AIXM 5 Temporality Model

DNSWG #2

Washington DC, 5 December 2007

EUROCONTROL

Contents

- The Temporality Model
- Applying the Temporality Model
- Usage Examples

EUROCONTROL

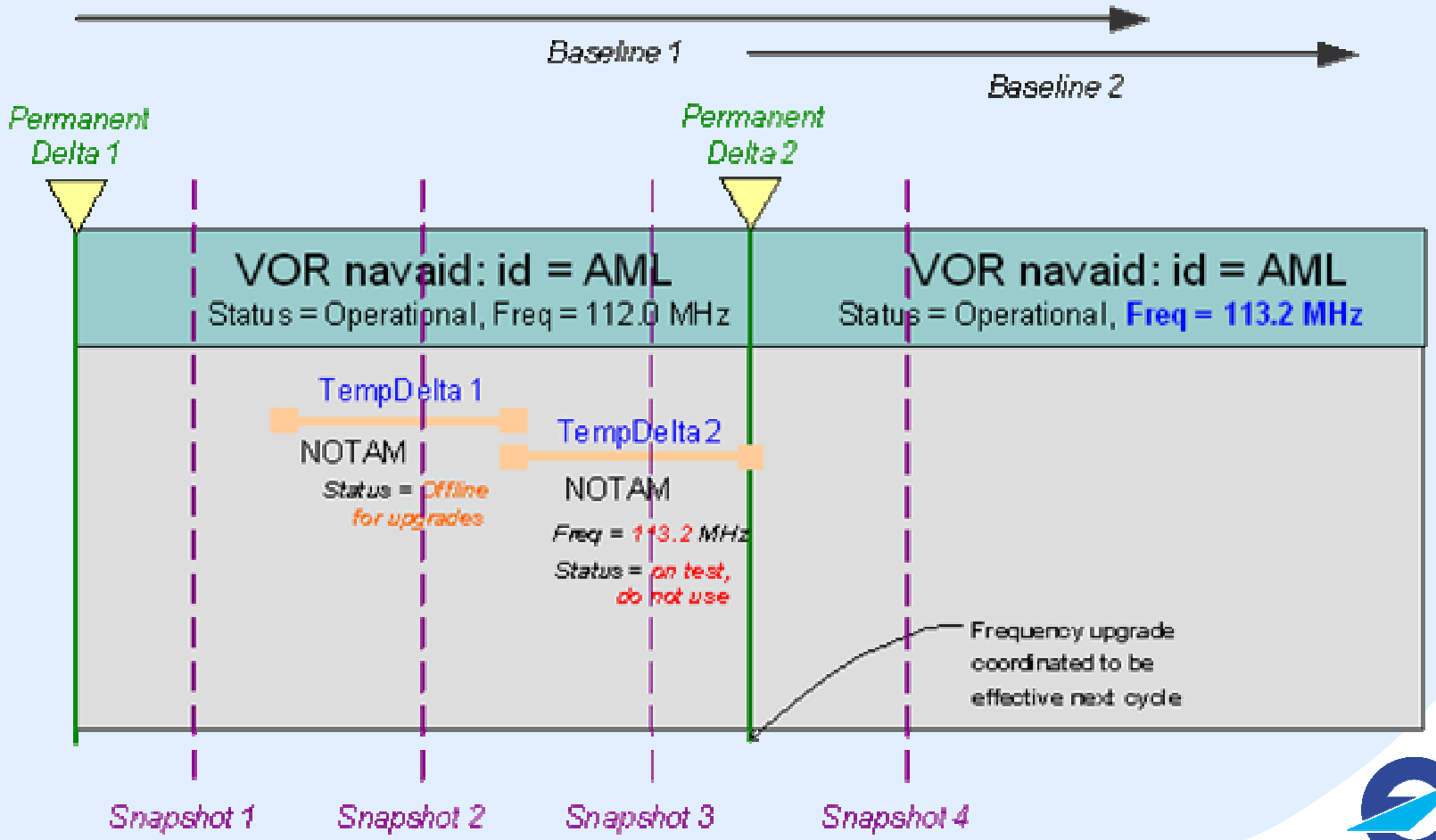
Summary - Time Slice types

- ***BASELINE*** = A kind of Time Slice that describes the feature state (the set of all feature's properties) as result of a permanent change;
- ***PERMDELTA*** = A kind of Time Slice that describes the difference in a feature state as result of a permanent change;
- ***TEMPDELTA*** = A kind of Time Slice that describes the transitory changes of a feature state during a temporary event;
- ***SNAPSHOT*** = A kind of Time Slice that describes the state of a feature at a time instant, as result of combining the actual BASELINE Time Slice valid at that time instant with all TEMPDELTA Time Slices applicable at that time instant.

EUROCONTROL

An Example: Navaid frequency change

Imagine that AML Navaid undergoes an upgrade that changes its frequency from 112.0 MHz to 113.2 MHz…



