European Network of
Transmission System Operators
for Electricity

# STATE INSTRUCTION SCHEDULE PROFILE SPECIFICATION

2024-10-16

ICTC APPROVED
VERSION 2.3.1

# Copyright notice:



**This document is maintained by the ENTSO-E CIM WG. Comments or remarks are to be provided at cim@entsoe.eu**

**NOTE CONCERNING WORDING USED IN THIS DOCUMENT**

The force of the following words is modified by the requirement level of the document in which they are used.

- SHALL: This word, or the terms “REQUIRED” or “MUST”, means that the definition is an absolute requirement of the specification.
- SHALL NOT: This phrase, or the phrase “MUST NOT”, means that the definition is an absolute prohibition of the specification.
- SHOULD: This word, or the adjective “RECOMMENDED”, means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
- SHOULD NOT: This phrase, or the phrase “NOT RECOMMENDED”, means that there may exist valid reasons in particular circumstances when the particular behaviour is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behaviour described with this label.
- MAY: This word, or the adjective “OPTIONAL”, means that an item is truly optional.

33

# Revision History

| Version | Date | Paragraph | Comments |
|---|---|---|---|
| 2.2.0 | 2023-03-24 | | For review. |
| 2.2.0 | 2023-04-20 | | For ICTC approval. |
| 2.3.0-alpha | 2024-02-17 | | For internal review. |
| 2.3.0-beta | 2024-03-20 | | For CIM WG review. |
| 2.3.1-alpha | 2024-09-07 | | For CIM WG review. |

# CONTENTS

**List of figures**

## List of tables

# 1 Introduction

The state instruction schedule enables an exchange of additional information related to schedules like bid schedule, GLKS schedule or availability schedule of Assessed elements, RAs or Contingencies amongst others.

# 2 Application profile specification

## 2.1 Version information

The content is generated from UML model file CIM17-2_CGMES31v01_PROF-20v02_NC23v65_MS10v01_DES10v01.eap.

This edition is based on the IEC 61970 UML version 'IEC61970CIM17v40', dated '2020-08-24'.

- Title: State instruction schedule vocabulary
- Keyword: SIS
- Description: This vocabulary is describing the state instruction schedule profile.
- Version IRI: https://ap-voc.cim4.eu/StateInstructionSchedule/2.3
- Version info: 2.3.1
- Prior version: http://entsoe.eu/ns/CIM/StateInstructionSchedule-EU/2.2
- Conforms to: urn:iso:std:iec:61970-600-2:ed-1|urn:iso:std:iec:61970-301:ed-7:amd1|file://iec61970cim17v40_iec61968cim13v13a_iec62325cim03v17a.eap|urn:iso:std:iec:61970-401:draft:ed-1|urn:iso:std:iec:61970-501:draft:ed-2|file://CIM100_CGMES31v01_501-20v02_NC23v62_MM10v01.eap
- Identifier: urn:uuid:af884936-ea95-416b-b4c9-1214caa68658

## 2.2 Constraints naming convention

The naming of the rules shall not be used for machine processing. The rule names are just a string. The naming convention of the constraints is as follows.

"{rule.Type}:{rule.Standard}:{rule.Profile}:{rule.Property}:{rule.Name}"

where

rule.Type: C – for constraint; R – for requirement

rule.Standard: the number of the standard e.g. 301 for 61970-301, 456 for 61970-456, 13 for 61968-13. 61970-600 specific constraints refer to 600 although they are related to one or combination of the 61970-450 series profiles. For NC profiles, NC is used.

rule.Profile: the abbreviation of the profile, e.g. TP for Topology profile. If set to "ALL" the constraint is applicable to all IEC 61970-600 profiles.

rule.Property: for UML classes, the name of the class, for attributes and associations, the name of the class and attribute or association end, e.g. EnergyConsumer, IdentifiedObject.name, etc. If set to "NA" the property is not applicable to a specific UML element.

rule.Name: the name of the rule. It is unique for the same property.

Example: C:600:ALL:IdentifiedObject.name:stringLength

## 2.3 Profile constraints

This clause defines requirements and constraints that shall be fulfilled by applications that conform to this document.

This document is the master for rules and constraints tagged "NC". For the sake of self-containment, the list below also includes a copy of the relevant rules from IEC 61970-452, tagged "452".

- C:452:ALL:NA:datatypes

  According to 61970-501, datatypes are not exchanged in the instance data. The UnitMultiplier is 1 in cases none value is specified in the profile.

- R:452:ALL:NA:exchange

  Optional and required attributes and associations must be imported and exported if they are in the model file prior to import.

- R:452:ALL:NA:exchange1

  If an optional attribute does not exist in the imported file, it does not have to be exported in case exactly the same data set is exported, i.e. the tool is not obliged to automatically provide this attribute. If the export is resulting from an action by the user performed after the import, e.g. data processing or model update the export can contain optional attributes.

- R:452:ALL:NA:exchange2

  In most of the profiles the selection of optional and required attributes is made so as to ensure a minimum set of required attributes without which the exchange does not fulfil its basic purpose. Business processes governing different exchanges can require mandatory exchange of certain optional attributes or associations. Optional and required attributes and associations shall therefore be supported by applications which claim conformance with certain functionalities of the IEC 61970-452. This provides flexibility for the business processes to adapt to different business requirements and base the exchanges on IEC 61970-452 compliant applications.

- R:452:ALL:NA:exchange3

  An exporter may, at his or her discretion, produce a serialization containing additional class data described by the CIM Schema but not required by this document provided these data adhere to the conventions established in Clause 5.

- R:452:ALL:NA:exchange4

  From the standpoint of the model import used by a data recipient, the document describes a subset of the CIM that importing software shall be able to interpret in order to import exported models. Data providers are free to exceed the minimum requirements described herein as long as their resulting data files are compliant with the CIM Schema and the conventions established in Clause 5. The document, therefore, describes additional classes and class data that, although not required, exporters will, in all likelihood, choose to include in their data files. The additional classes and data are labelled as required (cardinality 1..1) or as optional (cardinality 0..1) to distinguish them from their required counterparts. Please note, however, that data importers could potentially receive data containing instances of any and all classes described by the CIM Schema.

- R:452:ALL:NA:cardinality

The cardinality defined in the CIM model shall be followed, unless a more restrictive cardinality is explicitly defined in this document. For instance, the cardinality on the association between VoltageLevel and BaseVoltage indicates that a VoltageLevel shall be associated with one and only one BaseVoltage, but a BaseVoltage can be associated with zero to many VoltageLevels.

- R:452:ALL:NA:associations

Associations between classes referenced in this document and classes not referenced here are not required regardless of cardinality.

- R:452:ALL:IdentifiedObject.name:rule

The attribute “name” inherited by many classes from the abstract class IdentifiedObject is not required to be unique. It must be a human readable identifier without additional embedded information that would need to be parsed. The attribute is used for purposes such as User Interface and data exchange debugging. The MRID defined in the data exchange format is the only unique and persistent identifier used for this data exchange. The attribute IdentifiedObject.name is, however, always required for CoreEquipment profile and Short Circuit profile.

- R:452:ALL:IdentifiedObject.description:rule

The attribute “description” inherited by many classes from the abstract class IdentifiedObject must contain human readable text without additional embedded information that would need to be parsed.

- R:452:ALL:NA:uniqueIdentifier

All IdentifiedObject-s shall have a persistent and globally unique identifier (Master Resource Identifier - mRID).

- R:452:ALL:NA:unitMultiplier

For exchange of attributes defined using CIM Data Types (ActivePower, Susceptance, etc.) a unit multiplier of 1 is used if the UnitMultiplier specified in this document is “none”.

- C:452:ALL:IdentifiedObject.name:stringLength

The string IdentifiedObject.name has a maximum of 128 characters.

- C:452:ALL:IdentifiedObject.description:stringLength

The string IdentifiedObject.description is maximum 256 characters.

- C:452:ALL:NA:float

An attribute that is defined as float (e.g. has a type Float or a type which is a Datatype with .value attribute of type Float) shall support ISO/IEC 60559:2020 for floating-point arithmetic using single precision floating point. A single precision float supports 7 significant digits where the significant digits are described as an integer, or a decimal number with 6 decimal digits. Two float values are equal when the significant with 7 digits are identical, e.g. 1234567 is equal 1.234567E6 and so are 1.2345678 and 1.234567E0.

- R:NC:ALL:NA:serialization

The profiles are defined in the EnterpriseArchitect application and have multiple artifacts that describe them. The main artifacts are:

1) the EAP file (EnterpriseArchitect project file),
2) the profiles' specification document and
3) the application profiles (RDFS and SHACL).

Due to the complexity of the profiles, there are various cross profile associations that, from profiling and profile maintenance point of view, it is not practical to include the complete inheritance structure in all profiles. If this is done the documentation provided for all profiles would also include duplicated information on the description of classes defined in other profiles. The following cases are often observed in profiles:

- Case 1: An association end refers to an abstract class
- Case 2: An abstract class (stereotyped with "Description") has an association (direction to another class)
- Case 3: An abstract class (not stereotyped with "Description") has an association (direction to another class)
- Case 4: An abstract class has attributes and subclasses are not in the profile

In all cases, the datasets shall only include the subtypes of the abstract classes with the related properties (i.e. association or attributes) defined in the profile. The information is taken from either canonical model or the profiles where complete (expected) inheritance structure for the related abstract class is described. SHACL based constraints include constraints only for the concrete classes that are subtypes of the abstract class in the profile, and this can be used to inform which are the concrete classes expected in a dataset that conforms to this profile.

It should be taken into account that this approach deviates from MVAL5 (IEC 61970-600-1:2021), which creates multiple inheritance at serialization. For instance, with this more explicit exchange the serialization of the association between abstract class Equipment and abstract class Circuit for a PowerTransformer will be serialized as follows:

- for association

```
<cim:PowerTransformer rdf:about="_c328f787-bc17-47ad-a59f-6ba7133340d0">
  <nc:Equipment.Circut rdf:resource="#_9ced16ac-d076-4ef9-a241-a998a579e77b"/>
</cim:PowerTransformer>
```

- for attribute

```
<cim:ACLineSegment rdf:about="_04f681aa-6999-4fb3-9775-acaa5eb7ceff">
  <cim:Equipment.inService>true</cim:Equipment.inService>
</cim:ACLineSegment>
```

The usage of rdf:ID or rdf:about depends on the stereotype of the class. rdf:about is used if the class has the stereotype "Description".

An example of not allowed serialization, as the Equipment is an abstract class

```
<cim:Equipment rdf:about="_c328f787-bc17-47ad-a59f-6ba7133340d0">
  <nc:Equipment.Circut rdf:resource="#_9ced16ac-d076-4ef9-a241-a998a579e77b"/>
</cim:Equipment>
```

- R:NC:SIS:ParticipationFactorTimePoint.atTime:cardinality
  The attribute ParticipationFactorTimePoint.atTime is optional because

PowerShiftKeySchedule can be used as a part of a bid pattern. In that case, the schedules are following the bid time.

- C:NC:SIS:PowerShiftKeySchedule.ParticipationFactorTimePoint:dependency

  The multiplicity of the association end PowerShiftKeySchedule.ParticipationFactorTimePoint shall be 1..n for all cases except if the PowerShiftKeySchedule is associated with PowerShiftKeyDistribution.

- C:NC:SIS:PowerShiftKeySchedule:associations
  PowerShiftKeySchedule shall include one and only one association to ScheduleResource, PowerElectronicUnit, GeneratingUnit, DCPole, EnergyGroup, EquivalentInjection, EnergyConsumer, EnergySource, ExternalNetworkInjection and HydroPump.

- C:NC:SIS:PowerShiftKeyDistribution:associations
  PowerShiftKeyDistribution shall include one and only one association to either PowerBidSchedule or PowerShiftKeyStrategy.

- C:NC:SIS:GenericSequenceSchedule:associations
  Association end GenericSequenceSchedule.EnergyBlockOrder is required.

## 2.4 Metadata

ENTSO-E agreed to extend the header and metadata definitions by IEC 61970-552 Ed2. This new header definitions rely on W3C recommendations which are used worldwide and are positively recognised by the European Commission. The new definitions of the header mainly use Provenance ontology (PROV-O), Time Ontology and Data Catalog Vocabulary (DCAT). The global new header applicable for this profile is included in the metadata and document header specification document.

The header vocabulary contains all attributes defined in IEC 61970-552. This is done only for the purpose of having one vocabulary for header and to ensure transition for data exchanges that are using IEC 61970-552:2016 header. This profile does not use IEC 61970-552:2016 header attributes and relies only on the extended attributes.

