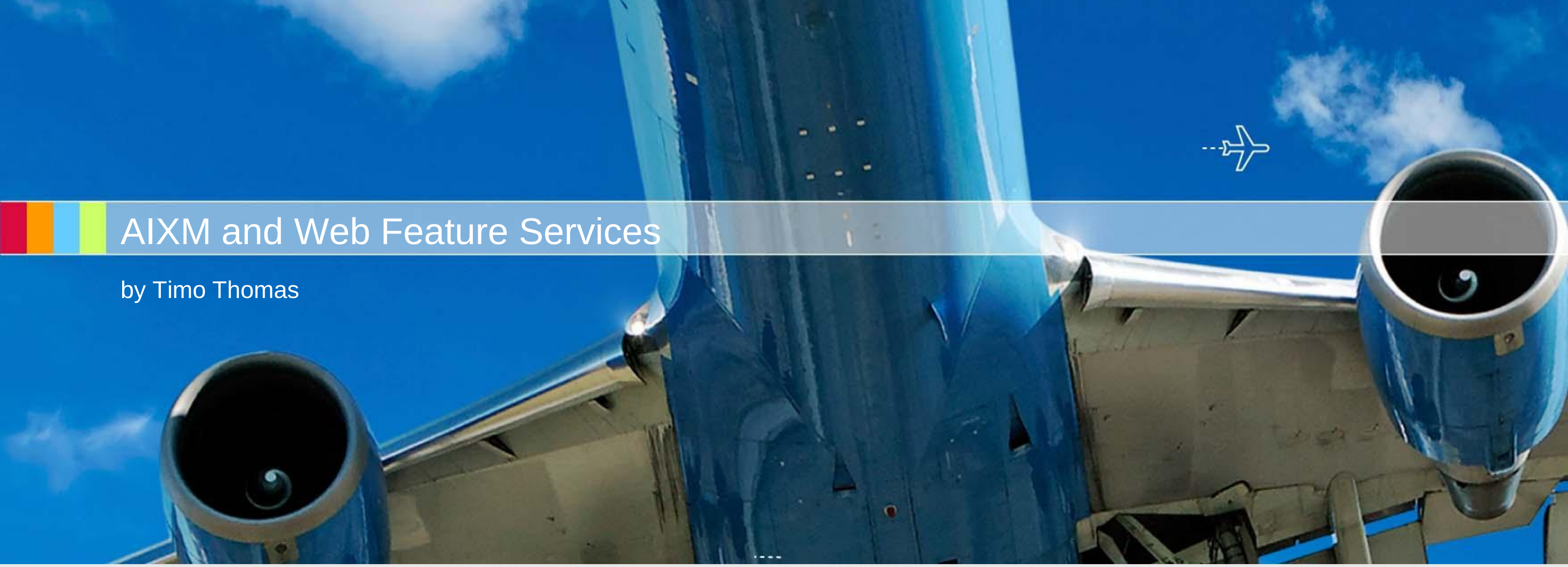

AIXM and Web Feature Services

by Timo Thomas

Bringing together the AIXM temporality model and the WFS feature concept

COMSOFT
is making the link

Query operations in WFS: GetFeature

Purpose: retrieve features

- The result is a list of features
- Individual features are selected out of the WFS‘s feature data store using either a
  - an adhoc query (ISO 19143) or a
  - stored query

Problem:

- Can‘t filter the time slices, i.e. all time slices are returned

Query operations in WFS: GetPropertyValue

Purpose: retrieve values from features

- Can be seen as a GetFeature operation + XPath expression
- The XPath expression identifies the value to retrieve
- The result is a list of values

Problems:

- The time slices do not carry the gml:identifier of the feature. If querying multiple features, the returned time slices can‘t be linked to the features they belong to.
- All queries on time slices have to be encoded in the XPath expression
  - The powerful temporal and spatial operators defined for adhoc queries can‘t be used
  - All spatial and temporal operators would have to be defined as XPath functions
  - XPath expressions are much harder to read than XML adhoc queries

Existing Projection Clauses in WFS

- The only existing projection clause (PropertyName) allows the including of *non-mandatory* properties. The term "property" applies to the direct child elements of the feature only.