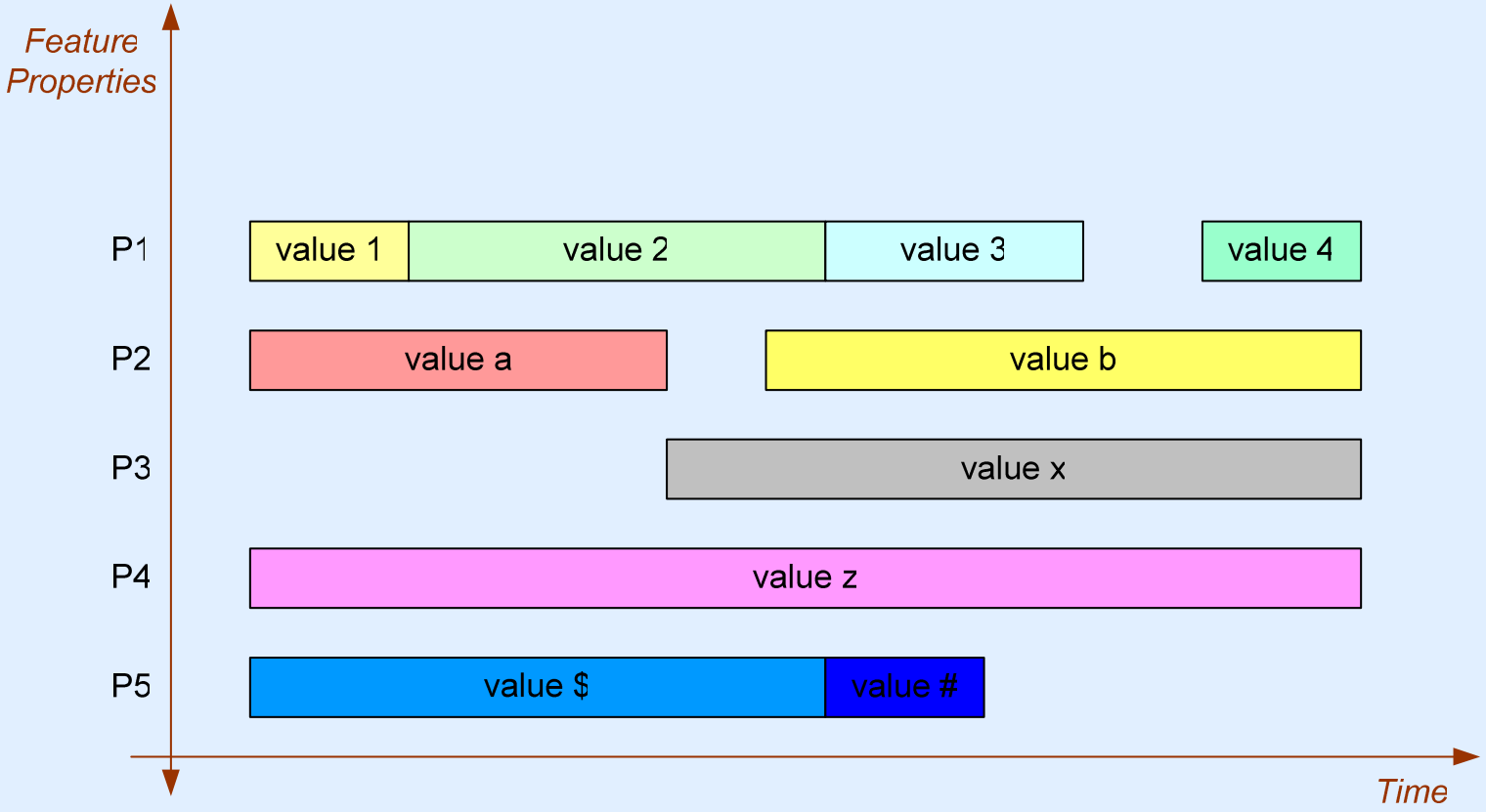
EUROCONTROL

Building the Temporality Model

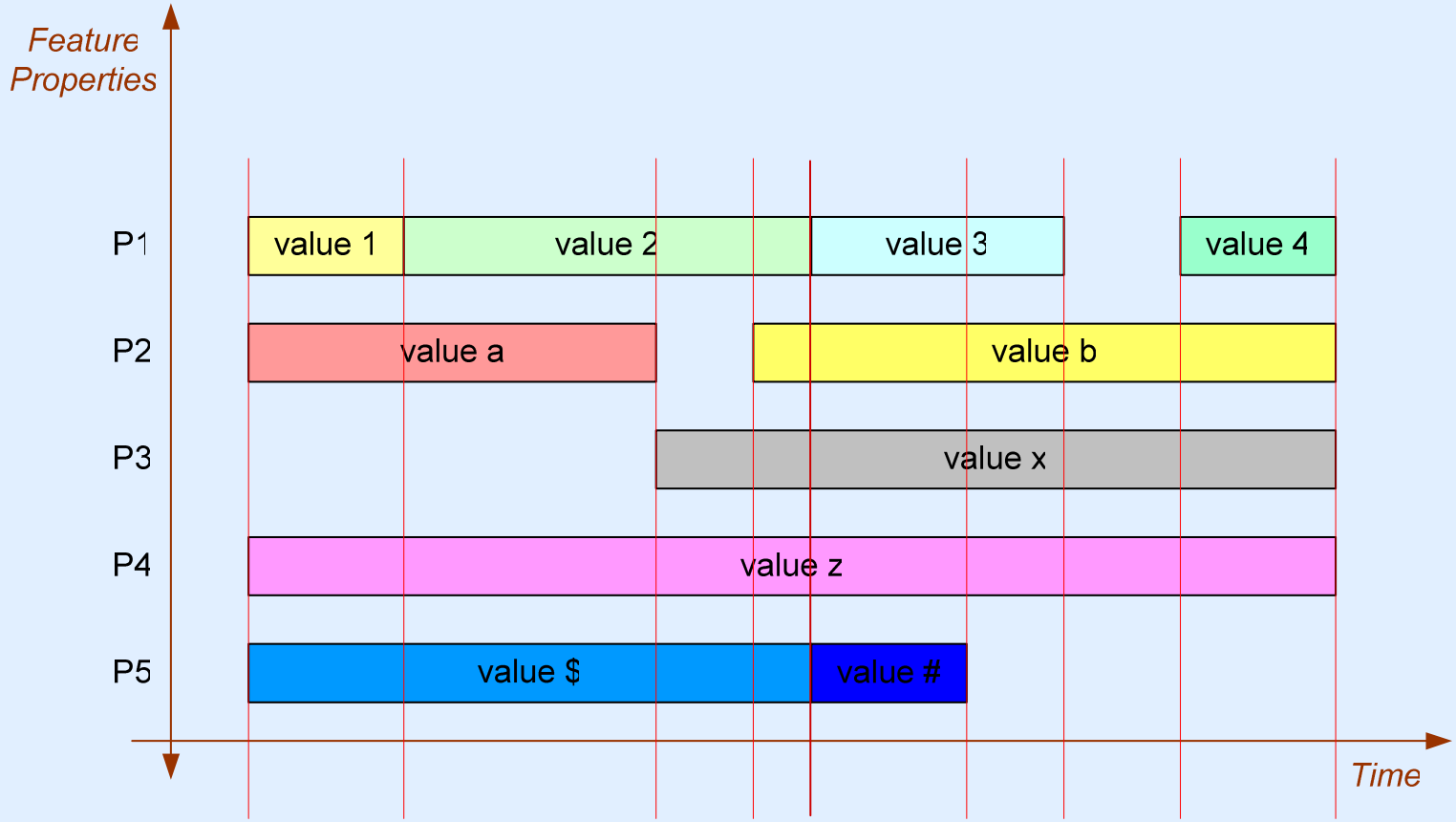
- Time is essential for AI
- For operational reasons, distinction between:
  - Permanent changes
  - Temporary status
- Two levels:
  - Life of the feature
  - Changes in properties

