

Change ID: 5.1-35

Schedules and Temporality - concept review

Summary

It is proposed to introduce a new "PropertiesWithSchedule" concept in order to replace the Timetable/Timesheet classes inherited from earlier AIXM versions. This change will put in agreement the "schedules" modelling with the new Temporality Concept of AIXM 5. It will also indicate clearly which feature attributes are actually concerned with schedules.

Background

AIXM 4.5

AIXM version 3.3 and 4.5 contained many entities that had an associated Timetable with optional Timesheet(s). For example:

- "describing operational hours for one and only one AD_HP"
- "describing operational hours for one and only one NDB"
- "describing activity hours for one and only one AIRSPACE"

AIXM 5

A complete "Temporality Concept" has been included in AIXM 5.0, which allows us to describe the states and the events characterising or affecting an aeronautical information feature during its lifetime. This is done through Feature TimeSlices.

AIXM 5.0 has also inherited from the earlier versions the [Timetable](#) class. It is almost identical to the one of AIXM 4.5 and it is associated with basically the same features as it was in AIXM 4.5.

Rationale for the change

In the earlier AIXM versions, the Timetable/Timesheet construct was used to compensate for the lack of a real temporality mechanism. It provides a simple means to indicate when a feature is, for example, active/operational - if according to a regular schedule.

The introduction of the Temporality Concept in AIXM 5.0 raises the question whether the former Timetable/Timesheet are still needed in the model. To answer this question, it is necessary to analyse the typical cases where an association with Timetable exists. Two typical situations have been identified:

Usage cases for Timetable

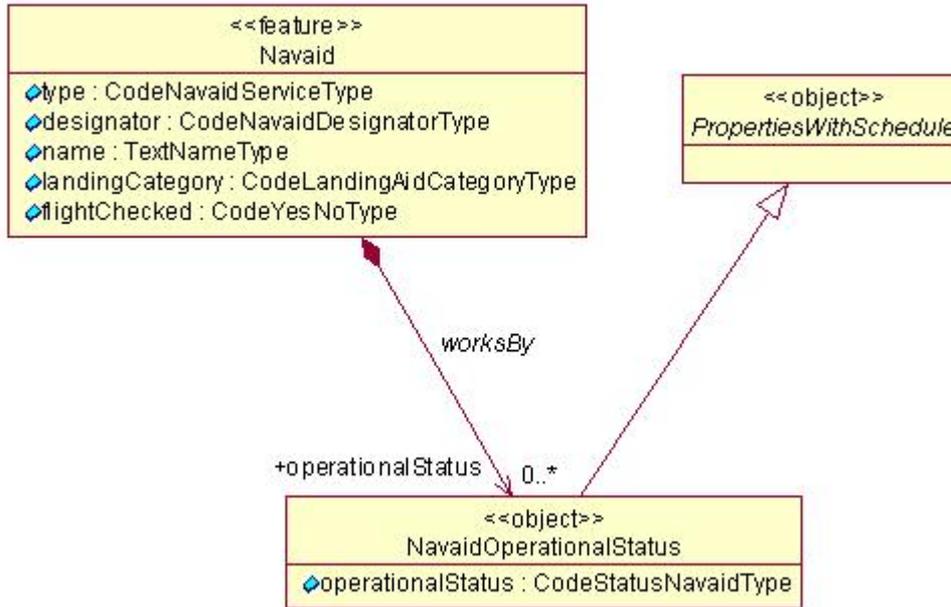
Case 1 - existing attribute(s) affected by schedule

The [Navaid](#) feature is one such case. A Navaid has an association with Timetable that indicates "serviceHours". The "service hours" are in fact related to the operationalStatus property of the Navaid. A navaid could be:

- operationalStatus = "OPERATIONAL" permanently or according to a certain schedule;
- operationalStatus = "UNSERVICEABLE" for a limited time period or according to a certain schedule or both;

- operationalStatus = "ONTEST" for a limited time period (typically it does not require a schedule).