- In XML Schema, an element is non-mandatory if its existence is not required to make the document (Schema-) valid. For AIXM this means there is only one time slice mandatory per feature. Thus, if no PropertyName projection clause is used, the WFS should return only one time slice per feature.

→ The specification is unclear about which property is to be returned if there is more than one. It makes no sense in AIXM (and most likely in other domains too) to choose a random one. The specification should make clear that properties with a cardinality > 1 should all be returned.

→ A projection clause capable of selecting properties out of a property list is missing

→ A projection clause capable of excluding non top-level properties is missing

Introduction of projection clause for selecting properties from a property list

- Needs a valueReference pointing to a list of properties, i.e. a property with a cardinality > 1

- Enables the user to make use of the powerful ISO 19143 filter operators

Example: Projection on PERMDELTA time slices of Airspaces with a specific UUId

```
<wfs:Query typeNames="aixm:Airspace">
  <wfs:PropertyName>gml:identifier</wfs:PropertyName> <!-- gml:identifier is optional -->
  <wfs-ext:PropertyList>
    <wfs-ext:valueReference>aixm:timeSlice</wfs-ext:valueReference>
    <fes:PropertyIsEqualTo>           <!-- any ISO 19143 filter definition is allowed here -->
      <fes:ValueReference>aixm:AirspaceTimeSlice/aixm:interpretation</fes:ValueReference>
      <fes:Literal>PERMDELTA</fes:Literal>
    </fes:PropertyIsEqualTo>
  </wfs-ext:PropertyList>
  <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>
```

Response of the example query with PropertyList projection

```
<wfs:FeatureCollection>
  <wfs:member>
    <aixm:Airspace>
      <gml:identifier codeSpace="">0083defb-b42e-4417-9be2-7aba2db2674d</gml:identifier>
      <aixm:timeSlice>
        <aixm:AirspaceTimeSlice>
          …
          <aixm:interpretation>PERMDELTA</aixm:interpretation>
          …
        </aixm:AirspaceTimeSlice>
      </aixm:timeSlice>
      <aixm:timeSlice>
        <aixm:AirspaceTimeSlice>
          …
          <aixm:interpretation>PERMDELTA</aixm:interpretation>
          …
        </aixm:AirspaceTimeSlice>
      </aixm:timeSlice>
      … <!-- more PERMDELTA time slices if any -->
    </aixm:Airspace>
  </wfs:member>
</wfs:FeatureCollection>
```

PropertyList projection clause explained

```
<wfs-ext:PropertyList>
    <wfs-ext:valueReference>aixm:timeSlice</wfs-ext:valueReference>
    <fes:PropertyIsEqualTo>
      <fes:ValueReference>aixm:AirspaceTimeSlice/aixm:interpretation</fes:ValueReference>
      <fes:Literal>PERMDELTA</fes:Literal>
    </fes:PropertyIsEqualTo>
</wfs-ext:PropertyList>
```

wfs-ext: namespace (prefix) for additional (extended) elements in WFS

wfs-ext:PropertyList: a projection clause capable of filtering elements from a list. There may be multiple projection clauses, wich is fine. This element contains a filter expression like fes:Filter does. Thus, we can use all of the filter operators (spatial, temporal etc.).

valueReference: an XPath expression pointing to a list of elements to filter (here: aixm:timeSlice)

Problem: features without time slices may be returned

- If the PropertyList projection clause finds no matching property, the returned list is "empty".