EUROCONTROL

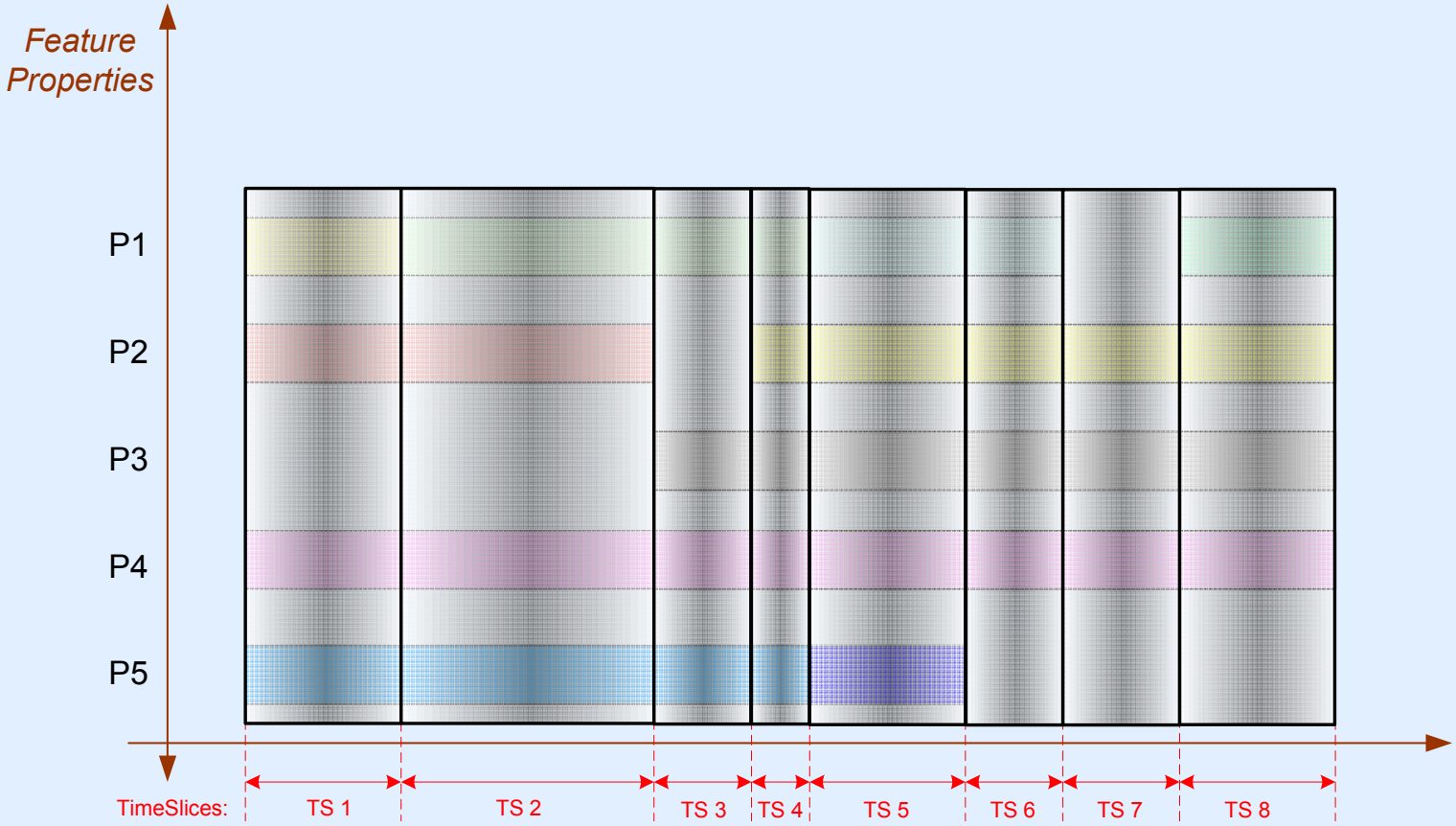
Time varying properties



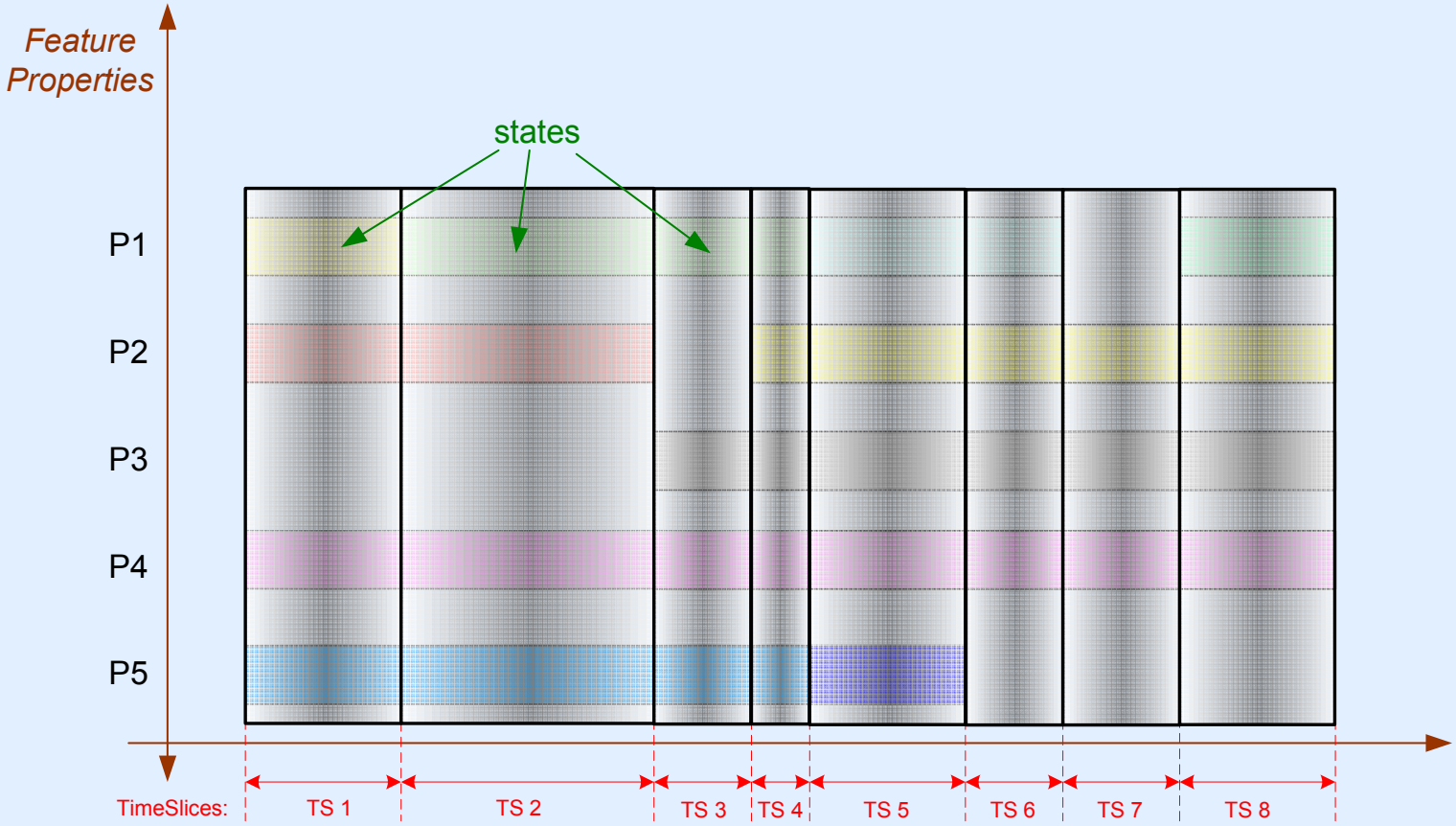
The basic Time Slice model



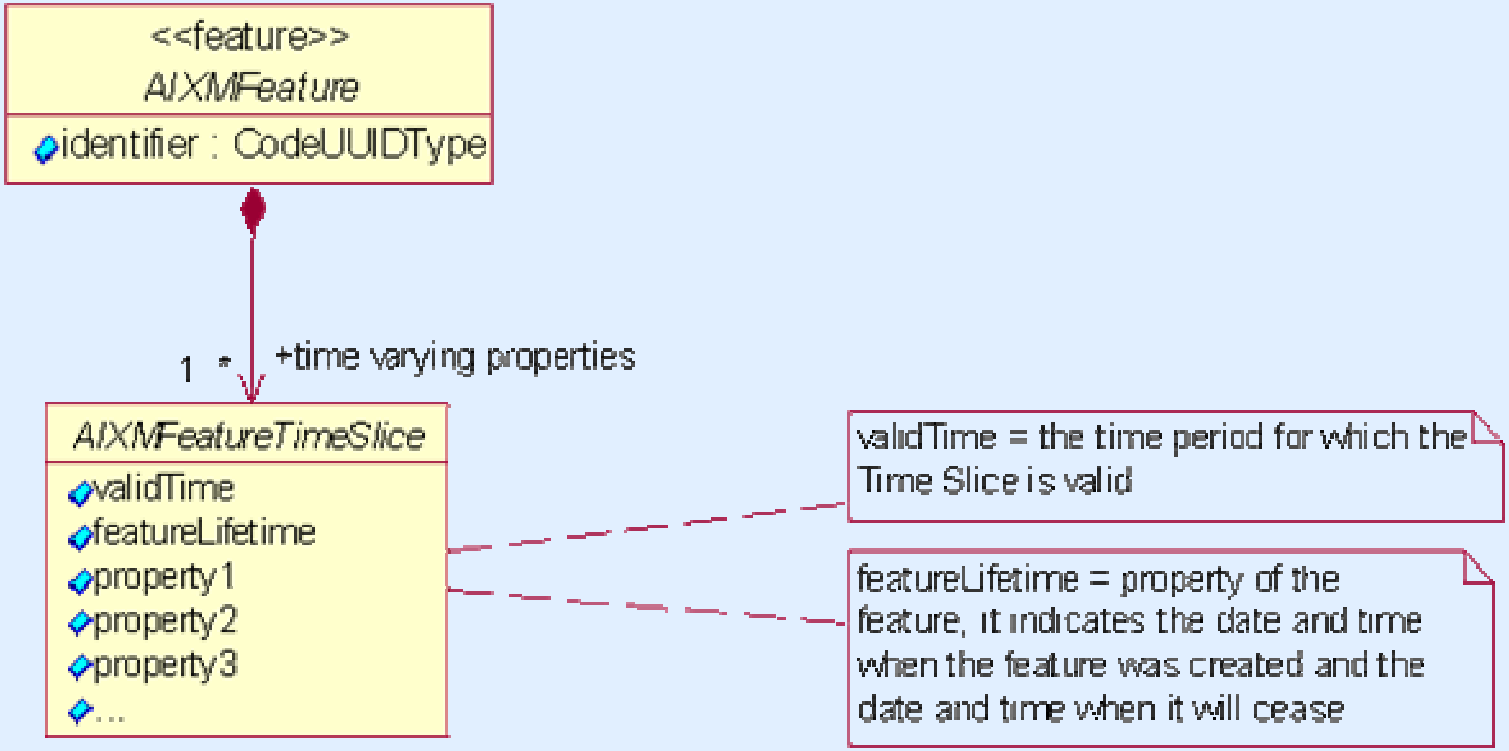
The basic Time Slice model



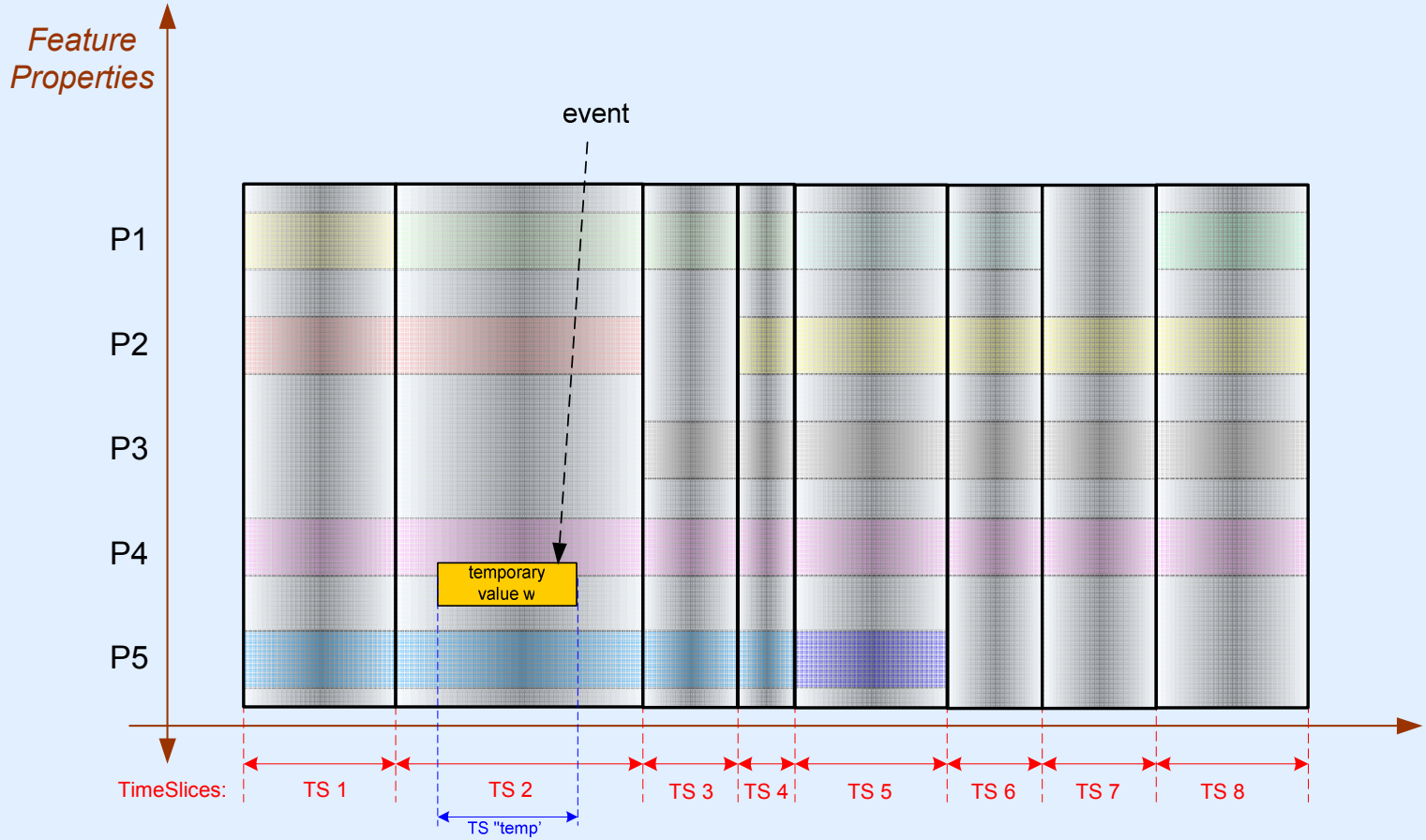
The basic Time Slice model



The basic Time Slice model
UML

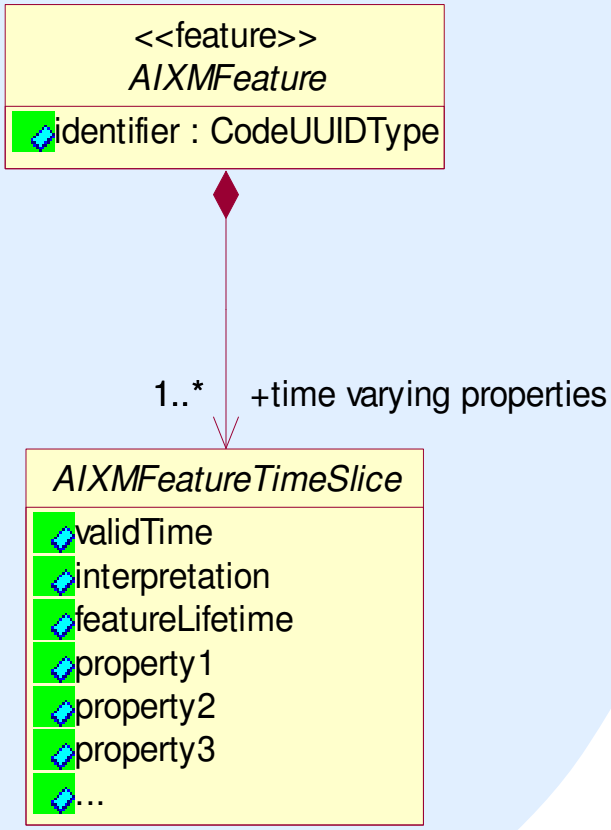


Temporary events – digital NOTAM



Temporary events

- Two kinds of Time Slices
  - ***Baseline*** = describes the feature state (the set of all feature's properties) as result of a permanent change;
  - ***Temporary Delta*** = describes the transitory overlay of a feature state during a temporary event.