### 2.4.1 Constraints

The identification of the constraints related to the metadata follows the same convention for naming of the constraints as for profile constraints.

- R:NC:ALL:wasAttributedTo:usage

  The prov:wasAttributedTo should normally be the “X” EIC code of the actor or their URI (prov:Agent).

### 2.4.2 Reference metadata

The header defined for this profile requires availability of a set of reference metadata. For instance, the attribute prov:wasGeneratedBy requires a reference to an activity which produced the model or the related process. The activities are defined as reference metadata and their identifiers are referenced from the header to enable the receiving entity to retrieve the “static” (reference) information that is not modified frequently. This approach imposes a requirement that both the sending entity and the receiving entity have access to a unique version of the

reference metadata. Therefore, each business process shall define which reference metadata is used and where it is located.

# 3 Detailed Profile Specification

## 3.1 General

This package contains the state instruction schedule profile.

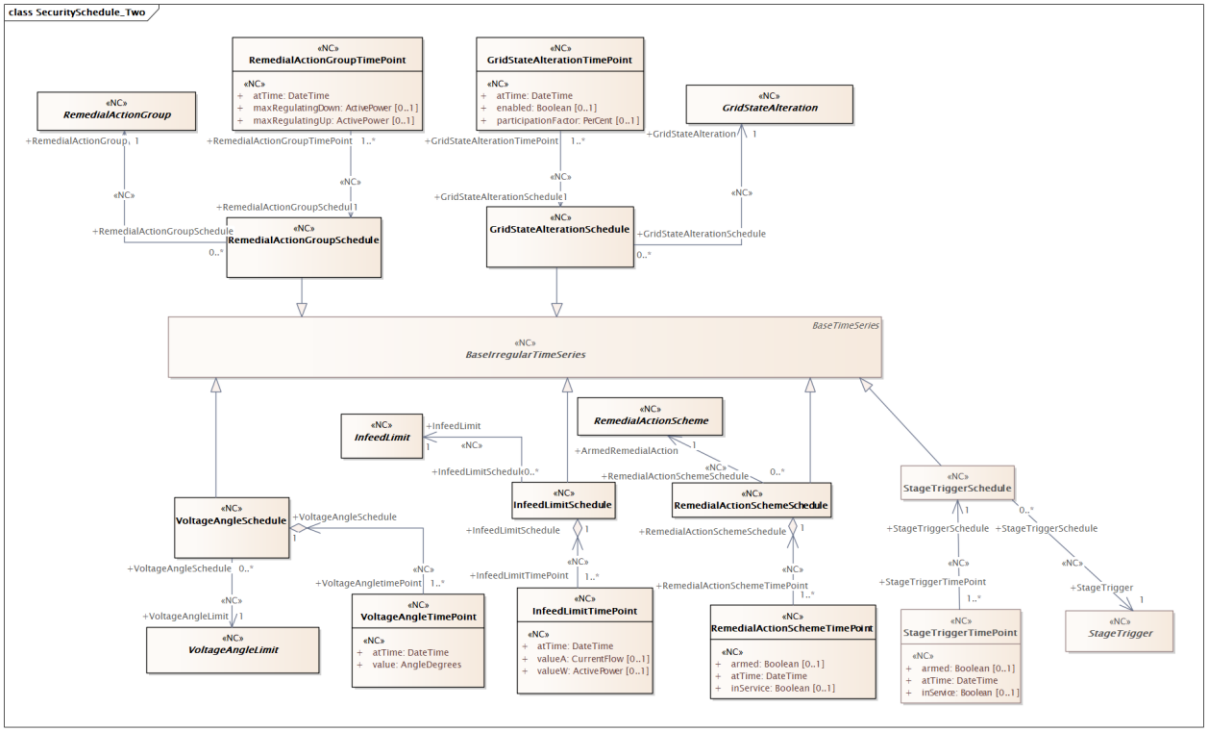



**Figure 1 – Class diagram StateInstructionScheduleProfile::SecuritySchedule_Two**

Figure 1: The diagram shows security schedule related classes.

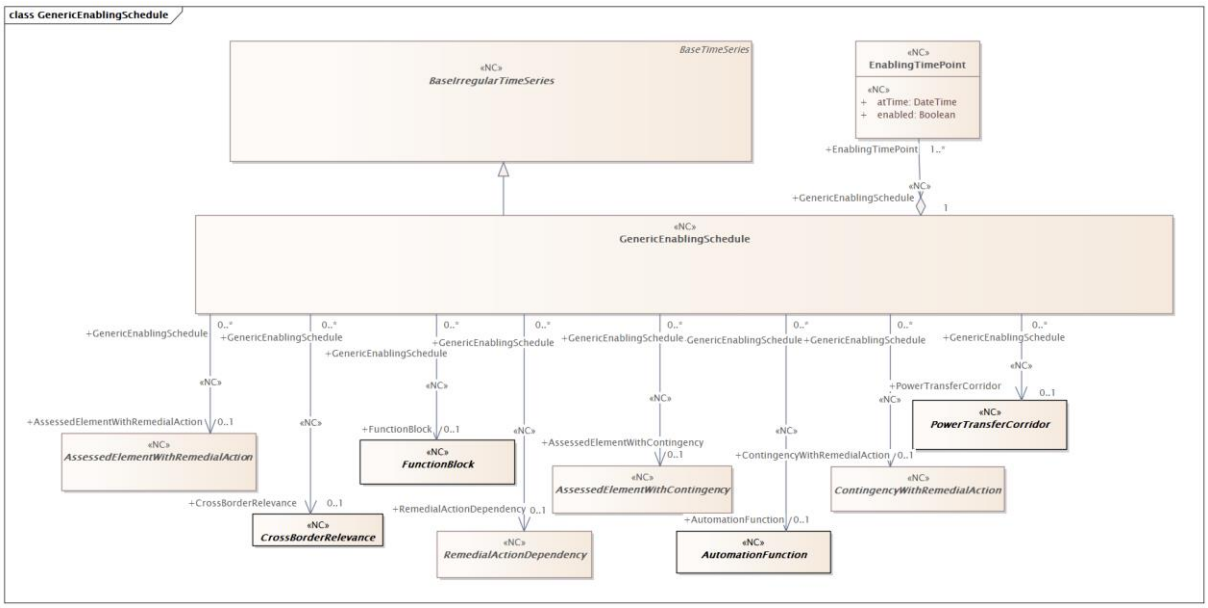
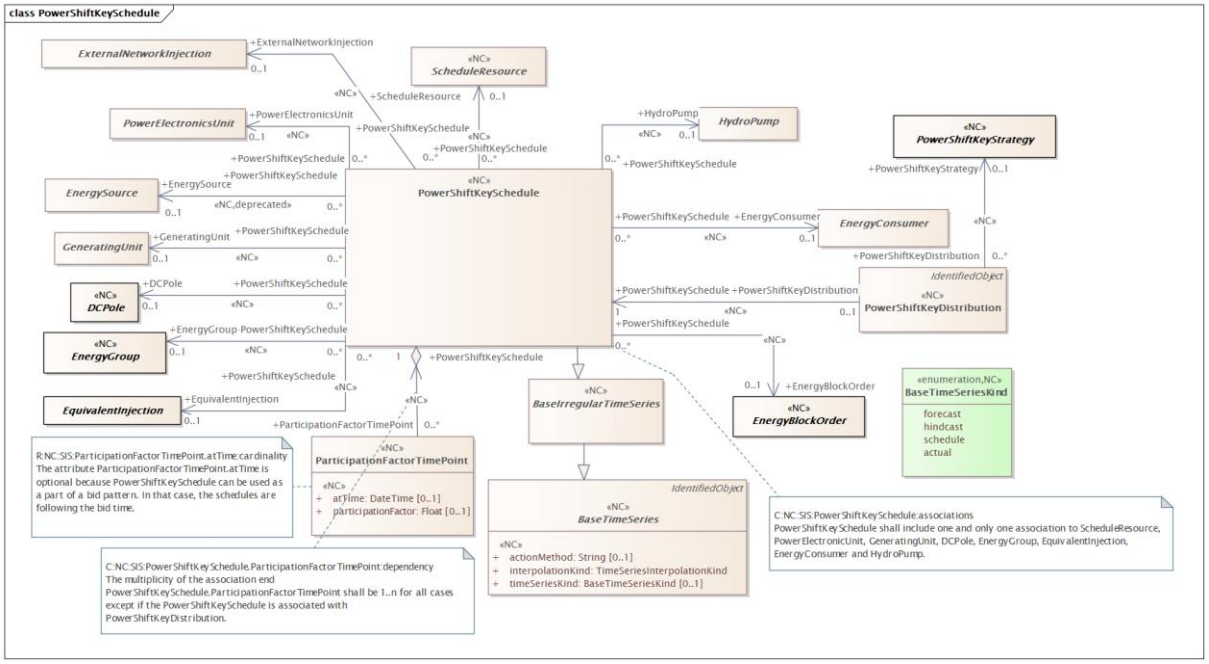



**Figure 2 – Class diagram StateInstructionScheduleProfile::GenericEnablingSchedule**

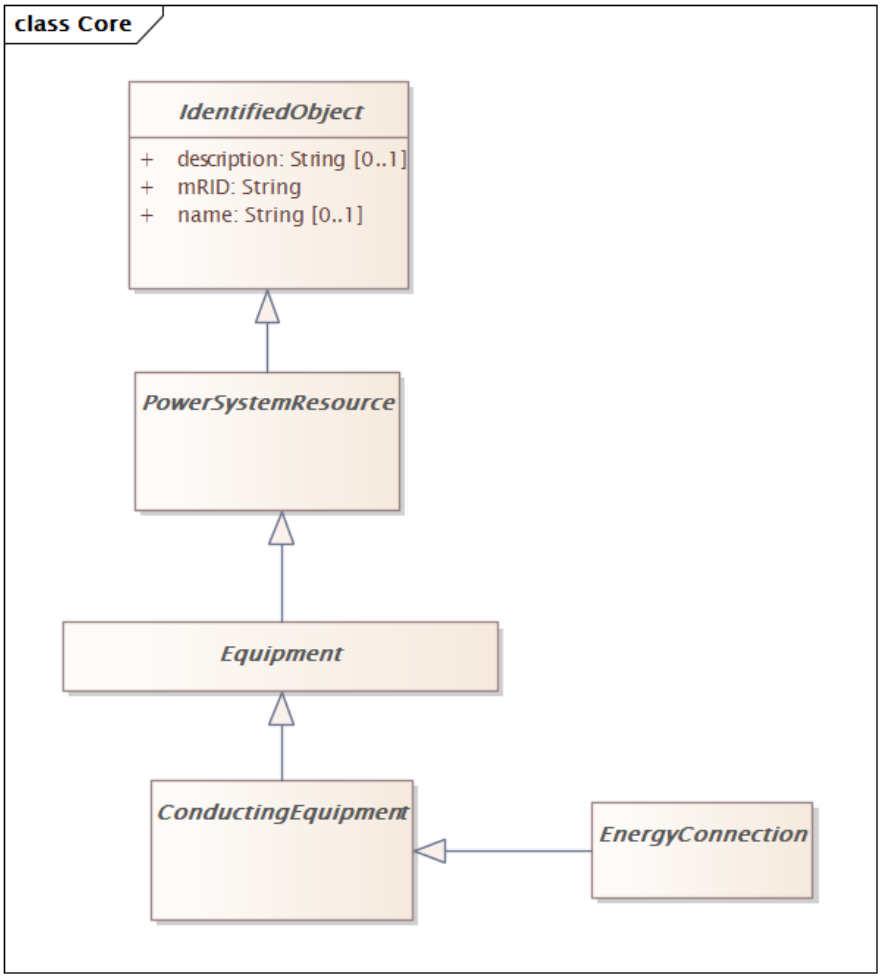
546 Figure 2: The diagram shows generic enabling schedule related classes.