- In the previous example this would be the case if the Airspace feature had no PERMDELTA time slices at all. The result would be

```
<wfs:FeatureCollection>
  <wfs:member>
    <aixm:Airspace>
      <gml:identifier codeSpace="">0083defb-b42e-4417-9be2-7aba2db2674d</gml:identifier>
    </aixm:Airspace>
  </wfs:member>
</wfs:FeatureCollection>
```

- This is a Schema-invalid document, as there at list one time slice required. We would rather have no result at all.

Possible work around

- If the user wants to wants no features returned that do not match a PropertyList projection clause, he has to include a similar filter in the selection part of the query:

```
<wfs:Query typeNames=“aixm:Airspace“>
  <wfs:PropertyName>gml:identifier</wfs:PropertyName>
  <wfs-ext:PropertyList>
    <wfs-ext:valueReference>aixm:timeSlice</wfs-ext:valueReference>
    <fes:PropertyIsEqualTo>
      <fes:ValueReference>aixm:AirspaceTimeSlice/aixm:interpretation</fes:ValueReference>
      <fes:Literal>PERMDELTA</fes:Literal>
    </fes:PropertyIsEqualTo>
  </wfs-ext:PropertyList>
  <fes:Filter>
     <fes:And>
       <fes:PropertyIsEqualTo>
         <fes:ValueReference>gml:identifier</fes:ValueReference>
         <fes:Literal>0083defb-b42e-4417-9be2-7aba2db2674d</fes:Literal>
       </fes:PropertyIsEqualTo>
       <fes:PropertyIsEqualTo>
         <fes:ValueReference>aixm:timeSlice/aixm:AirspaceTimeSlice/aixm:interpretation</fes:ValueReference>
         <fes:Literal>PERMDELTA</fes:Literal>
       </fes:PropertyIsEqualTo>
    </fes:And>
  </fes:Filter>
</wfs:Query>
```

Better: avoid filter duplication with references

- Introduce ids on filter operations to make them referencable:

```
<wfs:Query typeNames="aixm:Airspace">
  <wfs:PropertyName>gml:identifier</wfs:PropertyName>
  <wfs-ext:PropertyList>
    <wfs-ext:valueReference>aixm:timeSlice</wfs-ext:valueReference>
    <wfs-ext:filterOperation ref="filterOp1"/>
  </wfs-ext:PropertyList>
  <fes:Filter>
     <fes:And>
       <fes:PropertyIsEqualTo>
         <fes:ValueReference>gml:identifier</fes:ValueReference>
         <fes:Literal>0083defb-b42e-4417-9be2-7aba2db2674d</fes:Literal>
       </fes:PropertyIsEqualTo>
       <fes:PropertyIsEqualTo id="filterOp1">
         <fes:ValueReference>aixm:timeSlice/aixm:AirspaceTimeSlice/aixm:interpretation</fes:ValueReference>
         <fes:Literal>PERMDELTA</fes:Literal>
       </fes:PropertyIsEqualTo>
    </fes:And>
  </fes:Filter>
</wfs:Query>
```

- Constraint: the XPath of the PropertyList's valueReference must be a prefix of all XPaths in the filter operation

2nd extension: a projection clause to exclude non top-level properties

- The properties are identified by an XPath expression
- In AIXM this would enable the user to filter out information he may not always need like metadata, notes etc. Example:

```
<wfs:Query typeNames="aixm:Airspace">
  <wfs-ext:ExcludeProperties valueReference="//aixm:timeSliceMetadata"/>
  <wfs-ext:ExcludeProperties valueReference="//gml:metaDataProperty"/>
  <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>
```

- Remark: the filtering may result in (Schema-)invalid documents. It's the users responsibility to avoid this. The service may return an error in such a case.

XML Schemas for the new features

- No change of existing schemas is necessary
- New projection clauses can be introduced as a member of the substitution group `fes:AbstractProjectionClause`.
- New filter operators can be introduced as a member of the substitution group `fes:extensionOps` and an extension of `fes:ExtensionOpsType`.