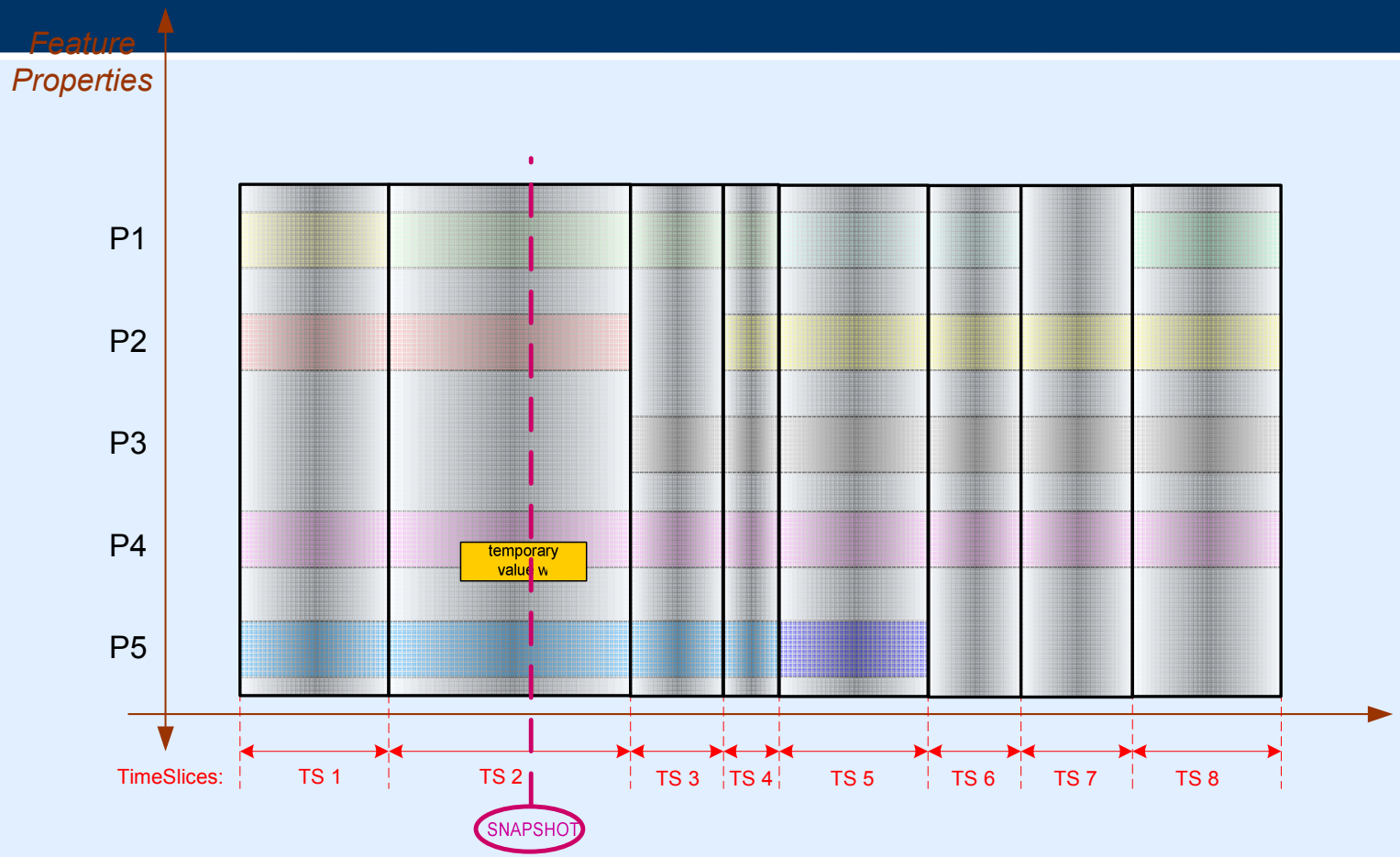
The model up to this point

BASELINE & TEMPDELTA

- Compliant with the requirements for
  - Completeness - all temporal states must be representable;
  - Minimalism - use of minimal number of elements;
  - Consistency - no reuse of elements with different meaning;
  - Context-free - meaning of (atomic) elements independent of context; no functional dependency of (atomic) elements at the data encoding level;
- But, a bit inconvenient for real applications…

EUROCONTROL

Current status of a feature



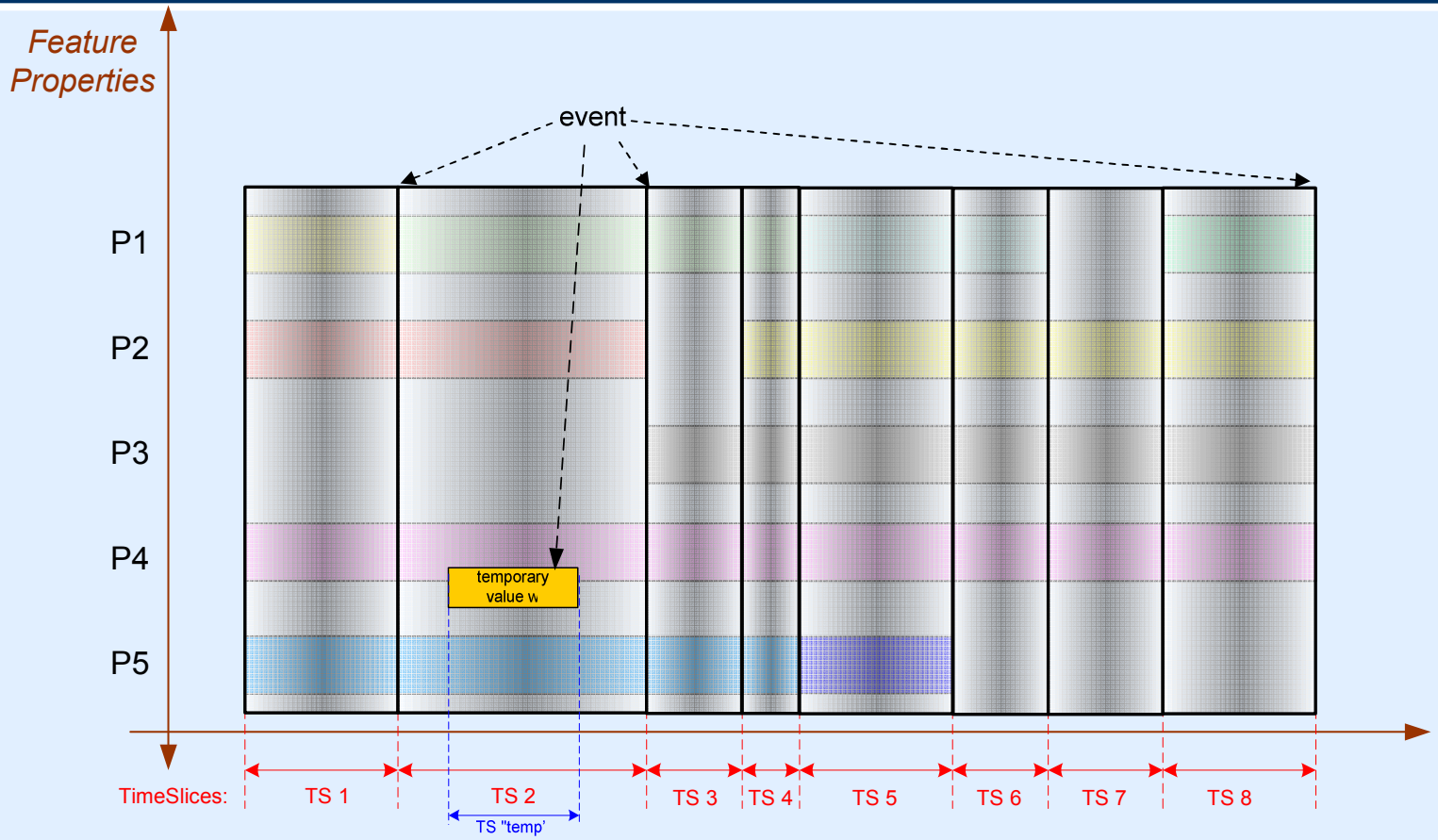
SNAPSHOT = describes the state of a feature at a time instant, as result of combining the actual BASELINE Time Slice valid at that time instant with all TEMPDELTA Time Slices applicable at that time instant.

