancelation and canceled timeSlices of a selected
feature are to be included in the result, else false (default).</documentation>
          </annotation>
        </attribute>
      </extension>
    </complexContent>
  </complexType>
  <complexType name="TimeSliceProjectionPropertyType">
    <sequence minOccurs="0">
      <element ref="fes-te:TimeSliceProjection"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
</schema>
```

## WFS Temporality Extension Schema

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:wfs-te="http://www.opengis.net/wfs-te/1.0"
xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:fes-te="http://www.opengis.net/fes-te/1.0"
targetNamespace="http://www.opengis.net/wfs-te/1.0" elementFormDefault="qualified"
  attributeFormDefault="unqualified" version="1.0">
  <import namespace="http://www.opengis.net/gml/3.2"
schemaLocation="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
  <import namespace="http://www.opengis.net/wfs/2.0"
schemaLocation="http://schemas.opengis.net/wfs/2.0/wfs.xsd"/>
  <import namespace="http://www.opengis.net/fes-te/1.0" schemaLocation="../fes-
te/fesTemporality.xsd"/>
  <!-- ================================================= -->
  <element name="DynamicFeatureQuery" type="wfs-te:DynamicFeatureQueryType"
substitutionGroup="wfs:Query">
    <annotation>
      <documentation>WFS query that supports dynamic feature selection and
projection.</documentation>
    </annotation>
  </element>
  <complexType name="DynamicFeatureQueryType">
    <complexContent>
      <extension base="wfs:QueryType"/>
    </complexContent>
  </complexType>
  <complexType name="DynamicFeatureQueryPropertyType">
    <sequence minOccurs="0">
      <element ref="wfs-te:DynamicFeatureQuery"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
</schema>
```