In consequence, in the Navaid case, it's logical to group together the "operationalStatus" and the schedule. This suggests a separate "NavaidOperationalStatus" class, associated with the Navaid and derived from an abstract **PropertiesWithSchedule class** (a simplified Timetable), as it can be seen in the diagram below.



This construct has the following advantages:

- It makes explicit that the schedule concerns only the operationalStatus attribute. In similar cases, where more attributes are concerned at once, they can be grouped together in a single association class;
- It makes it possible to completely and explicitly describe the operationalStatus of the Navaid, at all times; that's because the association with the new NavaidOperationalStatus can have a multiplicity higher than one. This is really needed, because the Navaid could be, for example, "operational day time" and "unserviceable at night times". The single occurring operationalStatus property of the current model does not support such complex situations.

Case 2 - implicit attribute(s) affected by schedule

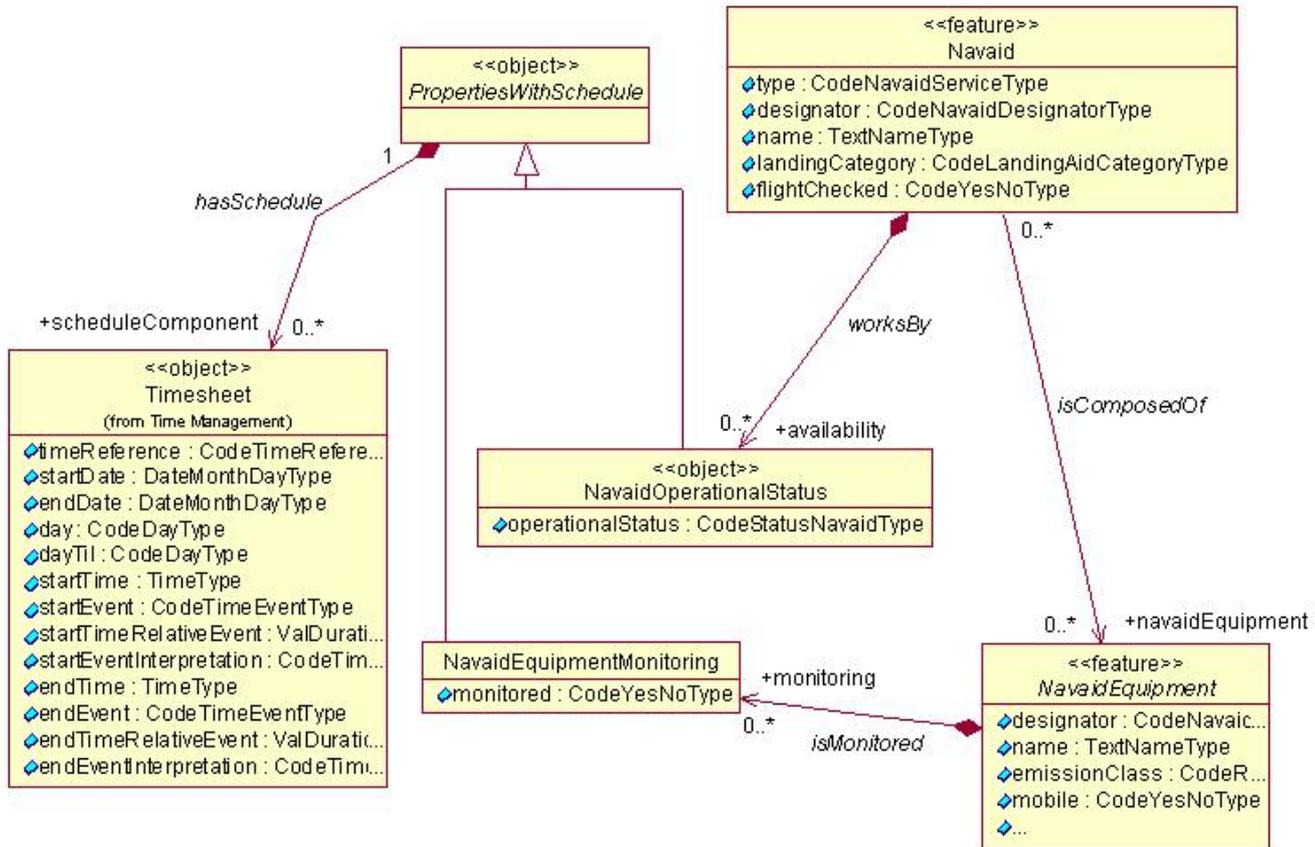
The [NavaidEquipment](#) abstract feature has an association with Timetable that indicates "unmonitoredHours". This association implies that the equipment can be monitored or unmonitored at certain times, but there is no explicit attribute for that property.