Permanent changes

- Two types of applications:
  - “Pull” Systems - provide an interface by which a client can query the aeronautical information;
  - “Push” Systems - generate and transmit to the client notifications about aeronautical information changes.
    - Need to notify a permanent change
    - Need to notify “end of life”

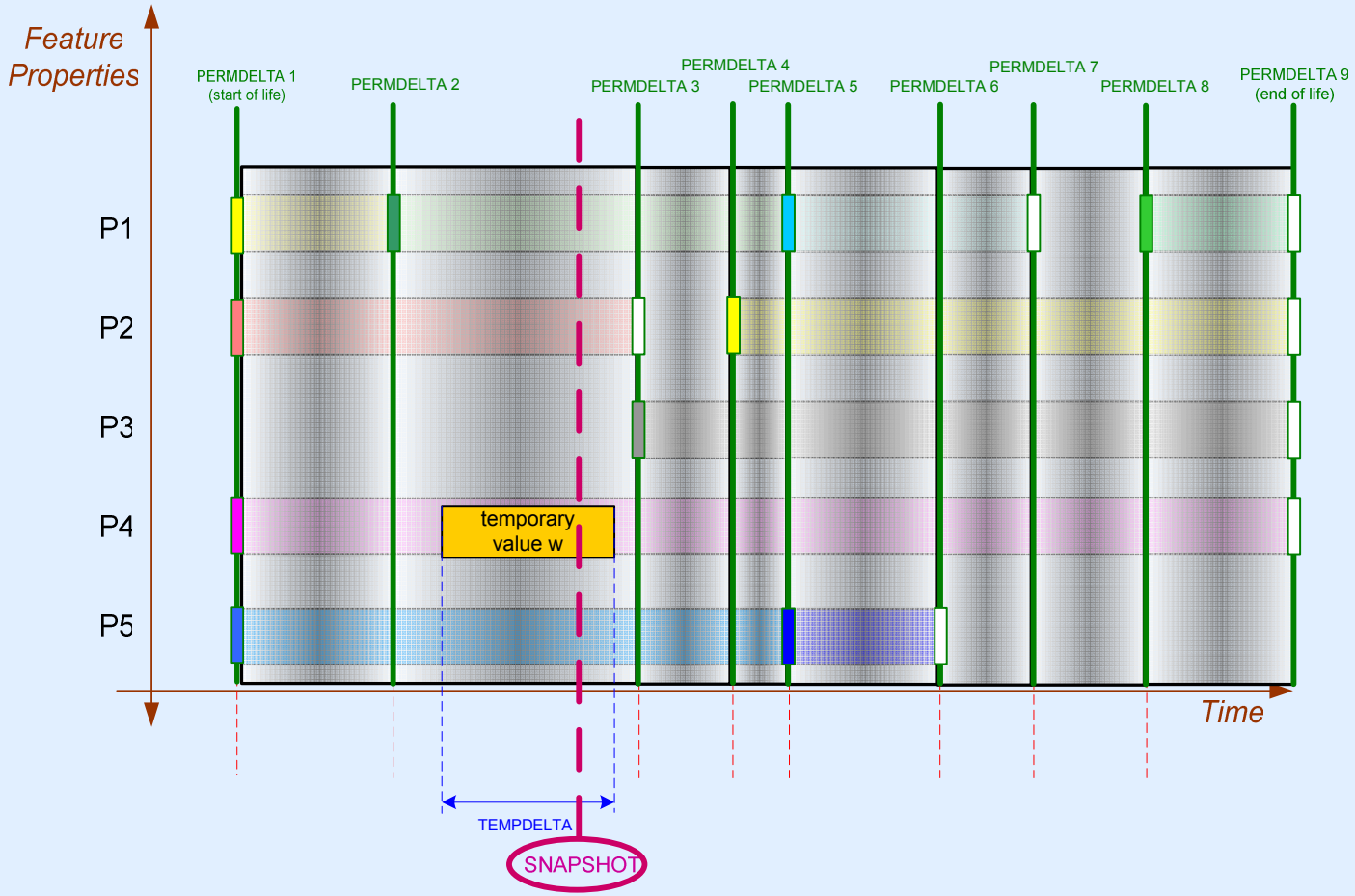
EUROCONTROL

Permanent changes

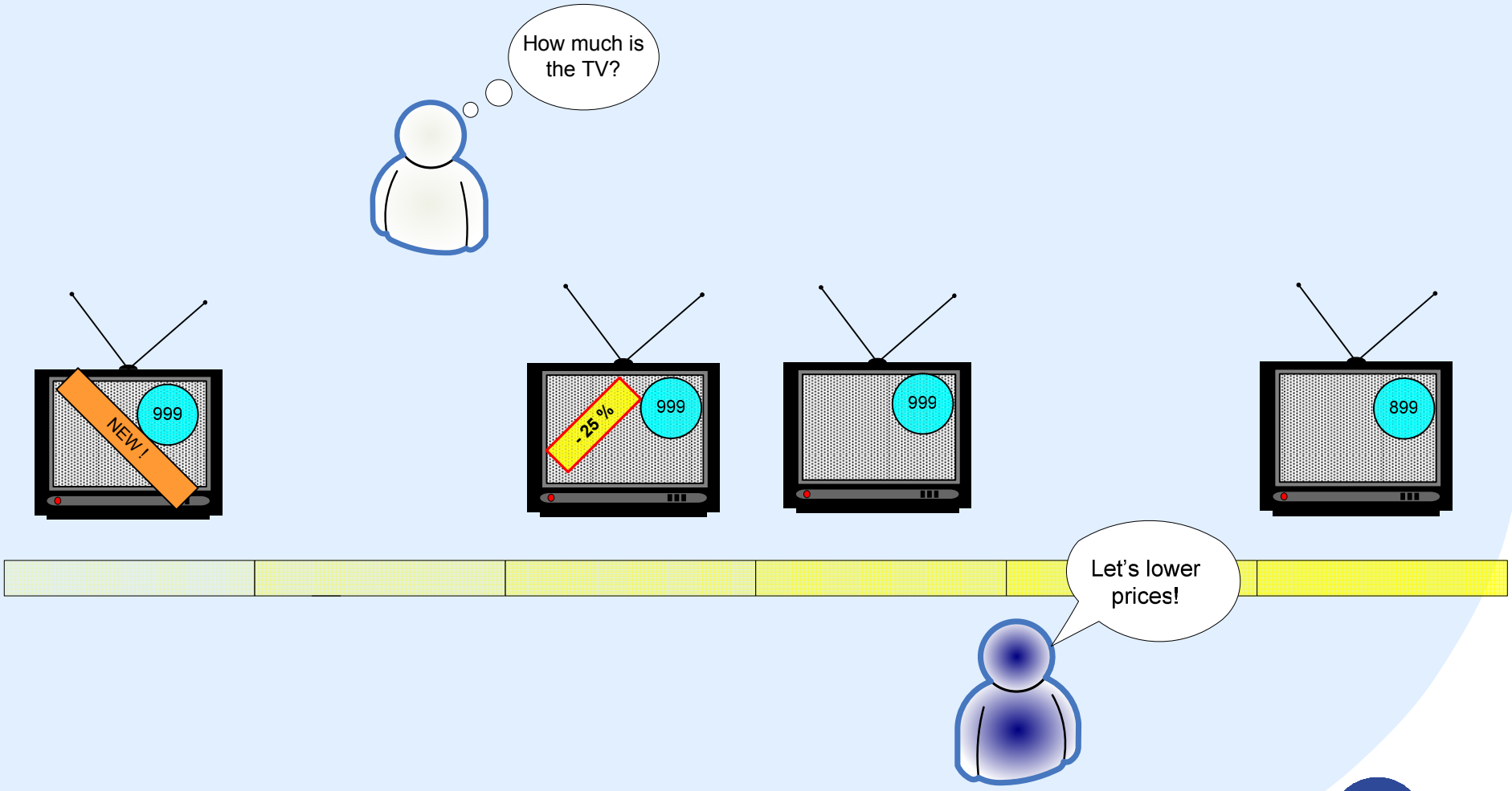


PERMDELTA = A kind of Time Slice that describes the difference in a feature state as result of a permanent change.

