# Requirements from the NAV group - DRAFT

|  |  |
| --- | --- |
| 10 | Events shall contain timing information |

|  |  |
| --- | --- |
| 20 | It shall be possible to express the timing information as either an absolute or a relative time. |

|  |  |
| --- | --- |
| 30 | Absolute time shall consist of:   * an absolute time stamp: date + time. * an optional time scale. |

|  |  |
| --- | --- |
| 35 | The representation of absolute time shall adhere to CCSDS Type A and Type B. |

|  |  |
| --- | --- |
| 40 | Relative time shall consist of:   * an elapsed time expressed as a real number. * an optional time scale. * an optional reference epoch (from which the elapsed time is measured).   Note: respecting the specification in SI units, the elapsed time should be given in seconds. |

|  |  |
| --- | --- |
| 45 | The reference epoch shall be an absolute time (as defined in 30). |

|  |  |
| --- | --- |
| 50 | The optional time scale shall default to UTC or implicitly to the scale defined in the events container.  Note: The events container refers to the element in the data hierarchy that embraces a set of events to be managed as a collection (e.g. an events list, a class EventsCollection, etc.) |

|  |  |
| --- | --- |
| 55 | The optional time scale (in each context) shall override the time scale of the containing context.  Note: this requirement is meant to provide flexibility in the management of events with more than one time scale simultaneously. |

|  |  |
| --- | --- |
| 60 | For absolute timing, the order of priority in the assignment of time scales shall be:   * Own time scale given. * Time scale of the events container. * UTC |

|  |  |
| --- | --- |
| 65 | For relative time, the order of priority in the assignment of time scales shall be:   * Own time scale given. * Time scale of the reference epoch (if given). * Time scale of the events container. * UTC |

|  |  |
| --- | --- |
| 70 | Not needed |

|  |  |
| --- | --- |
| 80 | It shall be possible to collect events in a container (e.g. events list) |

|  |  |
| --- | --- |
| 90 | The definition of the time (absolute or relative) shall be independent of the definition of the events.  Note: the naming of the time types shall then be independent of the containing element, whether an event or any other data type. Hence AbsoluteEventTime 🡪 Evnettime, RelativeEventTime 🡪 EventTime, etc. |

|  |  |
| --- | --- |
| 100 | The definition of the time scales shall be independent of the definition of the events. |

|  |  |
| --- | --- |
| 105 | Each type of event shall be uniquely identified by an event identifier.  Note: event identifiers are in general application domain specific and therefore the event structure shall mandate the appearance of the event type but not the detailed enumeration of event types in the application domain. |

|  |  |
| --- | --- |
| 110 | The event structure shall allow children parameters to qualify the event. |

|  |  |
| --- | --- |
| 120 | Each event type shall have a predefined set of fully qualified parameters (that may be none). |

|  |  |
| --- | --- |
| 130 | The events container shall allow attributes to provide general information for events and default values. |

|  |  |
| --- | --- |
| 135 | The events container shall provide for the definition of a default time scale for the contained events. |

Notes :

* AL: The type of timing (absolute or relative) has no default value because it is not ambiguous whether it is absolute or relative.
* FM: Yet the labels shall be specific identifying the type of time. Sorry, but this is more easily thought in terms of XML structure (ort UML as provided by CSSM) where the information is hierarchically structured.

<epoch>

<absoluteTime>2017-10-02T23:00:00</absoluteTime>

</epoch>

<epoch>

<relativeTime>-5.0</relativeTime>

<!--- Epoch time in different time scale (use case?) -->

<epochTime scale="UTC">2017-10-02T23:00:00</epochTime>

</epoch>

* AL: One question is whether it shall be possible to possibly have 2 time scales for relative timing: one associated with the (relative) time value, one with the reference epoch. Isn't it too complex ?
* FM: I think that the scheme is flexible enough to allow for this, although we may not be able to find a realistic use case.
* AL: One may want to define event time relative to other events: similar to 40-a or 40-b with the reference epoch being defined by another events. But this may be too complex.
* FM: This can be done assigning to the events unique identifiers. It is a common practice in other technical areas, again something easy to implement in XML and not so much in KVN (PRM uses this feature). Note that this to some extent violates 90.

<event event="second" type="ordinal" UID="second\_event\_id">

<epoch>

<absoluteTime>2017-10-02T23:00:00</absoluteTime>

</epoch>

</event>

<event event="sixth" type="ordinal">

<epoch>

<relativeTime>-5.0</relativeTime>

<!--- Epoch time in different time scale (use case?) -->

<eventTime ref="second\_event\_id" />

</epoch>

</event>

* FM: Ordering of events. This is always possible with absolute timing. I dare say that it is always possible with relative time in one time scale. If the complexity increases (many time scales, absolute and relative mixed and times relative to other event times) it may rather complex. Mind also circular relationships!

# Additional remarks on events structure

(structure described in the draft Planning Information Book)

Additional remarks on the

|  |  |
| --- | --- |
| Table A-1 - parameter "event" | AL: Because the event is also the class itself, calling this field "event" is confusing.  eventName seems to be a better name.  FM: actually, encapsulation rules would recommend to remove the reference to the container, hence eventName should be just name as it is already contained in event and can only be the name of the event. Analogously we could go for eventType 🡪 type, eventIdentifier 🡪 identifier, … |
| Table A-1 - parameter eventTimeLatestOffset and eventTimeEarliestOffset | AL: For the events structure to be used in a greater number of applications, why not make this parameter a real number (for applications that need a smaller timing resolution)  FM: if expressed in second probably the granularity I enough at least in the flight dynamics world. I think. |