547

548 **Figure 3 – Class diagram StateInstructionScheduleProfile::PowerShiftKeySchedule**

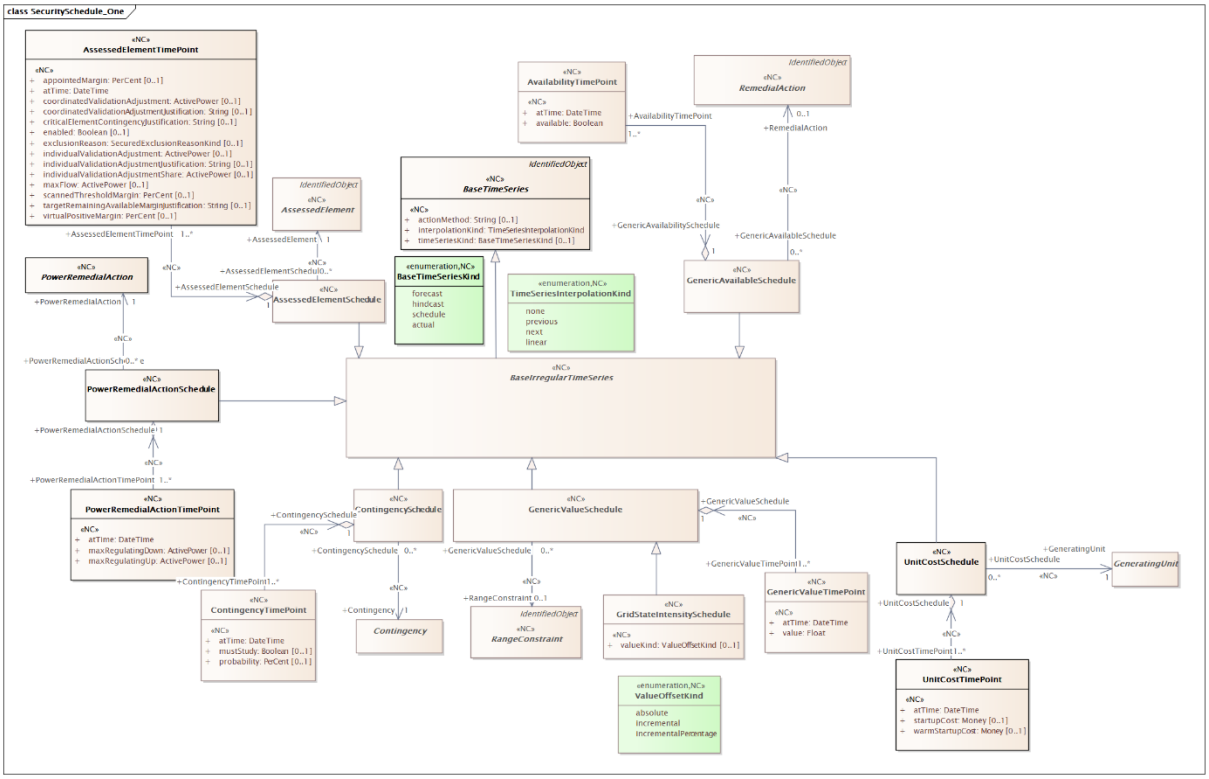
549 Figure 3: The diagram shows power shift keys related classes.




550

551 **Figure 4 – Class diagram StateInstructionScheduleProfile::Core**

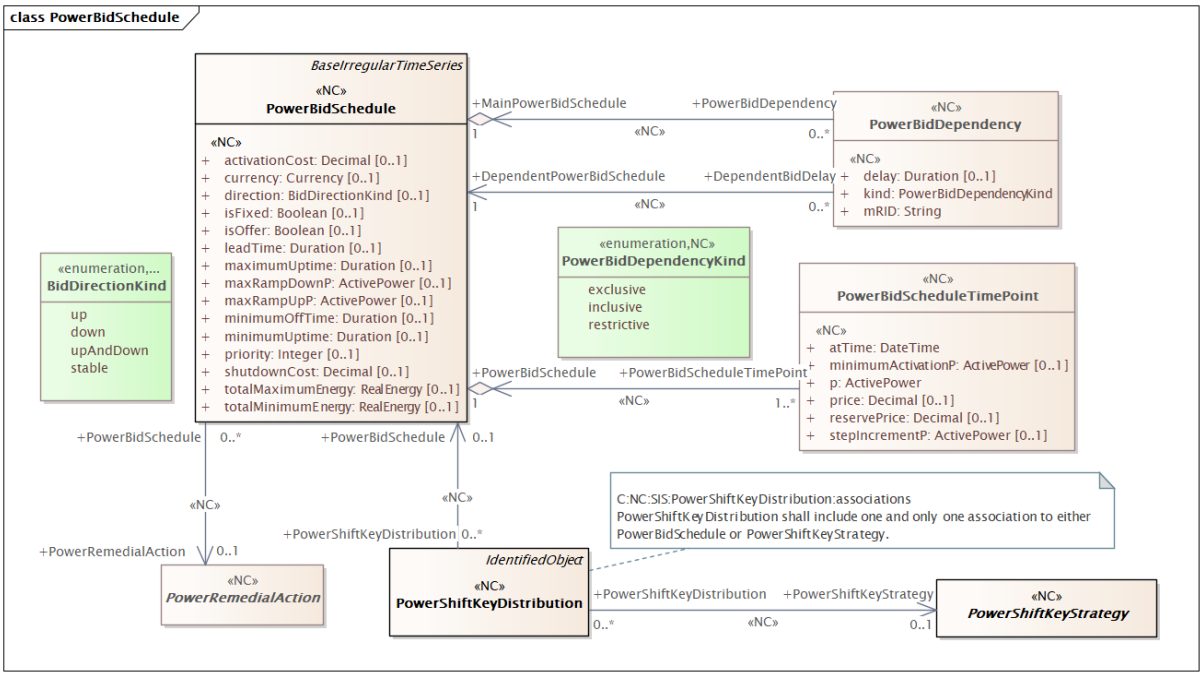
552 Figure 4: The diagram shows classes from Base CIM used in the security schedule profile.




553

554 **Figure 5 – Class diagram StateInstructionScheduleProfile::SecuritySchedule_One**

555 Figure 5: The diagram shows security schedule related classes.




556

557 **Figure 6 – Class diagram StateInstructionScheduleProfile::PowerBidSchedule**

558 Figure 6: The diagram shows power bid schedule related classes. The power bid schedule part
559 of the security schedule profile shall be used only for Coordinated Security Analysis (CSA)
560 process. We shall not use this profile for any market related bidding process. This profile should

not prevent to use the Reserve Bid document market profile if users want to use it for their local markets.

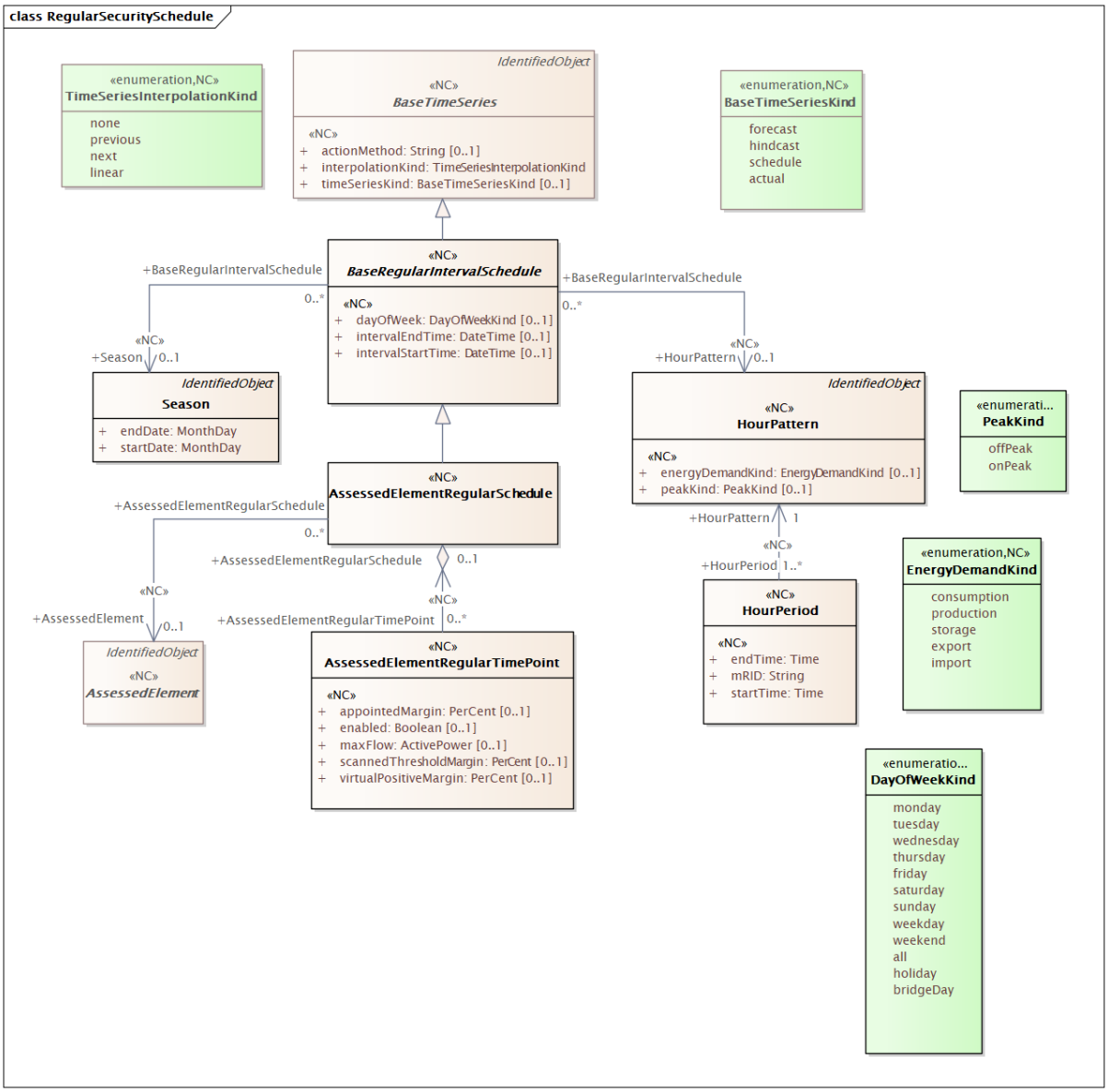



**Figure 7 – Class diagram StateInstructionScheduleProfile::RegularSecuritySchedule**

Figure 7: The diagram shows regular security schedule related classes.

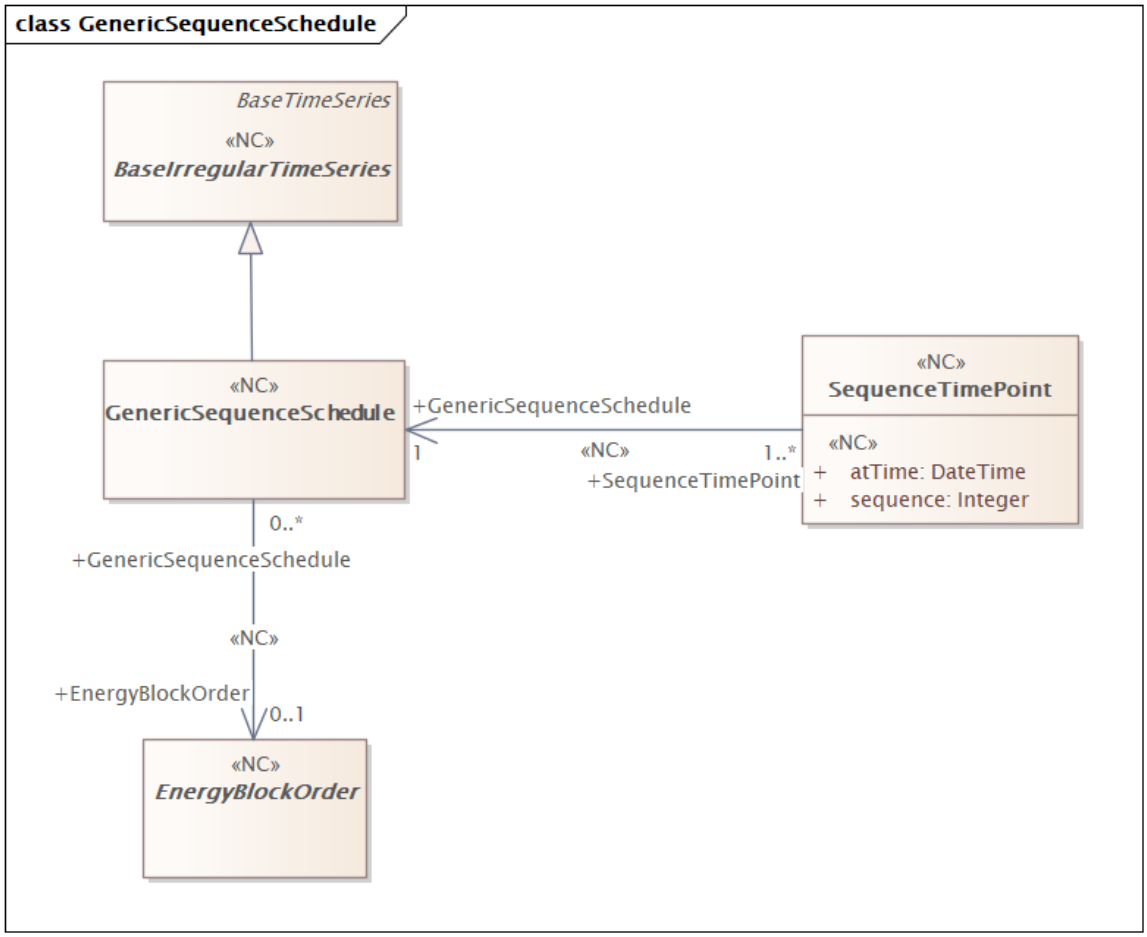



**Figure 8 – Class diagram StateInstructionScheduleProfile::GenericSequenceSchedule**

Figure 8: The diagram shows generic sequence schedule related classes.