Time Slice Types



Temporality – something different

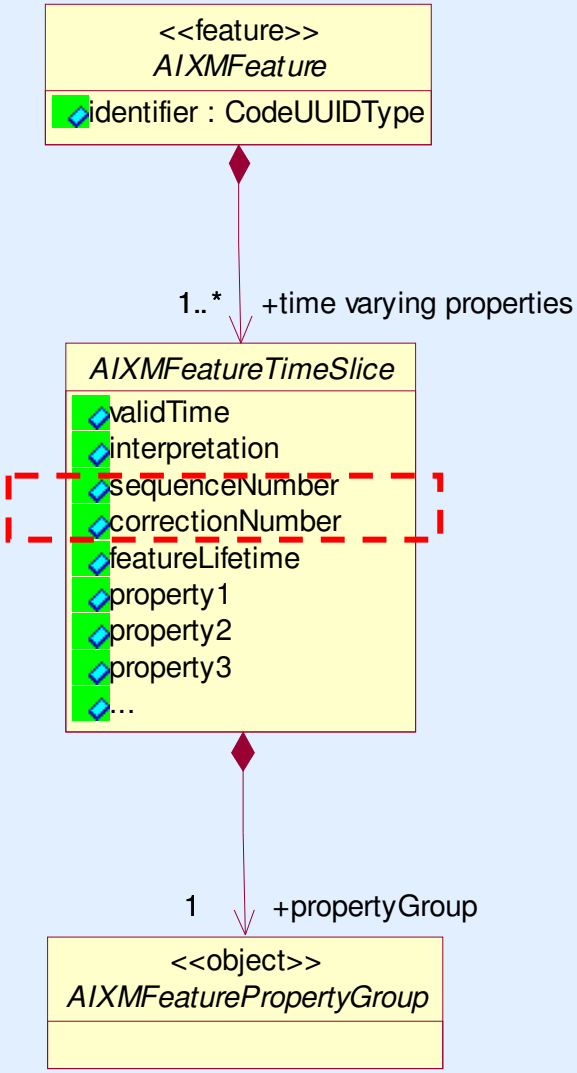


Corrections

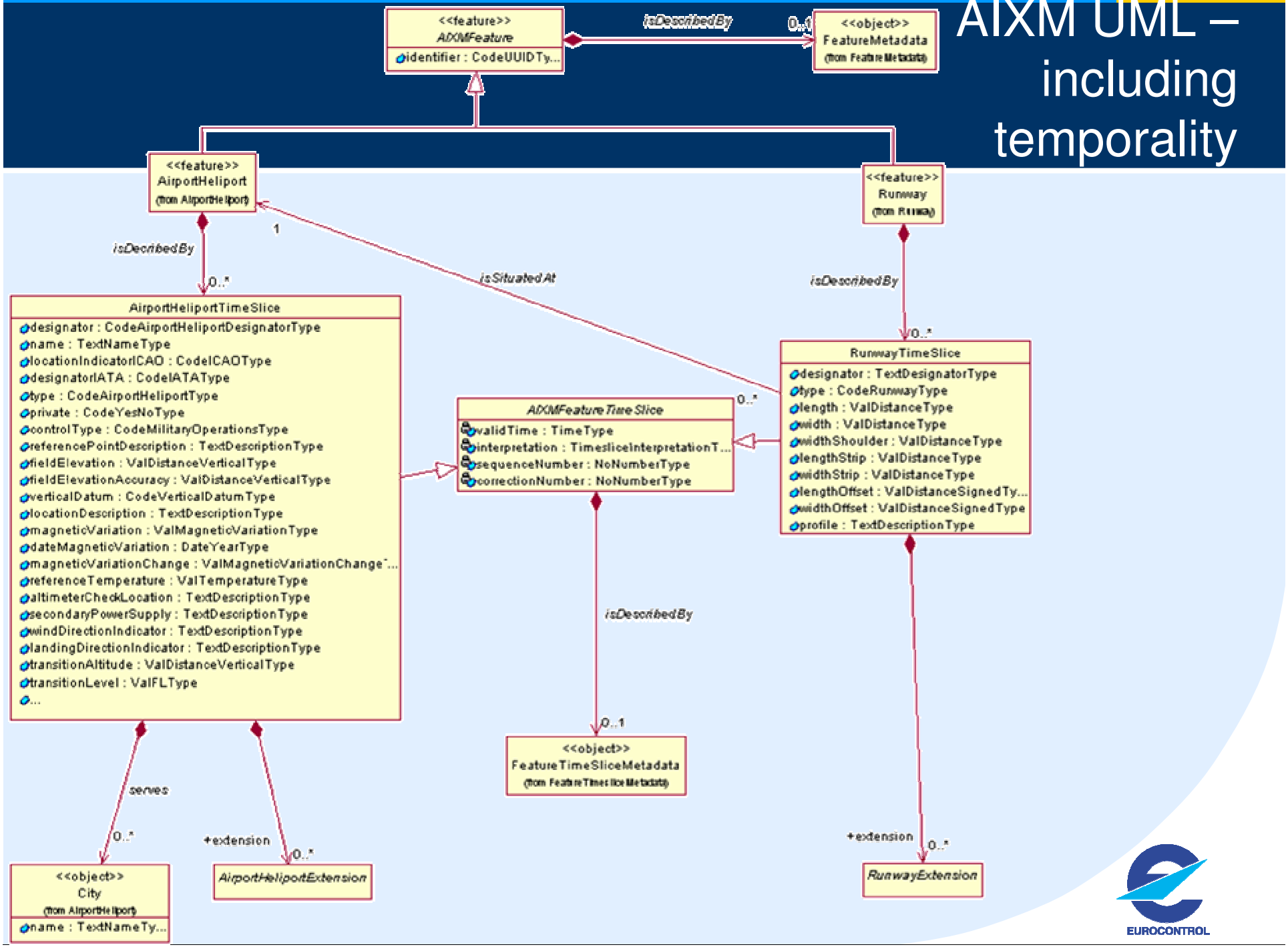
- Information about events that are planned to take place in future
  - update the already communicated information
  - may affect “event” Time Slices - of type PERMDELTA and TEMPDELTA.
- Solution
  - Identify Time Slice concerned by a “sequence number”
  - Also needed: “correctionNumber”

EUROCONTROL

Final UML representation



AIXM UML – including temporality



EUROCONTROL

Summary - Time Slice types

- ***BASELINE*** = A kind of Time Slice that describes the feature state (the set of all feature's properties) as result of a permanent change;
- ***PERMDELTA*** = A kind of Time Slice that describes the difference in a feature state as result of a permanent change;
- ***TEMPDELTA*** = A kind of Time Slice that describes the transitory changes of a feature state during a temporary event;
- ***SNAPSHOT*** = A kind of Time Slice that describes the state of a feature at a time instant, as result of combining the actual BASELINE Time Slice valid at that time instant with all TEMPDELTA Time Slices applicable at that time instant.

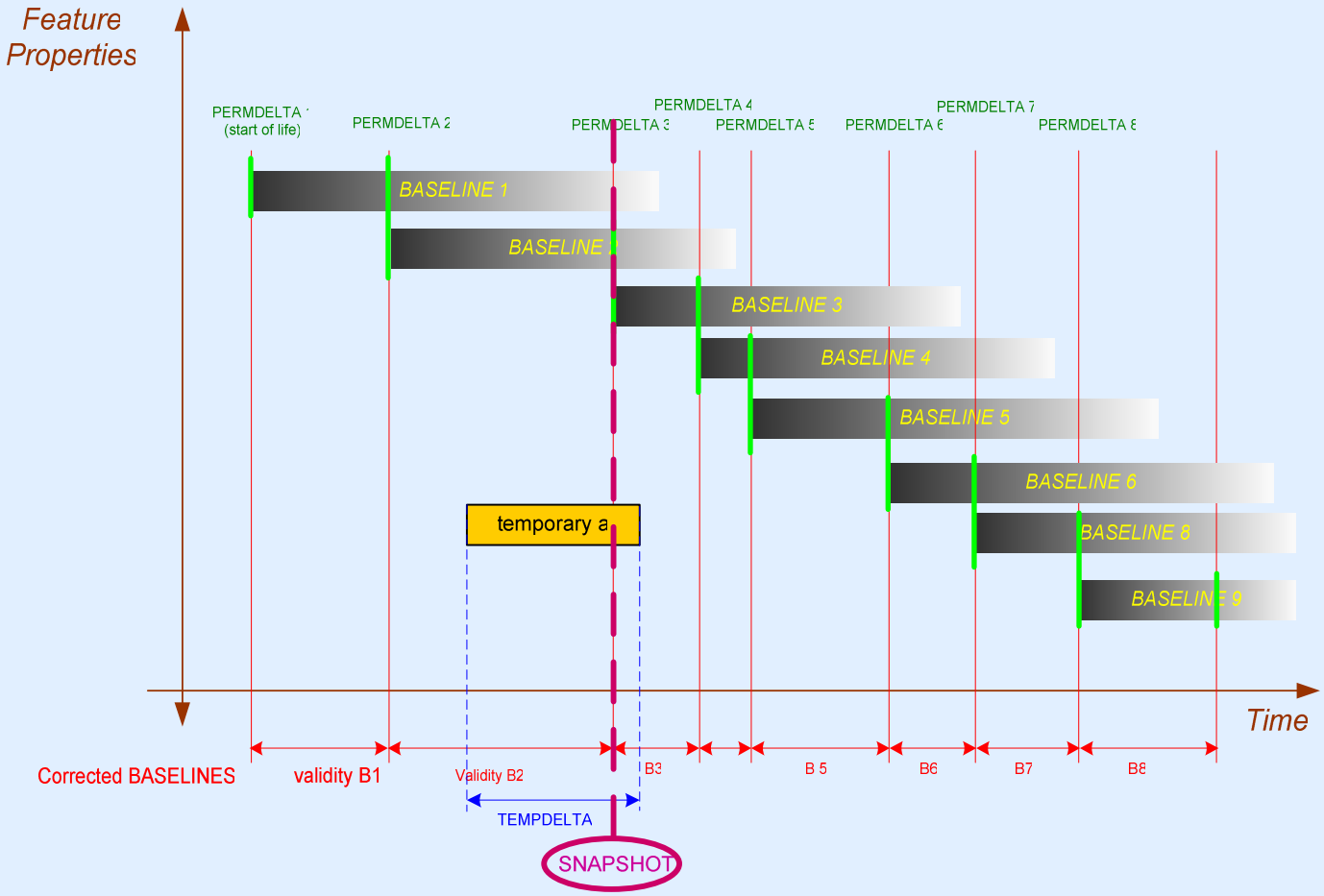
EUROCONTROL

Applying the model

- Undetermined end of validity
- Multiple occurring properties
- Complex properties
- Identifying the feature affected
- Cancelling a time slice