# Annex C

# UML Models

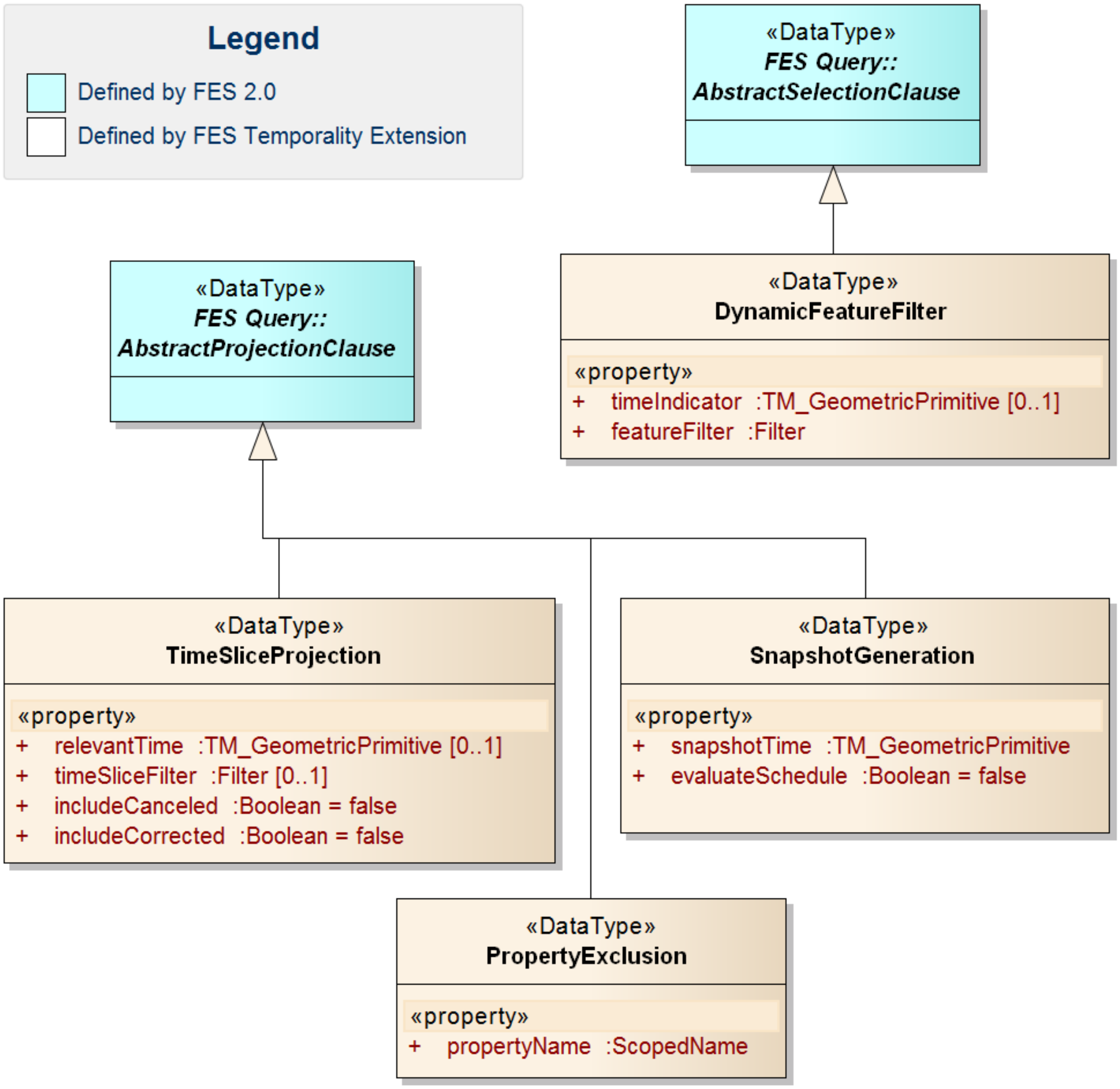

Legend
Defined by FES 2.0
Defined by FES Temporality Extension
«DataType»
FES Query::
AbstractSelectionClause
«DataType»
DynamicFeatureFilter
«property»
+ timeIndicator :TM_GeometricPrimitive [0..1]
+ featureFilter :Filter
«DataType»
FES Query::
AbstractProjectionClause
«DataType»
TimeSliceProjection
«property»
+ relevantTime :TM_GeometricPrimitive [0..1]
+ timeSliceFilter :Filter [0..1]
+ includeCanceled :Boolean = false
+ includeCorrected :Boolean = false
«DataType»
SnapshotGeneration
«property»
+ snapshotTime :TM_GeometricPrimitive
+ evaluateSchedule :Boolean = false
«DataType»
PropertyExclusion
«property»
+ propertyName :ScopedName