### 3.2 (abstract,NC) BaseIrregularTimeSeries

Inheritance path = BaseTimeSeries : IdentifiedObject
Time series that has irregular points in time.
Table 1 shows all attributes of BaseIrregularTimeSeries.

**Table 1 – Attributes of StateInstructionScheduleProfile::BaseIrregularTimeSeries**

| name | mult | type | description |
|---|---|---|---|
| interpolationKind | 1..1 | TimeSeriesInterpolationKind | (NC) inherited from: BaseTimeSeries |
| timeSeriesKind | 0..1 | BaseTimeSeriesKind | (NC) inherited from: BaseTimeSeries |
| actionMethod | 0..1 | String | (NC) inherited from: BaseTimeSeries |
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

### 3.3 (abstract,NC) BaseTimeSeries

Inheritance path = IdentifiedObject
Time series of values at points in time.

Table 2 shows all attributes of BaseTimeSeries.

**Table 2 – Attributes of StateInstructionScheduleProfile::BaseTimeSeries**

| name | mult | type | description |
|---|---|---|---|
| interpolationKind | 1..1 | TimeSeriesInterpolationKind | (NC) Kind of interpolation done between time point. |
| timeSeriesKind | 0..1 | BaseTimeSeriesKind | (NC) Kind of base time series. |
| actionMethod | 0..1 | String | (NC) Action method used to create the value. This is used for identification in the case where there is multiple time series for the same validity period and kind. |
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

## 3.4 (abstract) ConductingEquipment

Inheritance path = Equipment : PowerSystemResource : IdentifiedObject

The parts of the AC power system that are designed to carry current or that are conductively connected through terminals.

Table 3 shows all attributes of ConductingEquipment.

**Table 3 – Attributes of StateInstructionScheduleProfile::ConductingEquipment**

| name | mult | type | description |
|---|---|---|---|
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

## 3.5 (abstract) EnergyConnection

Inheritance path = ConductingEquipment : Equipment : PowerSystemResource : IdentifiedObject

A connection of energy generation or consumption on the power system model.

Table 4 shows all attributes of EnergyConnection.

**Table 4 – Attributes of StateInstructionScheduleProfile::EnergyConnection**

| name | mult | type | description |
|---|---|---|---|
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

## 3.6 (abstract) EnergyConsumer root class

Generic user of energy - a point of consumption on the power system model.

EnergyConsumer.pfixed, .qfixed, .pfixedPct and .qfixedPct have meaning only if there is no LoadResponseCharacteristic associated with EnergyConsumer or if LoadResponseCharacteristic.exponentModel is set to False.

## 3.7 (abstract) Equipment

Inheritance path = PowerSystemResource : IdentifiedObject

The parts of a power system that are physical devices, electronic or mechanical.

Table 5 shows all attributes of Equipment.

**Table 5 – Attributes of StateInstructionScheduleProfile::Equipment**

| name | mult | type | description |
|---|---|---|---|
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

## 3.8 (abstract) GeneratingUnit root class

A single or set of synchronous machines for converting mechanical power into alternating-current power. For example, individual machines within a set may be defined for scheduling purposes while a single control signal is derived for the set. In this case there would be a GeneratingUnit for each member of the set and an additional GeneratingUnit corresponding to the set.

## 3.9 (NC) PowerShiftKeySchedule

Inheritance path = BaseIrregularTimeSeries : BaseTimeSeries : IdentifiedObject
The schedule for Power Shift Keys.
Table 6 shows all attributes of PowerShiftKeySchedule.

**Table 6 – Attributes of StateInstructionScheduleProfile::PowerShiftKeySchedule**

| name | mult | type | description |
|---|---|---|---|
| interpolationKind | 1..1 | TimeSeriesInterpolationKind | (NC) inherited from: BaseTimeSeries |
| timeSeriesKind | 0..1 | BaseTimeSeriesKind | (NC) inherited from: BaseTimeSeries |
| actionMethod | 0..1 | String | (NC) inherited from: BaseTimeSeries |
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

Table 7 shows all association ends of PowerShiftKeySchedule with other classes.

**Table 7 – Association ends of StateInstructionScheduleProfile::PowerShiftKeySchedule with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | EnergyConsumer | 0..1 | EnergyConsumer | (NC) The EnergyConsumer that has a Power Shift Key schedule. |
| 0..* | GeneratingUnit | 0..1 | GeneratingUnit | (NC) The Generating Unit which has a Power Shift Key Schedule. |
| 0..* | HydroPump | 0..1 | HydroPump | (NC) The Hydro Pump which has a Power Shift Key schedule. |
| 0..* | PowerElectronicsUnit | 0..1 | PowerElectronicsUnit | (NC) The Power Electronics Unit which has a Power Shift Key schedule. |
| 0..* | ScheduleResource | 0..1 | ScheduleResource | (NC) The Schedule Resource which has a Power Shift Key schedule. |
| 0..* | EnergyBlockOrder | 0..1 | EnergyBlockOrder | (NC) An energy block order which has a Power Shift Key Schedule. |
| 0..* | EnergyGroup | 0..1 | EnergyGroup | (NC) The energy group which has a Power Shift Key Schedule. |
| 0..* | DCPole | 0..1 | DCPole | (NC) A DC Pole which has a Power Shift Key Schedule. |

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | EquivalentInjection | 0..1 | EquivalentInjection | (NC) Equivalent injection which is part of a powoer shift key schedule. |
| 0..* | ExternalNetworkInjection | 0..1 | ExternalNetworkInjection | (NC) The energy source which has a power shift key schedule. |
| 0..* | EnergySource | 0..1 | EnergySource | (deprecated,NC) The energy source which has a power shift key schedule. The renewable resources should be modelled as PowerElecronicsUnit. |

## 3.10 (abstract) HydroPump root class

A synchronous motor-driven pump, typically associated with a pumped storage plant.

## 3.11 (abstract) IdentifiedObject root class

This is a root class to provide common identification for all classes needing identification and naming attributes.
Table 8 shows all attributes of IdentifiedObject.

**Table 8 – Attributes of StateInstructionScheduleProfile::IdentifiedObject**

| name | mult | type | description |
|---|---|---|---|
| description | 0..1 | String | The description is a free human readable text describing or naming the object. It may be non unique and may not correlate to a naming hierarchy. |
| mRID | 1..1 | String | Master resource identifier issued by a model authority. The mRID is unique within an exchange context. Global uniqueness is easily achieved by using a UUID, as specified in RFC 4122, for the mRID. The use of UUID is strongly recommended.<br>For CIMXML data files in RDF syntax conforming to IEC 61970-552, the mRID is mapped to rdf:ID or rdf:about attributes that identify CIM object elements. |
| name | 0..1 | String | The name is any free human readable and possibly non unique text naming the object. |

## 3.12 (abstract) PowerElectronicsUnit root class

A generating unit or battery or aggregation that connects to the AC network using power electronics rather than rotating machines.

## 3.13 (abstract) PowerSystemResource

Inheritance path = IdentifiedObject
A power system resource (PSR) can be an item of equipment such as a switch, an equipment container containing many individual items of equipment such as a substation, or an organisational entity such as sub-control area. Power system resources can have measurements associated.
Table 9 shows all attributes of PowerSystemResource.

**Table 9 – Attributes of StateInstructionScheduleProfile::PowerSystemResource**

| name | mult | type | description |
|---|---|---|---|
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |

| name | mult | type | description |
|---|---|---|---|
| name | 0..1 | String | inherited from: IdentifiedObject |

## 3.14 (abstract,NC) ScheduleResource root class

A schedule resource is a market-based method for handling participation of small units, particularly located on the lower voltage level that is controlled by a Distributed System Operator (DSO). It is a collection of units that can operate in the market by providing bids, offers and a resulting committed operational schedule for the collection.
Table 10 shows all association ends of ScheduleResource with other classes.

**Table 10 – Association ends of StateInstructionScheduleProfile::ScheduleResource with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..1 | PowerBidSchedule | 0..* | PowerBidSchedule | (NC) Power bid schedule which belongs to a schedule resource. |

## 3.15 (NC) BaseTimeSeriesKind enumeration

Kind of time series.
Table 11 shows all literals of BaseTimeSeriesKind.

**Table 11 – Literals of StateInstructionScheduleProfile::BaseTimeSeriesKind**

| literal | value | description |
|---|---|---|
| schedule | | Time series is schedule data. The values represent the result of a committed and plan forecast data that has been through a quality control and could incur penalty when not followed. |
| forecast | | Time series is forecast data. The values represent the result of scientific predictions based on historical time stamped data. |
| hindcast | | Time series is hindcast data. The value represent probable past (historic) condition given by calculation done using actual values. For instance, determine the among of wind based on the energy produced by wind. However, hindcast is typical the result of a simulated forecasts for historical periods. |
| actual | | Time series is actual data. The values represent measured or calculated values that represent the actual behaviour. |

## 3.16 (NC) TimeSeriesInterpolationKind enumeration

Kinds of interpolation of values between two time point.
Table 12 shows all literals of TimeSeriesInterpolationKind.

**Table 12 – Literals of StateInstructionScheduleProfile::TimeSeriesInterpolationKind**

| literal | value | description |
|---|---|---|
| none | | No interpolation is applied. |
| previous | | The value between two time points is set to previous value. |
| next | | The value between two time points is set to next value. |

| literal | value | description |
|---|---|---|
| linear | | Linear interpolation is applied for values between two time points. |

## 3.17 UnitMultiplier enumeration

The unit multipliers defined for the CIM. When applied to unit symbols, the unit symbol is treated as a derived unit. Regardless of the contents of the unit symbol text, the unit symbol shall be treated as if it were a single-character unit symbol. Unit symbols should not contain multipliers, and it should be left to the multiplier to define the multiple for an entire data type.
For example, if a unit symbol is "m2Pers" and the multiplier is "k", then the value is k(m**2/s), and the multiplier applies to the entire final value, not to any individual part of the value. This can be conceptualized by substituting a derived unit symbol for the unit type. If one imagines that the symbol "Þ" represents the derived unit "m2Pers", then applying the multiplier "k" can be conceptualized simply as "kÞ".
For example, the SI unit for mass is "kg" and not "g". If the unit symbol is defined as "kg", then the multiplier is applied to "kg" as a whole and does not replace the "k" in front of the "g". In this case, the multiplier of "m" would be used with the unit symbol of "kg" to represent one gram. As a text string, this violates the instructions in IEC 80000-1. However, because the unit symbol in CIM is treated as a derived unit instead of as an SI unit, it makes more sense to conceptualize the "kg" as if it were replaced by one of the proposed replacements for the SI mass symbol. If one imagines that the "kg" were replaced by a symbol "Þ", then it is easier to conceptualize the multiplier "m" as creating the proper unit "mÞ", and not the forbidden unit "mkg".
Table 13 shows all literals of UnitMultiplier.