EUROCONTROL

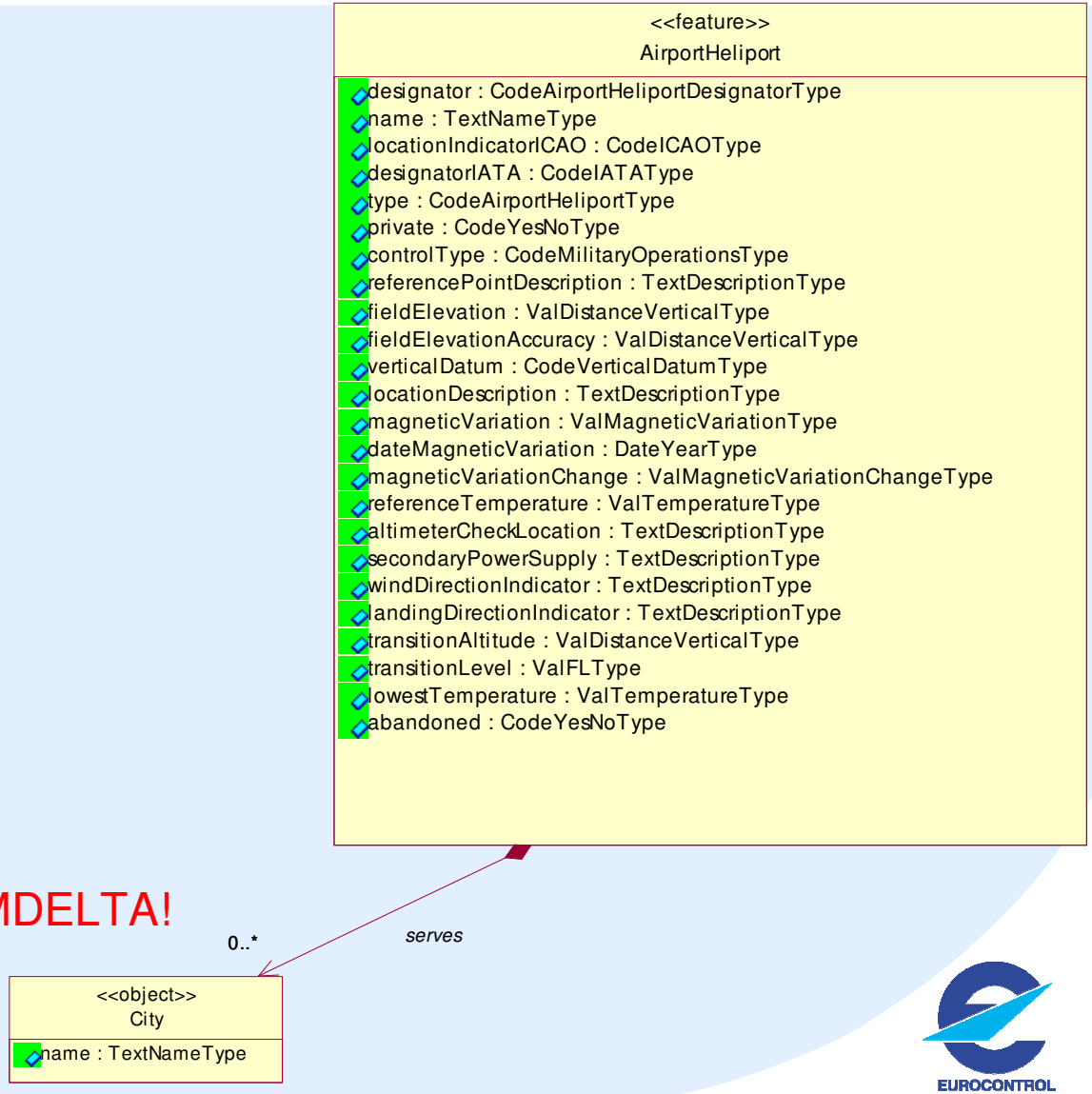
Baseline – undetermined end of validity



“Delta” for multiple occurring properties

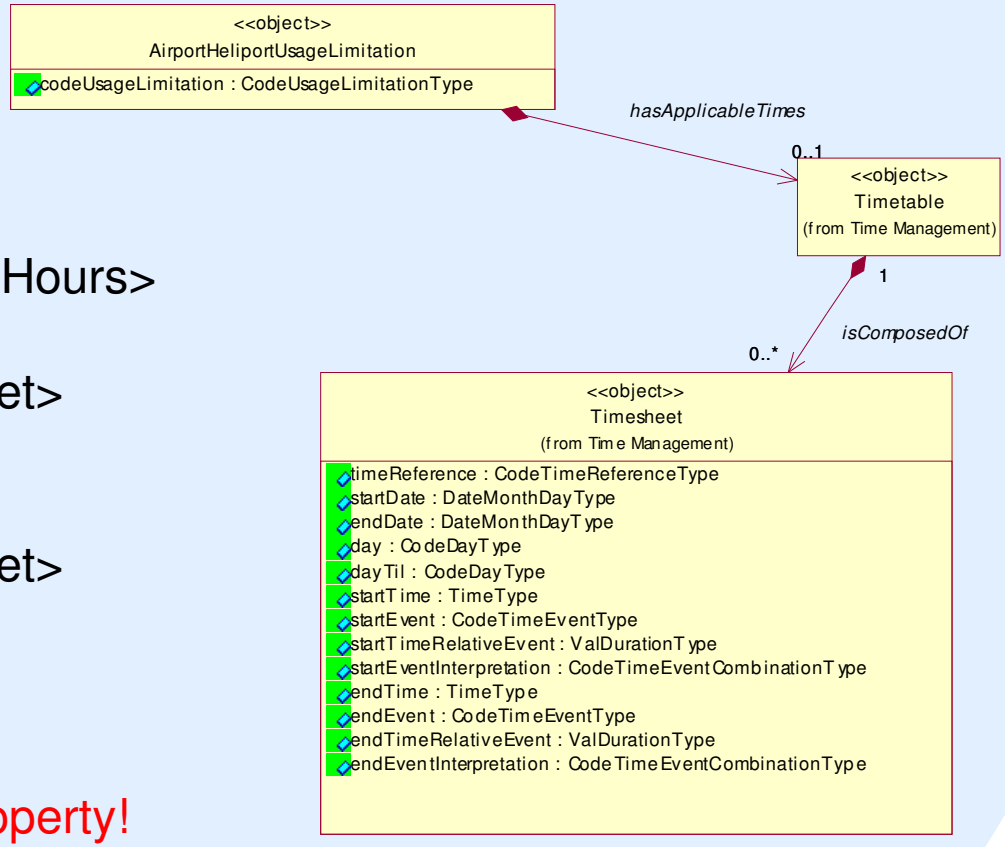
```
<AirportHeliportTimeSlice>
  ….
  <serves>
     <City>
          <name>…</name>
     </City>
  </serves>
  <serves>
     <City>
          <name>…</name>
     </City>
  </serves>
…
```

Rule: Include all occurrences of the <serves> property in a TEMPDELTA or a PERMDELTA!



“Delta” for complex properties

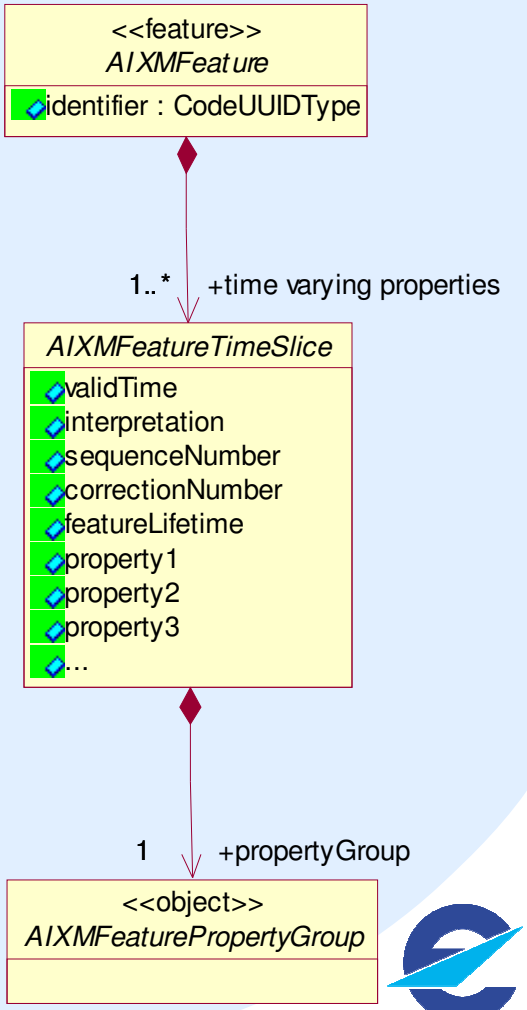
```
<AirportHeliportLimitation>
  ….
  <hasApplicableTimes>
     <Timetable>
        <workingHours>…</workingHours>
       <isComposedOf>
          <Timesheet>…</Timesheet>
       </isComposedOf>
       <isComposedOf>
          <Timesheet>…</Timesheet>
       </isComposedOf>
       ….
  </hasApplicableTimes>
```



Rule: Include the whole complex property!
(“delta” applies strictly to the 1st level of properties!)

Identifying the feature affected by “delta”

```
<Feature>
  <gml:identifier>35149fb2-971f-492c-94c2-5671d4580587</gml:identifier>
  <timeSlice>
    <validTime>….</validTime>
    <interpretation>BASELINE</interpretation>
    <sequenceNumber>23</sequenceNumber>
    <property1>….</property1>
    <property2>….</property2>
    ….
  </timeSlice>
  <timeSlice>
    <validTime>….</validTime>
    <interpretation>TEMPDELTA</interpretation>
    <sequenceNumber>23</sequenceNumber>
    <property4>….</property4>
  </timeSlice>
</Feature>
```



Cancelling a Time Slice

- Indicate a cancellation:
  - validTime is empty
  - Use same sequence number
  - Use a higher correction number

```
<Feature>
  <gml:identifier>35149fb2-971f-492c-94c2-
   5671d4580587</gml:identifier>
  <timeSlice>
    <gml:validTime nilReason="inapplicable" />
    <interpretation>PERMDELTA</interpretation>
    <sequenceNumber>23</sequenceNumber>
    <correctionNumber>1</correctionNumber>
    ….
  </timeSlice>
</Feature>
```


EUROCONTROL

Other implementation considerations

- The temporal model is complete.
- No requirement on systems to support all the types of Time Slices
  - E.g. some systems may only store BASELINE Time Slices and disregard temporary changes
- Interacting parties should negotiate specific temporal data exchange requirements

EUROCONTROL

Business Rules

- These are to be developed. Examples:
- TEMPDELTA Time Slices cannot change the start of life and the end of life
- PERMDELTA shall have a TimeInstant as the validTime

EUROCONTROL

Usage examples

- Start of life
- Change of properties
- Temporary situation (digital NOTAM)
- End of life
- …

EUROCONTROL

Start of life

- Feature creation (commissioning)

Feature
- gml:identifier

TimeSlice
- validTime = timeInstant…
- *interpretation = PERMDELTA*
- sequenceNumber = 1
- startOfLife = same timeInstant…
- property 1
- property 2
- property 3
- property 4
- property 5

TimeSlice
- validTime = timeInterval with undetermined end
- *interpretation = BASELINE*
- sequenceNumber = 1
- startOfLife = same timeInstant…
- property 1
- property 2
- property 3
- property 4
- property 5

EUROCONTROL

Change of properties

- Permanent change (amendment)

Feature
- gml:identifier

TimeSlice
- validTime = timeInstant…
- *interpretation = PERMDELTA*
- sequenceNumber = 2
- property 3 (new value)
- property 5 (new value)