**Figure 7: FES Temporality Extension Model**

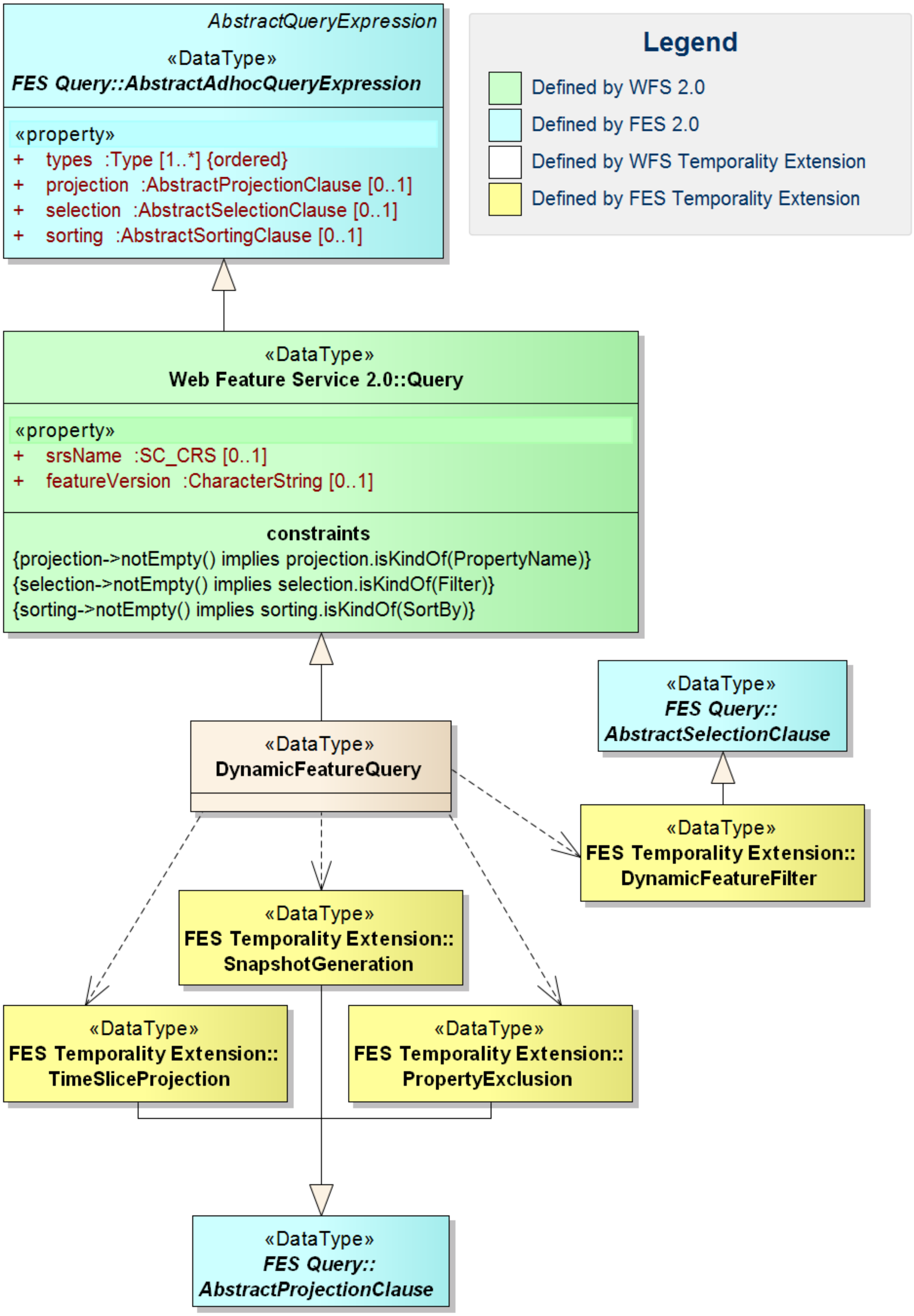

AbstractQueryExpression
«DataType»
FES Query::AbstractAdhocQueryExpression
«property»
+ types :Type [1..*] {ordered}
+ projection :AbstractProjectionClause [0..1]
+ selection :AbstractSelectionClause [0..1]
+ sorting :AbstractSortingClause [0..1]
Legend
Defined by WFS 2.0
Defined by FES 2.0
Defined by WFS Temporality Extension
Defined by FES Temporality Extension
«DataType»
Web Feature Service 2.0::Query
«property»
+ srsName :SC_CRS [0..1]
+ featureVersion :CharacterString [0..1]
constraints
{projection->notEmpty() implies projection.isKindOf(PropertyName)}
{selection->notEmpty() implies selection.isKindOf(Filter)}
{sorting->notEmpty() implies sorting.isKindOf(SortBy)}
«DataType»
FES Query::
AbstractSelectionClause
«DataType»
DynamicFeatureQuery
«DataType»
FES Temporality Extension::
DynamicFeatureFilter
«DataType»
FES Temporality Extension::
SnapshotGeneration
«DataType»
FES Temporality Extension::
TimeSliceProjection
«DataType»
FES Temporality Extension::
PropertyExclusion
«DataType»
FES Query::
AbstractProjectionClause

**Figure 8: WFS Temporality Extension Model**