**Table 13 – Literals of StateInstructionScheduleProfile::UnitMultiplier**

| literal | value | description |
|---|---|---|
| none | 0 | No multiplier or equivalently multiply by 1. |
| M | 6 | Mega 10**6. |

## 3.18 UnitSymbol enumeration

The derived units defined for usage in the CIM. In some cases, the derived unit is equal to an SI unit. Whenever possible, the standard derived symbol is used instead of the formula for the derived unit. For example, the unit symbol Farad is defined as "F" instead of "CPerV". In cases where a standard symbol does not exist for a derived unit, the formula for the unit is used as the unit symbol. For example, density does not have a standard symbol and so it is represented as "kgPerm3". With the exception of the "kg", which is an SI unit, the unit symbols do not contain multipliers and therefore represent the base derived unit to which a multiplier can be applied as a whole.
Every unit symbol is treated as an unparseable text as if it were a single-letter symbol. The meaning of each unit symbol is defined by the accompanying descriptive text and not by the text contents of the unit symbol.
To allow the widest possible range of serializations without requiring special character handling, several substitutions are made which deviate from the format described in IEC 80000-1. The division symbol "/" is replaced by the letters "Per". Exponents are written in plain text after the unit as "m3" instead of being formatted as "m" with a superscript of 3 or introducing a symbol as in "m^3". The degree symbol "°" is replaced with the letters "deg". Any clarification of the meaning for a substitution is included in the description for the unit symbol.
Non-SI units are included in list of unit symbols to allow sources of data to be correctly labelled with their non-SI units (for example, a GPS sensor that is reporting numbers that represent feet instead of meters). This allows software to use the unit symbol information correctly convert and scale the raw data of those sources into SI-based units.
The integer values are used for harmonization with IEC 61850.
Table 14 shows all literals of UnitSymbol.

706 **Table 14 – Literals of StateInstructionScheduleProfile::UnitSymbol**

| literal | value | description |
|---|---|---|
| none | 0 | Dimension less quantity, e.g. count, per unit, etc. |
| s | 4 | Time in seconds. |
| W | 38 | Real power in watts (J/s). Electrical power may have real and reactive components. The real portion of electrical power (I²R or VIcos(phi)), is expressed in Watts. See also apparent power and reactive power. |
| Wh | 72 | Real energy in watt hours. |

707

708 ### 3.19 Seconds datatype

709 Time, in seconds.
710 Table 15 shows all attributes of Seconds.

711 **Table 15 – Attributes of StateInstructionScheduleProfile::Seconds**

| name | mult | type | description |
|---|---|---|---|
| value | 0..1 | Float | Time, in seconds |
| unit | 0..1 | UnitSymbol | (const=s) |
| multiplier | 0..1 | UnitMultiplier | (const=none) |

712

713 ### 3.20 DateTime primitive

714 Date and time as "yyyy-mm-ddThh:mm:ss.sss", which conforms with ISO 8601. UTC time zone
715 is specified as "yyyy-mm-ddThh:mm:ss.sssZ". A local timezone relative UTC is specified as
716 "yyyy-mm-ddThh:mm:ss.sss-hh:mm". The second component (shown here as "ss.sss") could
717 have any number of digits in its fractional part to allow any kind of precision beyond seconds.

718 ### 3.21 Float primitive

719 A floating point number. The range is unspecified and not limited.

720 ### 3.22 String primitive

721 A string consisting of a sequence of characters. The character encoding is UTF-8. The string
722 length is unspecified and unlimited.

723 ### 3.23 (NC) ParticipationFactorTimePoint root class

724 Participation factor for a given point in time.
725 Table 16 shows all attributes of ParticipationFactorTimePoint.

726 **Table 16 – Attributes of StateInstructionScheduleProfile::ParticipationFactorTimePoint**

| name | mult | type | description |
|---|---|---|---|
| atTime | 0..1 | DateTime | (NC) The time the data is valid for. |
| participationFactor | 0..1 | Float | (NC) Participation factor describing the entity part of the active power provided by a collection of entities (e.g. an active power forecast to a collection of entities is divided to each of the member entity according to the participation factor). Must be a positive value.<br>In the case of a sharing strategy, the distribution is following entities value (V) equals aggregated value (T) divided by sum of participation factors (PF), i.e. V=T/sum(PF).<br>In the case of priority strategy, the item with the lowest number gets allocated energy first. |

Table 17 shows all association ends of ParticipationFactorTimePoint with other classes.

**Table 17 – Association ends of StateInstructionScheduleProfile::ParticipationFactorTimePoint with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | PowerShiftKeySchedule | 1..1 | PowerShiftKeySchedule | (NC) The Power Shift Key schedule which belongs to the participation factor timepoint. |

## 3.24 (NC) ContingencySchedule

Inheritance path = BaseIrregularTimeSeries : BaseTimeSeries : IdentifiedObject
The schedule for Contingency.
Table 18 shows all attributes of ContingencySchedule.

**Table 18 – Attributes of StateInstructionScheduleProfile::ContingencySchedule**

| name | mult | type | description |
|---|---|---|---|
| interpolationKind | 1..1 | TimeSeriesInterpolationKind | (NC) inherited from: BaseTimeSeries |
| timeSeriesKind | 0..1 | BaseTimeSeriesKind | (NC) inherited from: BaseTimeSeries |
| actionMethod | 0..1 | String | (NC) inherited from: BaseTimeSeries |
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

Table 19 shows all association ends of ContingencySchedule with other classes.

**Table 19 – Association ends of StateInstructionScheduleProfile::ContingencySchedule with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | Contingency | 1..1 | Contingency | (NC) Contingency which has a contingency schedule. |

## 3.25 (NC) ContingencyTimePoint root class

Contingency instruction value at a given point in time.
Table 20 shows all attributes of ContingencyTimePoint.

**Table 20 – Attributes of StateInstructionScheduleProfile::ContingencyTimePoint**

| name | mult | type | description |
|---|---|---|---|
| atTime | 1..1 | DateTime | (NC) The time the data is valid for. |
| probability | 0..1 | PerCent | (NC) Probability of occurrence. The allowed value range is [0,100]. |
| mustStudy | 0..1 | Boolean | (NC) Set true if must study this contingency. |

Table 21 shows all association ends of ContingencyTimePoint with other classes.

748 **Table 21 – Association ends of StateInstructionScheduleProfile::ContingencyTimePoint**
749 **with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 1..* | ContingencySchedule | 1..1 | ContingencySchedule | (NC) The contingency schedule that has this time point. |

750

751 ### 3.26 (abstract) Contingency root class

752 An event threatening system reliability, consisting of one or more contingency elements.

753 ### 3.27 (NC) AssessedElementSchedule

754 Inheritance path = BaseIrregularTimeSeries : BaseTimeSeries : IdentifiedObject
755 Schedule for assessed element.
756 Table 22 shows all attributes of AssessedElementSchedule.

757 **Table 22 – Attributes of StateInstructionScheduleProfile::AssessedElementSchedule**

| name | mult | type | description |
|---|---|---|---|
| interpolationKind | 1..1 | TimeSeriesInterpolationKind | (NC) inherited from: BaseTimeSeries |
| timeSeriesKind | 0..1 | BaseTimeSeriesKind | (NC) inherited from: BaseTimeSeries |
| actionMethod | 0..1 | String | (NC) inherited from: BaseTimeSeries |
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

758
759 Table 23 shows all association ends of AssessedElementSchedule with other classes.

760 **Table 23 – Association ends of**
761 **StateInstructionScheduleProfile::AssessedElementSchedule with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | AssessedElement | 1..1 | AssessedElement | (NC) Assessed element which has an assessed element schedule. |

762

763 ### 3.28 (NC) AssessedElementTimePoint root class

764 Assessed element instruction value at a given point in time.
765 Table 24 shows all attributes of AssessedElementTimePoint.

766 **Table 24 – Attributes of StateInstructionScheduleProfile::AssessedElementTimePoint**

| name | mult | type | description |
|---|---|---|---|
| enabled | 0..1 | Boolean | (NC) It identifies if the assessed element is enabled. True means enabled, False means disabled. |
| appointedMargin | 0..1 | PerCent | (NC) The percentage (appointed to a region) of the remaining margin obtained in the grid model to reach its current limit. The maximum percentage shall by default be 10% of the remaining margin.<br>It is only used when an assessed element is considered conservative for a region.<br>The allowed value range is [0,100]. |

| name | mult | type | description |
|---|---|---|---|
| maxFlow | 0..1 | ActivePower | (NC) Maximum flow on an a conducting equipment or a collection of conducting equipment forming a power transfer corridor. For assessed elements that is becomes critical due to contingency, this value represents the maximum flow with remedial action taken into consideration. |
| atTime | 1..1 | DateTime | (NC) The time the data is valid for. |
| virtualPositiveMargin | 0..1 | PerCent | (NC) A margin defined only for scanned AssessedElement (If AssessedElement.ScannedForRegion is present) in order to represent the influence of available remedial action which is not cross-border relevant remedial action.<br>The margin is modifying the limits used for the assessment whatever the limit it is (e.g. PATL, TATL).This symbolizes a remedial action that can be applied internally by the System Operator. It will be resolved by the System Operator and not by the optimization of remedial actions. The attribute shall be a positive value.<br>The allowed value range is [0,100]. |
| scannedThresholdMargin | 0..1 | PerCent | (NC) Threshold percentage that a scanned element can be overloaded, on a given element, on top of any overload prior to optimisation (default= 5%). e.g. Initial loading of the element is 110%, with a 5% scanned threshold margin, the new maximum is 115% of the limit (e.g. PATL, TATL, etc).<br>The allowed value range is [0,100]. |
| exclusionReason | 0..1 | SecuredExclusionReasonKind | (NC) Reason for not associating this assessed element with a secured region. |
| individualValidationAdjustment | 0..1 | ActivePower | (NC) A positive value expressed in MW, calculated and provided by System Operators from their individual validation process for the reduction of Remaining Available Margin in order to ensure grid security. |
| coordinatedValidationAdjustment | 0..1 | ActivePower | (NC) A positive value expressed in MW, calculated and provided by the coordinated capacity calculator (CCC) for the reduction of Remaining Available Margin (RAM) in order to ensure grid security. |
| individualValidationAdjustmentShare | 0..1 | ActivePower | (NC) A positive value expressed in MW, calculated by the coordinated capacity calculator (CCC) based on the provided Individual Validation Adjustment (IVA) by System Operators in order to show the actual reduction of Remaining Available Margin (RAM). Individual Validation Adjustment Share is a positive non-zero value. It is equal or less than the Individual Validation Adjustment value. |
| individualValidationAdjustmentJustification | 0..1 | String | (NC) A text description provided by System Operators for justifying the reduction of Remaining Available Margin (RAM) by means of Individual Validation Adjustment (IVA). This justification is not intended for any application processing purpose, it should only be used for reporting. |
| coordinatedValidationAdjustmentJustification | 0..1 | String | (NC) A text description provided by the coordinated capacity calculator (CCC) for justifying the reduction of Remaining Available Margin (RAM) by means of Coordinated |

| name | mult | type | description |
|---|---|---|---|
| | | | Validation Adjustment (CVA). This justification is not intended for any application processing purpose, it should only be used for reporting. |
| criticalElementContingencyJustification | 0..1 | String | (NC) Justification indicating the kind of critical element contingency. This justification is not intended for any application processing purpose, it should only be used for reporting. |
| targetRemainingAvailableMarginJustification | 0..1 | String | (NC) Justification indicating the target remaining available margin. This justification is not intended for any application processing purpose, it should only be used for reporting. |

Table 25 shows all association ends of AssessedElementTimePoint with other classes.

**Table 25 – Association ends of StateInstructionScheduleProfile::AssessedElementTimePoint with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 1..* | AssessedElementSchedule | 1..1 | AssessedElementSchedule | (NC) The assessed element schedule that has this time point. |

### 3.29 (abstract,NC) AssessedElement

Inheritance path = IdentifiedObject

Assessed element is a network element for which the electrical state is evaluated in the regional or cross-regional process and which value is expected to fulfil regional rules function of the operational security limits.

The measurements and limits are as defined in the steady state hypothesis.

Table 26 shows all attributes of AssessedElement.

**Table 26 – Attributes of StateInstructionScheduleProfile::AssessedElement**

| name | mult | type | description |
|---|---|---|---|
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

### 3.30 (NC) GenericValueSchedule

Inheritance path = BaseIrregularTimeSeries : BaseTimeSeries : IdentifiedObject

Time series represent irregular generic value at given points in time. The type of value is given by the reference association.

Table 27 shows all attributes of GenericValueSchedule.

**Table 27 – Attributes of StateInstructionScheduleProfile::GenericValueSchedule**

| name | mult | type | description |
|---|---|---|---|
| interpolationKind | 1..1 | TimeSeriesInterpolationKind | (NC) inherited from: BaseTimeSeries |
| timeSeriesKind | 0..1 | BaseTimeSeriesKind | (NC) inherited from: BaseTimeSeries |
| actionMethod | 0..1 | String | (NC) inherited from: BaseTimeSeries |
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

Table 28 shows all association ends of GenericValueSchedule with other classes.