In this case, the solution would be to introduce an explicit property of the NavaidEquipment that would indicate whether the equipment is monitored or not and at which times. By the pattern proposed at Case 1, this will be located in a separate "NavaidEquipmentMonitoring" class.

Temporality Concept perspective

The analysis made above was bottom-up, starting from the existing classes and attributes that are affected by a schedule. Doing the same analysis top-down, from the AIXM Temporality concept, brings one similar conclusion. As the concept works at "feature level" (through TimeSlices), as soon as a property has its own cyclic variation during the validity time of the Timeslice, it needs to be isolated in a separate class. This justifies the "PropertiesWithSchedule" class proposed for Case 1.

The diagram below shows the resulting model for Navaid and NavaidEquipment, using the new PropertyWithSchedule concept:



Is there **any alternative to introducing the "properties with schedule" concept?** Another solution could be to include "schedules" in the TimeSlice concept and make a schedule usable for any feature. That would have two disadvantages:

- if an attribute, such as the value of a declared distance, has one value during day and another value during night, each of the two values would need to be part of a different Baseline. Each of the two Baseline would have a schedule that would indicate when they are applicable. But the two Baseline would have overlapping validity times. This would significantly complicate the Temporality concept of AIXM;
- the analysis shows that, frequently, schedules really concern just one or two attributes. Having the schedule at the level of the feature would cause this aspect to be lost.

Therefore, the introduction of the attribute with schedule concept seems to be most convenient approach.

The Timetable and Timesheet classes inherited from AIXM 4.5 also need to be adapted in order to better match the needs of the AIXM 5 model. Further changes are discussed below.

Coded 'working hours' - no longer necessary

The current Timetable class has an "workingHours" attribute that allows for certain schedules to be expressed in a compact form, such as HJ, HN, etc.

- The value 'H24' was used for static data to indicate 'continuously active'. This value is no longer needed in AIXM 5 because it does not represent a type of schedule. Even more, using a 'H24' could be confusing because the TimeSlice itself could be valid for less than 24 hours.
- The values HJ (day time) and HN (night time) can be expressed as Timeheet, using SR (sunrise) or SS (sunset) plus/minus a number of minutes. This would also make the HJ and HN meanings more precise. Therefore, the HJ and HN values are not really necessary in AIXM 5.1.

- The values HX (no specific working hours) and HO (will meet operational needs) are not real schedules, but explanations for why a schedule cannot be provided. They should be recorded as Notes and/or NilReason.
- The value 'NOTAM' indicates that the actual activity time is published by a NOTAM message. More general, it means a prior notification, some time before the effective activation time. But this is not really a schedule. It is better represented as either a Note or a specific value of the property affected by the schedule. For example, an Airspace that is activated by NOTAM would have as baseline status "available for activation". If this activation is expected to take place only at certain times, then the status "available for activation" can be combined with a true schedule, specifically indicating those times. It is therefore proposed to also eliminate the "NOTAM" value from the list of coded working hours.

In conclusion, the workingHours attribute of the Timetable is not needed for the new PropertiesWithSchedule class proposed for AIXM 5.1.

Other changes to Timesheet

A number of other changes to the Timesheet class will be made in a separate change proposal, which will also take into consideration the content of this one.