TimeSlice
- validTime = timeInterval with undetermined end …
- *interpretation = BASELINE*
- sequenceNumber = 2
- startOfLife = timeInstant…
- property 1
- property 2
- property 3 (new value)
- property 4
- property 5 (new value)

EUROCONTROL

Temporary event

Feature
- gml:identifier

TimeSlice
- validTime = timeInterval…
- *interpretation = TEMPDELTA*
- sequenceNumber = 1
- property 4 (temporary value)

Optional: include the BASELINE/SNAPSHOT

EUROCONTROL

End of Life

- Feature withdrawing (decommissioning)

Feature
- gml:identifier

Timeslice
- validTime = timeInstant…
- *interpretation = PERMDELTA*
- sequenceNumber = 3
- endOfLife = same timeInstant…

EUROCONTROL

Feature histories

- A history can be built up by transmitting the sequence of changes:

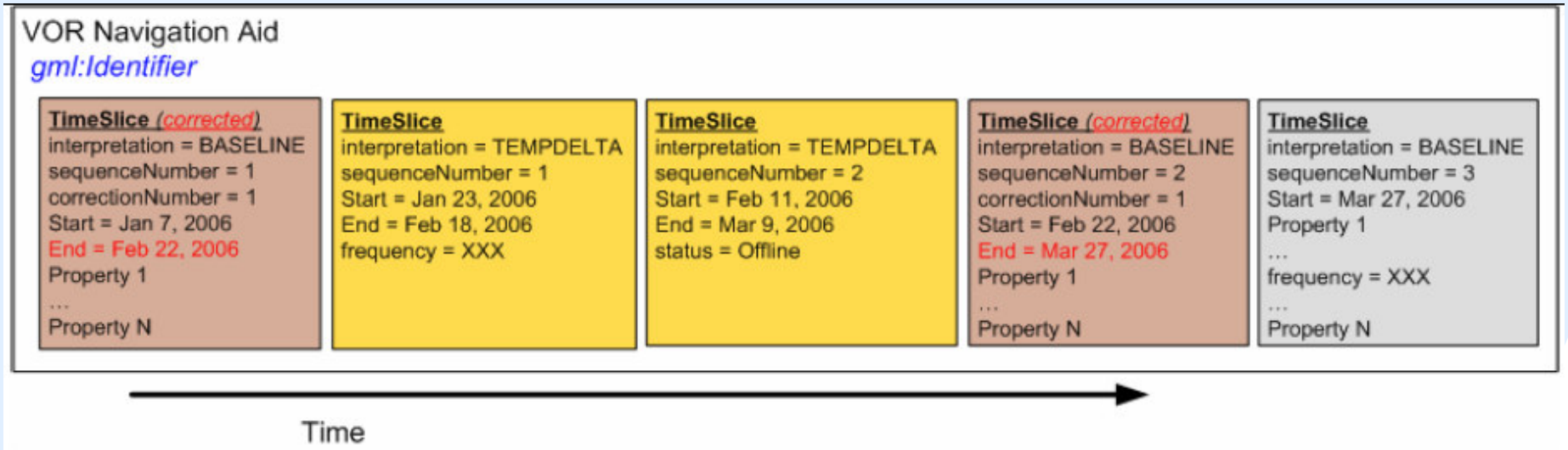
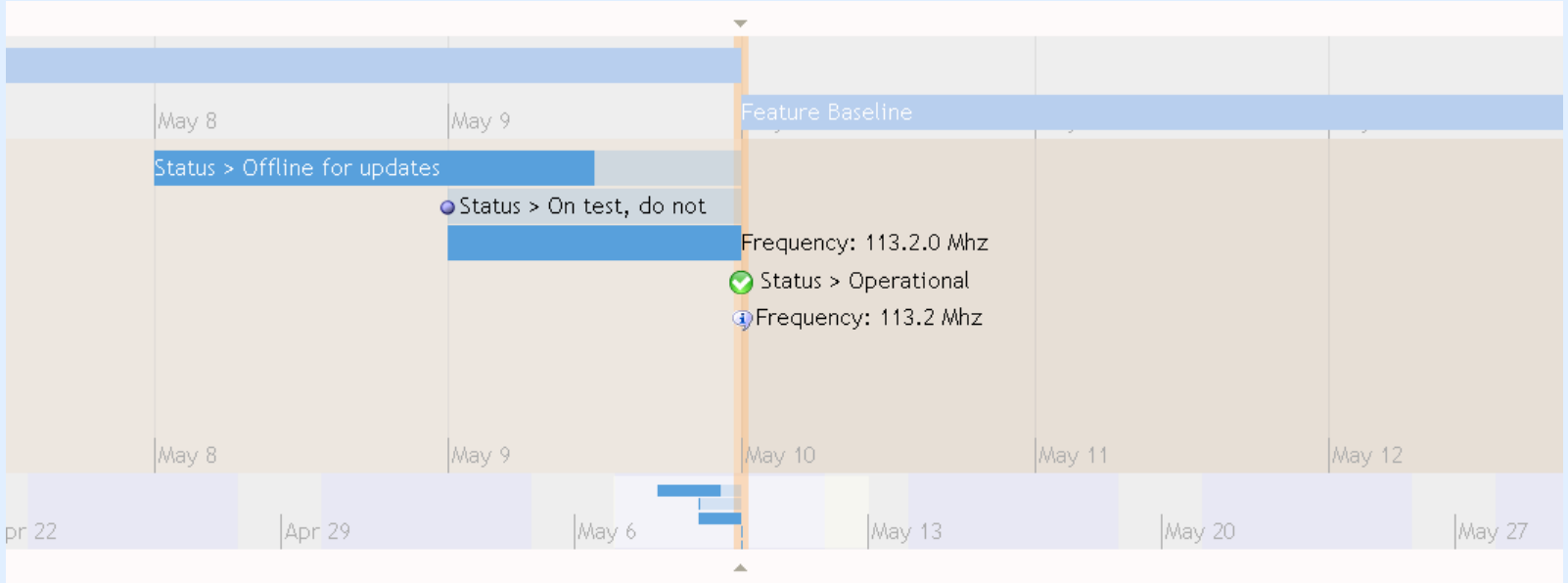


Figure 15: TimeSlices for the VOR navigation aid history

EUROCONTROL

xNOTAM TimeLine



EUROCONTROL

AIXM Temporality document

1. The need for a temporality model

Time is an essential aspect on the aeronautical information world, where change notifications are usually made well in advance of their effective dates. Aeronautical information systems are requested to store and to provide both the current situation and the future changes. The expired information needs to be archived for legal investigation purposes.

For operational[1] reasons, a distinction is usually made between:

- *permanent changes* (the effect of which will last until the next permanent change or until the end of the lifetime of the feature) and
- *temporary status* (changes of a limited duration that are considered to be overlaid on the permanent state of the feature).

A temporary change includes the concepts of overlay and reversion. The temporary change is overlaid on the permanent feature state. When the temporary change ends, the temporary changes no longer apply and we revert back to the permanent feature state.

Note that, from an operational point of view, "temporary status" also includes the concept of "temporary features". However, from the AIXM point of view, temporary features are in no way different from normal features. The feature is created and withdrawn, just that the life span is shorter than usual.

In order to satisfy the temporal requirements of aeronautical information systems, AIXM must include an exhaustive temporality model, which enables a precise representation of the states and events of aeronautical features. In particular, this shall enable the development and the implementation of digital NOTAM. By digital NOTAM we mean replacing the free text contained in a NOTAM message with structured facts, which enable the automated processing of the information.

A general temporal model should be uniformly applied to all aeronautical feature types and the temporality concept should be abstracted from the task of modeling object properties. At the conceptual level, the model should describe the temporal evolution of the features, as they occur in the real world. This shall be done in compliance with the following rules:

- Completeness - all temporal states must be representable;
- Minimalism - use of minimal number of elements;
- Consistency - no reuse of elements with different meaning;
- Context-free - meaning of (atomic) elements independent of context; no functional dependency of (atomic) elements at the data encoding level;

The data exchange specification shall support the conceptual model. In addition, convenience elements ("views") may be introduced in the data exchange specification in order to facilitate the operations. This means that the data exchange specification may deviate from the "minimalism" rule.

[1] For example, systems that produce printed aeronautical documentation (AIP, charts) tend to ignore temporary status information; only the static data is represented on such printed products.

2. Building the Temporality Model

2.1 (step 1) Time varying properties

There are two levels at which aeronautical feature instances are affected by time:

- Every feature has a start of life and an end of life;
- The properties of a feature can change within the lifetime of the feature; this includes the possibility for a property to not be defined over a time period.

The start of life and the end of life may also be considered as feature properties (attributes). This gives the following high-level list of properties for any AIXM feature:

- a global unique identifier;
- the start of life (date and time);
- the end of life (date and time);
- attributes and associations that qualify, quantify or relate in some form that feature.

It is considered that any feature property may change in time, except for the global unique identifier. This is a key assumption of the AIXM Temporality model.

The first step in the construction of the AIXM temporality model is represented by the diagram below, which shows the values of a feature's properties (P1, P2, … P5) along a timeline.

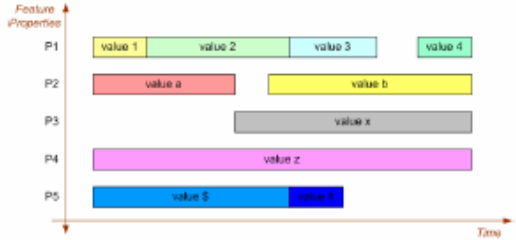


Figure 1

Discussion: ***Can the start of life and the end of life properties of a feature vary in time?***

At first sight, probably not. A feature is created at a moment in time and will cease to exist at another moment in time. But this is true only when considering the already known history of a feature. When exchanging data about the future, there might be situations where the start/end of life is planned to happen at a certain date/time and this date might change.

Therefore, we have to include the start/end of life of a feature in the time varying properties list.

Now on www.aixm.aero

EUROCONTROL

Thank you!

EUROCONTROL