**Table 28 – Association ends of StateInstructionScheduleProfile::GenericValueSchedule with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | RangeConstraint | 0..1 | RangeConstraint | (NC) Range constraint for the generic value schedule. |

### 3.31 (abstract,NC) RangeConstraint

Inheritance path = IdentifiedObject
Defines the rage constraint.
Table 29 shows all attributes of RangeConstraint.

**Table 29 – Attributes of StateInstructionScheduleProfile::RangeConstraint**

| name | mult | type | description |
|---|---|---|---|
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

### 3.32 (NC) GenericValueTimePoint root class

Generic value for a given point in time.
Table 30 shows all attributes of GenericValueTimePoint.

**Table 30 – Attributes of StateInstructionScheduleProfile::GenericValueTimePoint**

| name | mult | type | description |
|---|---|---|---|
| atTime | 1..1 | DateTime | (NC) The time the data is valid for. |
| value | 1..1 | Float | (NC) The value at the time. The meaning of the value is defined by the derived type of the associated schedule. The value can be integer, float or boolean. In case of boolean 1 equals true and 0 equals false. |

Table 31 shows all association ends of GenericValueTimePoint with other classes.

**Table 31 – Association ends of StateInstructionScheduleProfile::GenericValueTimePoint with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 1..* | GenericValueSchedule | 1..1 | GenericValueSchedule | (NC) Time series the time point values belongs to. |

### 3.33 (NC) GenericAvailableSchedule

Inheritance path = BaseIrregularTimeSeries : BaseTimeSeries : IdentifiedObject
The schedule for the availability of elements.
Table 32 shows all attributes of GenericAvailableSchedule.

**Table 32 – Attributes of StateInstructionScheduleProfile::GenericAvailableSchedule**

| name | mult | type | description |
|---|---|---|---|
| interpolationKind | 1..1 | TimeSeriesInterpolationKind | (NC) inherited from: BaseTimeSeries |
| timeSeriesKind | 0..1 | BaseTimeSeriesKind | (NC) inherited from: BaseTimeSeries |
| actionMethod | 0..1 | String | (NC) inherited from: BaseTimeSeries |
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

Table 33 shows all association ends of GenericAvailableSchedule with other classes.

**Table 33 – Association ends of StateInstructionScheduleProfile::GenericAvailableSchedule with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | RemedialAction | 0..1 | RemedialAction | (NC) Remedial action which has available schedules. |

## 3.34 (NC) GenericEnablingSchedule

Inheritance path = BaseIrregularTimeSeries : BaseTimeSeries : IdentifiedObject

The schedule for the enabling of elements.

Table 34 shows all attributes of GenericEnablingSchedule.

**Table 34 – Attributes of StateInstructionScheduleProfile::GenericEnablingSchedule**

| name | mult | type | description |
|---|---|---|---|
| interpolationKind | 1..1 | TimeSeriesInterpolationKind | (NC) inherited from: BaseTimeSeries |
| timeSeriesKind | 0..1 | BaseTimeSeriesKind | (NC) inherited from: BaseTimeSeries |
| actionMethod | 0..1 | String | (NC) inherited from: BaseTimeSeries |
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

Table 35 shows all association ends of GenericEnablingSchedule with other classes.

**Table 35 – Association ends of StateInstructionScheduleProfile::GenericEnablingSchedule with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | AssessedElementWithContingency | 0..1 | AssessedElementWithContingency | (NC) Assessed element with contingency that has enabling schedules. |
| 0..* | AssessedElementWithRemedialAction | 0..1 | AssessedElementWithRemedialAction | (NC) Assessed element with remedial action that has enabling schedules. |
| 0..* | ContingencyWithRemedialAction | 0..1 | ContingencyWithRemedialAction | (NC) Contingency with remedial action which has enabling schedules. |
| 0..* | AutomationFunction | 0..1 | AutomationFunction | (NC) Automation function which has enabling schedules. |

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | FunctionBlock | 0..1 | FunctionBlock | (NC) Function block which has enabling schedules. |
| 0..* | RemedialActionDependency | 0..1 | RemedialActionDependency | (NC) Remedial action dependency which has enabling schedules. |
| 0..* | PowerTransferCorridor | 0..1 | PowerTransferCorridor | (NC) Power transfer corridor which has generic enabling schedules. |
| 0..* | CrossBorderRelevance | 0..1 | CrossBorderRelevance | (NC) Cross border relevant that has enabling schedules. |

## 3.35 (abstract,NC) RemedialAction

Inheritance path = IdentifiedObject
Remedial action describes one or more actions that can be performed on a given power system model situation to eliminate one or more identified breaches of constraints. The remedial action can be costly, and have a cost characteristic, or non costly.
Table 36 shows all attributes of RemedialAction.

**Table 36 – Attributes of StateInstructionScheduleProfile::RemedialAction**

| name | mult | type | description |
|---|---|---|---|
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

## 3.36 (NC) EnablingTimePoint root class

Enabling instruction value at a given point in time.
Table 37 shows all attributes of EnablingTimePoint.

**Table 37 – Attributes of StateInstructionScheduleProfile::EnablingTimePoint**

| name | mult | type | description |
|---|---|---|---|
| atTime | 1..1 | DateTime | (NC) The time the data is valid for. |
| enabled | 1..1 | Boolean | (NC) It identifies if the element is enabled. True means enabled, False means not enabled. |

Table 38 shows all association ends of EnablingTimePoint with other classes.

**Table 38 – Association ends of StateInstructionScheduleProfile::EnablingTimePoint with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 1..* | GenericEnablingSchedule | 1..1 | GenericEnablingSchedule | (NC) The enabling schedule which belongs to the enabling timepoint. |

## 3.37 (NC) AvailabilityTimePoint root class

Availability instruction value at a given point in time.
Table 39 shows all attributes of AvailabilityTimePoint.

**Table 39 – Attributes of StateInstructionScheduleProfile::AvailabilityTimePoint**

| name | mult | type | description |
|---|---|---|---|
| atTime | 1..1 | DateTime | (NC) The time the data is valid for. |
| available | 1..1 | Boolean | (NC) It identifies if the element is available. True means available, False means unavailable. |

Table 40 shows all association ends of AvailabilityTimePoint with other classes.

**Table 40 – Association ends of StateInstructionScheduleProfile::AvailabilityTimePoint with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 1..* | GenericAvailabilitySchedule | 1..1 | GenericAvailableSchedule | (NC) The availability schedule which belongs to the availability timepoint. |

## 3.38 (NC) PowerBidDependency root class

Dependency between the related power bids.
Table 41 shows all attributes of PowerBidDependency.

**Table 41 – Attributes of StateInstructionScheduleProfile::PowerBidDependency**

| name | mult | type | description |
|---|---|---|---|
| kind | 1..1 | PowerBidDependencyKind | (NC) Type of dependency between bids. |
| delay | 0..1 | Duration | (NC) Time delay between activation of the parents until the dependent offer will be available. |
| mRID | 1..1 | String | (NC) Master resource identifier issued by a model authority. The mRID is unique within an exchange context. Global uniqueness is easily achieved by using a UUID, as specified in RFC 4122, for the mRID. The use of UUID is strongly recommended.<br>For CIMXML data files in RDF syntax conforming to IEC 61970-552, the mRID is mapped to rdf:ID or rdf:about attributes that identify CIM object elements. |

Table 42 shows all association ends of PowerBidDependency with other classes.

**Table 42 – Association ends of StateInstructionScheduleProfile::PowerBidDependency with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | DependentPowerBidSchedule | 1..1 | PowerBidSchedule | (NC) Dependent power bid which has some dependent bid delays. |
| 0..* | MainPowerBidSchedule | 1..1 | PowerBidSchedule | (NC) Main power bid which some dependent power bids. |

## 3.39 (NC) PowerBidSchedule

Inheritance path = BaseIrregularTimeSeries : BaseTimeSeries : IdentifiedObject
Power bid or offer related to a redispatch or countertrading measures. In the case of market place for economic efficiency of the bids and offers, this is equivalent to BidTimeSeries class in 62325 package.

867 Table 43 shows all attributes of PowerBidSchedule.

868 **Table 43 – Attributes of StateInstructionScheduleProfile::PowerBidSchedule**

| name | mult | type | description |
|---|---|---|---|
| isOffer | 0..1 | Boolean | (NC) Indicates if the power bid is an offer or not. True, means that the bid is an offer. False, means that the bid is not an offer. |
| totalMaximumEnergy | 0..1 | RealEnergy | (NC) Maximum total energy that can be activated by the bid. |
| direction | 0..1 | BidDirectionKind | (NC) Define the direction of the energy adjustment. |
| currency | 0..1 | Currency | (NC) Currency of the bid. |
| totalMinimumEnergy | 0..1 | RealEnergy | (NC) Minimum total energy that has to be activated by the bid. |
| priority | 0..1 | Integer | (NC) The numeric local priority given to a bid. Lower numeric values will have higher priority. |
| maximumUptime | 0..1 | Duration | (NC) Maximum duration the action needs to be remain active after startup. |
| minimumUptime | 0..1 | Duration | (NC) Minimum duration the action needs to be remain active after startup. |
| activationCost | 0..1 | Decimal | (NC) Cost to activate the bid. |
| shutdownCost | 0..1 | Decimal | (NC) Total shutdown cost incurred for all the units involved in the bid. This overrides any cost on the specific unit. |
| leadTime | 0..1 | Duration | (NC) Time it takes for the bid to be called upon until it is active. |
| minimumOffTime | 0..1 | Duration | (NC) Minimum time interval between activation of the bid involving startup and shutdown. This value overrides any value on the unit. |
| isFixed | 0..1 | Boolean | (NC) Indicates if the power bid schedule is fixed, meaning that all the different power bid schedule values need to be taken without changes. e.g. It is a take-it-or-leave-it bid offer. |
| maxRampDownP | 0..1 | ActivePower | (NC) Maximum decrease of the active power change from one time point to the next. |
| maxRampUpP | 0..1 | ActivePower | (NC) Maximum increase of the active power change from one time point to the next. |
| interpolationKind | 1..1 | TimeSeriesInterpolationKind | (NC) inherited from: BaseTimeSeries |
| timeSeriesKind | 0..1 | BaseTimeSeriesKind | (NC) inherited from: BaseTimeSeries |
| actionMethod | 0..1 | String | (NC) inherited from: BaseTimeSeries |
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

869

870 Table 44 shows all association ends of PowerBidSchedule with other classes.

**Table 44 – Association ends of StateInstructionScheduleProfile::PowerBidSchedule with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | ScheduleResource | 0..1 | ScheduleResource | (NC) Schedule resource which has several power bid schedules. |
| 0..* | PowerRemedialAction | 0..1 | PowerRemedialAction | (NC) Power remedial action for which the bid is given. |

### 3.40 (NC) PowerBidScheduleTimePoint root class

Time series represent irregular power, active and reactive, values at given points in time. Table 45 shows all attributes of PowerBidScheduleTimePoint.

**Table 45 – Attributes of StateInstructionScheduleProfile::PowerBidScheduleTimePoint**

| name | mult | type | description |
|---|---|---|---|
| atTime | 1..1 | DateTime | (NC) The time the data is valid for. |
| price | 0..1 | Decimal | (NC) Quantity given in the time points. |
| p | 1..1 | ActivePower | (NC) Active power given in the time point. |
| minimumActivationP | 0..1 | ActivePower | (NC) Minimum active power given in the time point. |
| reservePrice | 0..1 | Decimal | (NC) Price for reserving the step increment active power. |
| stepIncrementP | 0..1 | ActivePower | (NC) The minimum increment that can be applied for an increase in an activation request. |

Table 46 shows all association ends of PowerBidScheduleTimePoint with other classes.

**Table 46 – Association ends of StateInstructionScheduleProfile::PowerBidScheduleTimePoint with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 1..* | PowerBidSchedule | 1..1 | PowerBidSchedule | (NC) Power bid schedule that has many power bid schedule time points. |

### 3.41 (abstract,NC) AssessedElementWithRemedialAction root class

Combination of an assessed element and a remedial action

### 3.42 (abstract,NC) AssessedElementWithContingency root class

Combination of an assessed element and a contingency.

### 3.43 Duration primitive

Duration as "PnYnMnDTnHnMnS" which conforms to ISO 8601, where nY expresses a number of years, nM a number of months, nD a number of days. The letter T separates the date expression from the time expression and, after it, nH identifies a number of hours, nM a number of minutes and nS a number of seconds. The number of seconds could be expressed as a decimal number, but all other numbers are integers.