New properties with schedule

The digital NOTAM Trials have allowed us to identify some additional features that need schedules to be associated with some of their properties. These are described in a separate table at the end of the change proposal.

Change proposal details

Transform Timetable/Timesheet into PropertiesWithSchedule

Create a new sub-package "Schedules" under the Shared package and move into this package the following classes:

- Timetable;
- Timesheet;
- SpecialDate (including its association with OrganisationAuthority and the association between OrganisationAuthority and Timetable)

from the TimeManagement package. Delete the package TimeManagement.

Modify the Timetable class as follows:

- rename it into "PropertiesWithSchedule";
- declare it "abstract";
- change the definition into " *An abstract class that is used for associating a group of feature properties with a repetitive time period, occurring cyclically inside the validity of the Feature Timeslice, during which these properties have specific values*";
- remove the attribute workingHours.

Delete the data type CodeTimetableType

Modify the Timesheet class as follows:

- change the definition into " *The definition of a single time interval, that occurs repeatedly as part of a schedule.*";

Modify associations with Timetable

Based on the Cases identified above, the following changes are proposed for each individual association with Timetable that exists currently in AIXM 5.0.

AIXM Workarea - Schedules and Temporality - concept review

Feature / Object	Timetable association	Discussion	Proposal
RunwayDeclaredDistance	isValidBy	Case 1. The schedule applies to the distance (and distanceAccuracy) values	Move the distance and distanceAccuracy attributes in an associated RunwayDeclaredDistanceValue class(0..*, role="declaredValue"), derived from the PropertiesWithSchedule class. Remove the association with Timetable.
AirportHeliport UsageCondition	hasApplicableTimes	Case 1. The schedule applies to all the properties of the AirportHeliportUsageCondition, including the aggregated classes, such as AirportHeliportOperation	Make AirportHeliportUsageCondition a specialisation of the new PropertiesWithSchedule class. Remove association with Timetable.
AltimeterSource	isAvailableOn	Case 2. There is an implicit "operationalStatus" attribute behind the association with Timetable	Insert an associated class AltimeterSourceStatus (0..*, role = "availability"), derived from the PropertiesWithSchedule class. Insert an operationalStatus attribute in this class, data type CodeStatusOperationsType. Remove the association with Timetable.
Airspace	isActive	Case 1. The schedule should be associated with the "statusActivation" of the AirspaceLayerUsage.	Remove the association with Timetable. Note that a separate change proposal is being developed for the review of the AirspaceUsage concept.
ContactInformation	isAvailable	Case 1. The schedule applies to all the specialisations of the ContactInformation class.	Make PostAddress, OnlineContact and Telephone specialisations of the new PropertiesWithSchedule class. Remove association with Timetable. (Note that there exists another change proposal that will remove the inheritance between ContactInformation and PostAddress, OnlineContact and TelephoneContact. Therefore, there is no risk of double inheritance, which we try to avoid in AIXM 5).
LayerAndTime	isActiveBy	Case 1. The schedule refers to the whole LayerAndTime class.	Make LayerAndTime a specialisation of the new PropertiesWithSchedule class. Remove the association with Timetable. Note: please also read change description 5.1-41 (Harmonisation of Usage classes), which will further affect this change.
Service	isAvailableBy	Case 1. The schedule applies to operationalStatus attribute.	Move attribute operationalStatus into a separate class "ServiceOperationalStatus"(0..*, role = "availability"), derived from the the PropertiesWithSchedule class. Remove the association with Timetable.
RadioCommunication Channel	isAvailableBy	Case 1. The schedule applies to operationalStatus attribute.	Move the attribute operationalStatus into a separate class "RadioCommunicationOperationalStatus" (0..*, role = "availability"),