### 3.44 Boolean primitive

A type with the value space "true" and "false".

### 3.45 RealEnergy datatype

Real electrical energy.

Table 47 shows all attributes of RealEnergy.

**Table 47 – Attributes of StateInstructionScheduleProfile::RealEnergy**

| name | mult | type | description |
|---|---|---|---|
| multiplier | 0..1 | UnitMultiplier | (const=M) |
| unit | 0..1 | UnitSymbol | (const=Wh) |
| value | 0..1 | Float | |

## 3.46 (NC) BidDirectionKind enumeration

Kind of direction of the bid.
Table 48 shows all literals of BidDirectionKind.

**Table 48 – Literals of StateInstructionScheduleProfile::BidDirectionKind**

| literal | value | description |
|---|---|---|
| up | | Up signifies that the available power can be used by the purchasing area to increase energy. |
| down | | Down signifies that the available power can be used by the purchasing area to decrease energy. |
| upAndDown | | Up and down signifies that both up and down values are equal. |
| stable | | The direction at a given instant in time is considered to be stable. |

## 3.47 Currency enumeration

Monetary currencies. ISO 4217 standard including 3-character currency code.
Table 49 shows all literals of Currency.

**Table 49 – Literals of StateInstructionScheduleProfile::Currency**

| literal | value | description |
|---|---|---|
| AED | 784 | United Arab Emirates dirham. |
| AFN | 971 | Afghan afghani. |
| ALL | 008 | Albanian lek. |
| AMD | 051 | Armenian dram. |
| ANG | 532 | Netherlands Antillean guilder. |
| AOA | 973 | Angolan kwanza. |
| ARS | 032 | Argentine peso. |
| AUD | 036 | Australian dollar. |
| AWG | 533 | Aruban florin. |
| AZN | 944 | Azerbaijani manat. |
| BAM | 977 | Bosnia and Herzegovina convertible mark. |
| BBD | 052 | Barbados dollar. |
| BDT | 050 | Bangladeshi taka. |
| BGN | 975 | Bulgarian lev. |
| BHD | 048 | Bahraini dinar. |
| BIF | 108 | Burundian franc. |

| literal | value | description |
|---|---|---|
| BMD | 060 | Bermudian dollar (customarily known as Bermuda dollar). |
| BND | 096 | Brunei dollar. |
| BOB | 068 | Boliviano. |
| BOV | 984 | Bolivian Mvdol (funds code). |
| BRL | 986 | Brazilian real. |
| BSD | 044 | Bahamian dollar. |
| BTN | 064 | Bhutanese ngultrum. |
| BWP | 072 | Botswana pula. |
| BYR | 974 | Belarusian ruble. |
| BZD | 084 | Belize dollar. |
| CAD | 124 | Canadian dollar. |
| CDF | 976 | Congolese franc. |
| CHF | 756 | Swiss franc. |
| CLF | 990 | Unidad de Fomento (funds code), Chile. |
| CLP | 152 | Chilean peso. |
| CNY | 156 | Chinese yuan. |
| COP | 170 | Colombian peso. |
| COU | 970 | Unidad de Valor Real. |
| CRC | 188 | Costa Rican colon. |
| CUC | 931 | Cuban convertible peso. |
| CUP | 192 | Cuban peso. |
| CVE | 132 | Cape Verde escudo. |
| CZK | 203 | Czech koruna. |
| DJF | 262 | Djiboutian franc. |
| DKK | 208 | Danish krone. |
| DOP | 214 | Dominican peso. |
| DZD | 012 | Algerian dinar. |
| EEK | 233 | Estonian kroon. |
| EGP | 818 | Egyptian pound. |
| ERN | 232 | Eritrean nakfa. |
| ETB | 230 | Ethiopian birr. |
| EUR | 978 | Euro. |
| FJD | 242 | Fiji dollar. |
| FKP | 238 | Falkland Islands pound. |
| GBP | 826 | Pound sterling. |
| GEL | 981 | Georgian lari. |
| GHS | 936 | Ghanaian cedi. |
| GIP | 929 | Gibraltar pound. |
| GMD | 270 | Gambian dalasi. |
| GNF | 324 | Guinean franc. |

| literal | value | description |
|---|---|---|
| GTQ | 320 | Guatemalan quetzal. |
| GYD | 328 | Guyanese dollar. |
| HKD | 344 | Hong Kong dollar. |
| HNL | 340 | Honduran lempira. |
| HRK | 191 | Croatian kuna. |
| HTG | 332 | Haitian gourde. |
| HUF | 348 | Hungarian forint. |
| IDR | 360 | Indonesian rupiah. |
| ILS | 376 | Israeli new sheqel. |
| INR | 356 | Indian rupee. |
| IQD | 368 | Iraqi dinar. |
| IRR | 364 | Iranian rial. |
| ISK | 352 | Icelandic króna. |
| JMD | 388 | Jamaican dollar. |
| JOD | 400 | Jordanian dinar. |
| JPY | 392 | Japanese yen. |
| KES | 404 | Kenyan shilling. |
| KGS | 417 | Kyrgyzstani som. |
| KHR | 116 | Cambodian riel. |
| KMF | 174 | Comoro franc. |
| KPW | 408 | North Korean won. |
| KRW | 410 | South Korean won. |
| KWD | 414 | Kuwaiti dinar. |
| KYD | 136 | Cayman Islands dollar. |
| KZT | 398 | Kazakhstani tenge. |
| LAK | 418 | Lao kip. |
| LBP | 422 | Lebanese pound. |
| LKR | 144 | Sri Lanka rupee. |
| LRD | 430 | Liberian dollar. |
| LSL | 426 | Lesotho loti. |
| LTL | 440 | Lithuanian litas. |
| LVL | 428 | Latvian lats. |
| LYD | 434 | Libyan dinar. |
| MAD | 504 | Moroccan dirham. |
| MDL | 498 | Moldovan leu. |
| MGA | 969 | Malagasy ariary. |
| MKD | 807 | Macedonian denar. |
| MMK | 104 | Myanma kyat. |
| MNT | 496 | Mongolian tugrik. |
| MOP | 446 | Macanese pataca. |
| MRO | 478 | Mauritanian ouguiya. |

| literal | value | description |
|---|---|---|
| MUR | 480 | Mauritian rupee. |
| MVR | 462 | Maldivian rufiyaa. |
| MWK | 454 | Malawian kwacha. |
| MXN | 484 | Mexican peso. |
| MYR | 458 | Malaysian ringgit. |
| MZN | 943 | Mozambican metical. |
| NAD | 516 | Namibian dollar. |
| NGN | 566 | Nigerian naira. |
| NIO | 558 | Cordoba oro. |
| NOK | 578 | Norwegian krone. |
| NPR | 524 | Nepalese rupee. |
| NZD | 554 | New Zealand dollar. |
| OMR | 512 | Omani rial. |
| PAB | 590 | Panamanian balboa. |
| PEN | 604 | Peruvian nuevo sol. |
| PGK | 598 | Papua New Guinean kina. |
| PHP | 608 | Philippine peso. |
| PKR | 586 | Pakistani rupee. |
| PLN | 985 | Polish zloty. |
| PYG | 600 | Paraguayan guaraní. |
| QAR | 634 | Qatari rial. |
| RON | 946 | Romanian new leu. |
| RSD | 941 | Serbian dinar. |
| RUB | 643 | Russian rouble. |
| RWF | 646 | Rwandan franc. |
| SAR | 682 | Saudi riyal. |
| SBD | 090 | Solomon Islands dollar. |
| SCR | 690 | Seychelles rupee. |
| SDG | 938 | Sudanese pound. |
| SEK | 752 | Swedish krona/kronor. |
| SGD | 702 | Singapore dollar. |
| SHP | 654 | Saint Helena pound. |
| SLL | 694 | Sierra Leonean leone. |
| SOS | 706 | Somali shilling. |
| SRD | 968 | Surinamese dollar. |
| STD | 678 | São Tomé and Príncipe dobra. |
| SYP | 760 | Syrian pound. |
| SZL | 748 | Lilangeni. |
| THB | 764 | Thai baht. |
| TJS | 972 | Tajikistani somoni. |
| TMT | 934 | Turkmenistani manat. |

| literal | value | description |
|---|---|---|
| TND | 788 | Tunisian dinar. |
| TOP | 776 | Tongan pa'anga. |
| TRY | 949 | Turkish lira. |
| TTD | 780 | Trinidad and Tobago dollar. |
| TWD | 901 | New Taiwan dollar. |
| TZS | 834 | Tanzanian shilling. |
| UAH | 980 | Ukrainian hryvnia. |
| UGX | 800 | Ugandan shilling. |
| USD | 840 | United States dollar. |
| UYU | 858 | Uruguayan peso. |
| UZS | 860 | Uzbekistan som. |
| VEF | 937 | Venezuelan bolívar fuerte. |
| VND | 704 | Vietnamese Dong. |
| VUV | 548 | Vanuatu vatu. |
| WST | 882 | Samoan tala. |
| XAF | 950 | CFA franc BEAC. |
| XCD | 951 | East Caribbean dollar. |
| XOF | 952 | CFA Franc BCEAO. |
| XPF | 953 | CFP franc. |
| YER | 886 | Yemeni rial. |
| ZAR | 710 | South African rand. |
| ZMK | 894 | Zambian kwacha. |
| ZWL | 932 | Zimbabwe dollar. |

## 3.48 ActivePower datatype

Product of RMS value of the voltage and the RMS value of the in-phase component of the current.

Table 50 shows all attributes of ActivePower.

**Table 50 – Attributes of StateInstructionScheduleProfile::ActivePower**

| name | mult | type | description |
|---|---|---|---|
| multiplier | 0..1 | UnitMultiplier | (const=M) |
| unit | 0..1 | UnitSymbol | (const=W) |
| value | 0..1 | Float | |

## 3.49 PerCent datatype

Percentage on a defined base. For example, specify as 100 to indicate at the defined base.

Table 51 shows all attributes of PerCent.

**Table 51 – Attributes of StateInstructionScheduleProfile::PerCent**

| name | mult | type | description |
|---|---|---|---|
| value | 0..1 | Float | Normally 0 to 100 on a defined base. |
| unit | 0..1 | UnitSymbol | (const=none) |

| name | mult | type | description |
|---|---|---|---|
| multiplier | 0..1 | UnitMultiplier | (const=none) |

### 3.50 Decimal primitive

Decimal is the base-10 notational system for representing real numbers.

### 3.51 Integer primitive

An integer number. The range is unspecified and not limited.

### 3.52 (abstract,NC) RemedialActionDependency root class

Remedial action dependency is making two remedial actions depending on each other. Multiple dependency is done by multiple instances of this class. The dependency can arrive by having one of the following examples.
- The dependent remedial action is controlled by different system operator (Modeling Authority) (e.g. SIPS that goes across control area).
- The dependent remedial action is representing two or more remedial action that represent the same grid state alteration but with different modeling resolution (e.g. detail direct current model versus a simplified model).
- The remedial action can be combined with other remedial action without the need to create multiple remedial action with the same grid alteration for enabling dependency.

### 3.53 (NC) PowerBidDependencyKind enumeration

Kind of power bid dependency.
Table 52 shows all literals of PowerBidDependencyKind.

**Table 52 – Literals of StateInstructionScheduleProfile::PowerBidDependencyKind**

| literal | value | description |
|---|---|---|
| exclusive | | Bids are exclusive depending on each other. e.g. Only one of the bids can be activated at the same time. |
| inclusive | | Bids are inclusive depending on each other. e.g. Both bids need to be activated if one of them is activated. |
| restrictive | | Bids are restrictive depending on each other. e.g. You have to take the father bid before you might take the child bid. |

### 3.54 (abstract,NC) ContingencyWithRemedialAction root class

Combination of a contingency and a remedial action. ContingencyWithRemedialAction shall not be instantiated for preventive RemedialAction (RemedialAction.kind equals RemedialActionKind.preventive).

### 3.55 (NC) GridStateIntensitySchedule

Inheritance path = GenericValueSchedule : BaseIrregularTimeSeries : BaseTimeSeries : IdentifiedObject
Defines the intensity applied for a given grid state alteration. It is primarily used in exchanges related to the remedial action schedule. The value provided by the schedule replaces the value of the attribute to which the schedule refers to.
Table 53 shows all attributes of GridStateIntensitySchedule.