AIXM Workarea - Schedules and Temporality - concept review

Navaid	worksBy	Case 1. The schedule applies to the operationalStatus attribute	derived from the the PropertiesWithSchedule class. Remove the association with Timetable. Move the attribute operationalStatus into a separate class "NavaidOperationalStatus" (0..*, role = "availability"), derived from the the PropertiesWithSchedule class. Remove the association with Timetable.
NavaidEquipment	isUnmonitoredBy	Case 2. The schedule refers to an implicit "monitoringStatus" attribute.	Insert an associated class NavaidEquipmentMonitoring (0..*, role = "monitoring"), derived from the PropertiesWithSchedule class. Insert a "monitored" attribute in this class, data type CodeYesNoType. Remove the association with Timetable.
SpecialNavigationStation	isActiveBy	Case 2. There is an implicit "operationalStatus" attribute behind the association with Timetable	Insert an associated class SpecialNavigationStationStatus (0..*, role = "availability"), derived from the PropertiesWithSchedule class. Insert an operationalStatus attribute in this class, data type CodeStatusNavaidType. Remove the association with Timetable.
Unit	worksBy	Case 2. The schedule applies to an implicit "availability" attribute.	Insert an associated class UnitAvailability (0..*, role = "availability"), derived from the PropertiesWithSchedule class. Insert an operationalStatus attribute in this class, data type CodeStatusOperationsType. Remove the association with Timetable.
VerticalStructureLighting	isEffective	Case 1. The schedule applies to the status attribute of the vertical structure lighting	Move the attribute status into a separate class "VerticalStructureLightingStatus" (0..*, role = "availability"), derived from PropertiesWithSchedule. Remove the association with Timetable.
VerticalStructure	isActiveBy	Case 1. A schedule for a vertical structure is used when it changes cyclically its position or size. Therefore, the schedule applies to the VerticalStructurePart, which contains the geometry of the structure.	Make VerticalStructurePart a specialisation of PropertiesWithSchedule. Remove the association with Timetable.
CirclingRestriction	isEffective	Case 1. The schedule applies to the whole class.	Make CirclingRestriction a specialisation of PropertiesWithSchedule. Remove the association with Timetable.
ProcedureUsage	operatesAccordingTo	Case 1. The schedule applies to the attribute usageStatus.	Move the attribute usageStatus into a separate class "ProcedureAvailability" (0..*, role = "availability"), derived from the the PropertiesWithSchedule class. Remove the association with Timetable.
FlightRestriction	isApplicableBy	Case 1. The schedule is part of the condition.	Make FlightConditionCombination

a specialisation of the PropertiesWithSchedule class. Remove the association with Timetable.

Additional features that have properties with schedules

For the classes listed below, it is proposed to associate a schedule with one or more of their properties.

Feature / Object	Discussion	Proposal
AirportHelicopterResponsibleOrganisation (see Change 5.1-3)	Annex 15 requires the publication of the operational hours of the aerodrome administration. In addition, in some Countries, it is possible for the operating authority to change based on the weekly schedule - military during the week, a civil air club during the weekend.	Make the new class AirportHelicopterResponsibleOrganisation (see Change 5.1-3) a specialisation of the PropertiesWithSchedule class.
WorkArea	An airport/heliport work area can be active according to a specific schedule	Insert a new class "WorkareaActivity" (0..*, role = "activation"), derived from the the PropertiesWithSchedule class. Insert a new attribute "isActive" in this class, data type CodeYesNoType.
GroundLightingSystem	This class has an operationalStatus attribute, which might have a schedule	Move the attribute operationalStatus into a separate class "GroundLightingAvailability" (0..*, role = "availability"), derived from the the PropertiesWithSchedule class.
LightElement	This class has a status attribute, which might have a schedule	Move the attribute status into a separate class "LightElementAvailability" (0..*, role = "availability"), derived from the the PropertiesWithSchedule class.
NavaidEquipment	This class has an operationaStatus attribute, which might have a schedule	Delete the operationalStatus attribute. Insert an association with the newly create "NavaidOperationalStatus" class (0..*, role = "availability"), which will provide both the operationalStatus attribute and the schedule.