**Table 53 – Attributes of StateInstructionScheduleProfile::GridStateIntensitySchedule**

| name | mult | type | description |
|---|---|---|---|
| valueKind | 0..1 | ValueOffsetKind | (NC) The kind of value1 and value2 of the associated IrregularIntervalSchedule. |

| name | mult | type | description |
|---|---|---|---|
| interpolationKind | 1..1 | TimeSeriesInterpolationKind | (NC) inherited from: BaseTimeSeries |
| timeSeriesKind | 0..1 | BaseTimeSeriesKind | (NC) inherited from: BaseTimeSeries |
| actionMethod | 0..1 | String | (NC) inherited from: BaseTimeSeries |
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

Table 54 shows all association ends of GridStateIntensitySchedule with other classes.

**Table 54 – Association ends of StateInstructionScheduleProfile::GridStateIntensitySchedule with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | RangeConstraint | 0..1 | RangeConstraint | (NC) inherited from: GenericValueSchedule |

## 3.56 (NC) ValueOffsetKind enumeration

The kind of the value offset.
Table 55 shows all literals of ValueOffsetKind.

**Table 55 – Literals of StateInstructionScheduleProfile::ValueOffsetKind**

| literal | value | description |
|---|---|---|
| absolute | | Value of the range constraint is replacing the attribute value referenced by the PropertyReference in a determined operational scenario. |
| incremental | | Value of the range constraint is incrementing the attribute value referenced by the PropertyReference in a determined operational scenario. |
| incrementalPercentage | | Value of the range constraint is incrementing in percentage the attribute value referenced by the PropertyReference in a determined operational scenario. |

## 3.57 (NC) PowerRemedialActionSchedule

Inheritance path = BaseIrregularTimeSeries : BaseTimeSeries : IdentifiedObject
The schedule for a power remedial action.
Table 56 shows all attributes of PowerRemedialActionSchedule.

**Table 56 – Attributes of StateInstructionScheduleProfile::PowerRemedialActionSchedule**

| name | mult | type | description |
|---|---|---|---|
| interpolationKind | 1..1 | TimeSeriesInterpolationKind | (NC) inherited from: BaseTimeSeries |
| timeSeriesKind | 0..1 | BaseTimeSeriesKind | (NC) inherited from: BaseTimeSeries |
| actionMethod | 0..1 | String | (NC) inherited from: BaseTimeSeries |
| description | 0..1 | String | inherited from: IdentifiedObject |

| name | mult | type | description |
|---|---|---|---|
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

Table 57 shows all association ends of PowerRemedialActionSchedule with other classes.

**Table 57 – Association ends of StateInstructionScheduleProfile::PowerRemedialActionSchedule with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | PowerRemedialAction | 1..1 | PowerRemedialAction | (NC) Power remedial action for the power remedial action schedule. |

### 3.58 (NC) PowerRemedialActionTimePoint root class

Regulating values at a given point in time.
Table 58 shows all attributes of PowerRemedialActionTimePoint.

**Table 58 – Attributes of StateInstructionScheduleProfile::PowerRemedialActionTimePoint**

| name | mult | type | description |
|---|---|---|---|
| atTime | 1..1 | DateTime | (NC) The time the data is valid for. |
| maxRegulatingDown | 0..1 | ActivePower | (NC) Maximum net amount of active power that the remedial action can regulate down. |
| maxRegulatingUp | 0..1 | ActivePower | (NC) Maximum net amount of active power that the remedial action can regulate up. |

Table 59 shows all association ends of PowerRemedialActionTimePoint with other classes.

**Table 59 – Association ends of StateInstructionScheduleProfile::PowerRemedialActionTimePoint with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 1..* | PowerRemedialActionSchedule | 1..1 | PowerRemedialActionSchedule | (NC) The power remedial action schedule that has this time point. |

### 3.59 (abstract,NC) DCPole root class

The direct current (DC) system pole (IEC 60633) is part of a DC system consisting of all the equipment in the DC substations and the interconnecting transmission lines, if any, which during normal operation exhibit a common direct voltage polarity with respect to earth.

### 3.60 (NC) PowerShiftKeyDistribution

Inheritance path = IdentifiedObject
Distribution of the bid action on the power shift keys.
Table 60 shows all attributes of PowerShiftKeyDistribution.

**Table 60 – Attributes of StateInstructionScheduleProfile::PowerShiftKeyDistribution**

| name | mult | type | description |
|---|---|---|---|
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

Table 61 shows all association ends of PowerShiftKeyDistribution with other classes.

**Table 61 – Association ends of StateInstructionScheduleProfile::PowerShiftKeyDistribution with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..1 | PowerShiftKeySchedule | 1..1 | PowerShiftKeySchedule | (NC) Power Shift Key schedule in power shift key distribution. |
| 0..* | PowerBidSchedule | 0..1 | PowerBidSchedule | (NC) Power bid schedule for the given distribution. |
| 0..* | PowerShiftKeyStrategy | 0..1 | PowerShiftKeyStrategy | (NC) Power Shift Key Strategy which has a Power Shift Key Distribution. |

### 3.61 (abstract,NC) PowerShiftKeyStrategy root class

Strategy of the power shift key.

### 3.62 (abstract) EquivalentInjection root class

This class represents equivalent injections (generation or load). Voltage regulation is allowed only at the point of connection.

### 3.63 (NC) UnitCostSchedule

Inheritance path = BaseIrregularTimeSeries : BaseTimeSeries : IdentifiedObject
The schedule for a unit cost.
Table 62 shows all attributes of UnitCostSchedule.

**Table 62 – Attributes of StateInstructionScheduleProfile::UnitCostSchedule**

| name | mult | type | description |
|---|---|---|---|
| interpolationKind | 1..1 | TimeSeriesInterpolationKind | (NC) inherited from: BaseTimeSeries |
| timeSeriesKind | 0..1 | BaseTimeSeriesKind | (NC) inherited from: BaseTimeSeries |
| actionMethod | 0..1 | String | (NC) inherited from: BaseTimeSeries |
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

Table 63 shows all association ends of UnitCostSchedule with other classes.

**Table 63 – Association ends of StateInstructionScheduleProfile::UnitCostSchedule with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | GeneratingUnit | 1..1 | GeneratingUnit | (NC) GeneratingUnit which has unit cost schedules. |

### 3.64 (NC) UnitCostTimePoint root class

Unit cost at a given point in time.
Table 64 shows all attributes of UnitCostTimePoint.

**Table 64 – Attributes of StateInstructionScheduleProfile::UnitCostTimePoint**

| name | mult | type | description |
|---|---|---|---|
| atTime | 1..1 | DateTime | (NC) The time the data is valid for. |
| startupCost | 0..1 | Money | (NC) The initial startup cost incurred for each start of the GeneratingUnit. |
| warmStartupCost | 0..1 | Money | (NC) The warm startup cost incurred for each start of the GeneratingUnit. |

Table 65 shows all association ends of UnitCostTimePoint with other classes.

**Table 65 – Association ends of StateInstructionScheduleProfile::UnitCostTimePoint with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 1..* | UnitCostSchedule | 1..1 | UnitCostSchedule | (NC) The unit cost schedule that has time point. |

### 3.65 (abstract,NC) BaseRegularIntervalSchedule

Inheritance path = BaseTimeSeries : IdentifiedObject

Time series that has regular points in time.

Table 66 shows all attributes of BaseRegularIntervalSchedule.

**Table 66 – Attributes of StateInstructionScheduleProfile::BaseRegularIntervalSchedule**

| name | mult | type | description |
|---|---|---|---|
| dayOfWeek | 0..1 | DayOfWeekKind | (NC) Day of the week for which the schedule is valid for. |
| intervalStartTime | 0..1 | DateTime | (NC) Interval start time for which the schedule is valid for. |
| intervalEndTime | 0..1 | DateTime | (NC) Interval end time for which the schedule is valid for. |
| interpolationKind | 1..1 | TimeSeriesInterpolationKind | (NC) inherited from: BaseTimeSeries |
| timeSeriesKind | 0..1 | BaseTimeSeriesKind | (NC) inherited from: BaseTimeSeries |
| actionMethod | 0..1 | String | (NC) inherited from: BaseTimeSeries |
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

Table 67 shows all association ends of BaseRegularIntervalSchedule with other classes.

**Table 67 – Association ends of StateInstructionScheduleProfile::BaseRegularIntervalSchedule with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | HourPattern | 0..1 | HourPattern | (NC) HourPattern that has base regular interval schedule. |
| 0..* | Season | 0..1 | Season | (NC) Season associated with a base regular interval schedule. |

## 3.66 Season

Inheritance path = IdentifiedObject
A specified time period of the year.
Table 68 shows all attributes of Season.

**Table 68 – Attributes of StateInstructionScheduleProfile::Season**

| name | mult | type | description |
|---|---|---|---|
| endDate | 1..1 | MonthDay | Date season ends. |
| startDate | 1..1 | MonthDay | Date season starts. |
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

## 3.67 (NC) HourPattern

Inheritance path = IdentifiedObject
Pattern of hourly period in a day with the same kind of intensity.
Table 69 shows all attributes of HourPattern.

**Table 69 – Attributes of StateInstructionScheduleProfile::HourPattern**

| name | mult | type | description |
|---|---|---|---|
| peakKind | 0..1 | PeakKind | (NC) Type of peak or intensity that the pattern is valid for. |
| energyDemandKind | 0..1 | EnergyDemandKind | (NC) Type of energy demand that the pattern is valid for. |
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

## 3.68 (NC) HourPeriod root class

Period of hours in a day.
Table 70 shows all attributes of HourPeriod.

**Table 70 – Attributes of StateInstructionScheduleProfile::HourPeriod**

| name | mult | type | description |
|---|---|---|---|
| mRID | 1..1 | String | (NC) Master resource identifier issued by a model authority. The mRID is unique within an exchange context. Global uniqueness is easily achieved by using a UUID, as specified in RFC 4122, for the mRID. The use of UUID is strongly recommended.<br>For CIMXML data files in RDF syntax conforming to IEC 61970-552, the mRID is mapped to rdf:ID or rdf:about attributes that identify CIM object elements. |
| startTime | 1..1 | Time | (NC) Time the period start and including, e.g. 12:00 which means it include the time of 12:00. |
| endTime | 1..1 | Time | (NC) Time the period end and not including, e.g. 13:00 which means it does not include the time of 13:00 but 12:59. |

Table 71 shows all association ends of HourPeriod with other classes.

**Table 71 – Association ends of StateInstructionScheduleProfile::HourPeriod with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 1..* | HourPattern | 1..1 | HourPattern | (NC) HourPattern which has some hour periods. |

## 3.69 (NC) AssessedElementRegularSchedule

Inheritance path = BaseRegularIntervalSchedule : BaseTimeSeries : IdentifiedObject
Regular schedule for assessed element.
Table 72 shows all attributes of AssessedElementRegularSchedule.

**Table 72 – Attributes of StateInstructionScheduleProfile::AssessedElementRegularSchedule**

| name | mult | type | description |
|---|---|---|---|
| dayOfWeek | 0..1 | DayOfWeekKind | (NC) inherited from: BaseRegularIntervalSchedule |
| intervalStartTime | 0..1 | DateTime | (NC) inherited from: BaseRegularIntervalSchedule |
| intervalEndTime | 0..1 | DateTime | (NC) inherited from: BaseRegularIntervalSchedule |
| interpolationKind | 1..1 | TimeSeriesInterpolationKind | (NC) inherited from: BaseTimeSeries |
| timeSeriesKind | 0..1 | BaseTimeSeriesKind | (NC) inherited from: BaseTimeSeries |
| actionMethod | 0..1 | String | (NC) inherited from: BaseTimeSeries |
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

Table 73 shows all association ends of AssessedElementRegularSchedule with other classes.

**Table 73 – Association ends of StateInstructionScheduleProfile::AssessedElementRegularSchedule with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | AssessedElement | 0..1 | AssessedElement | (NC) Assessed Element that has regular schedules. |
| 0..* | HourPattern | 0..1 | HourPattern | (NC) inherited from: BaseRegularIntervalSchedule |
| 0..* | Season | 0..1 | Season | (NC) inherited from: BaseRegularIntervalSchedule |

## 3.70 (NC) AssessedElementRegularTimePoint root class

Assessed element instruction value at a given point in time.
Table 74 shows all attributes of AssessedElementRegularTimePoint.

**Table 74 – Attributes of StateInstructionScheduleProfile::AssessedElementRegularTimePoint**

| name | mult | type | description |
|---|---|---|---|
| appointedMargin | 0..1 | PerCent | (NC) The percentage (appointed to a region) of the remaining margin obtained in the grid model to reach its current limit. The maximum percentage shall by default be 10% of the remaining margin.<br>It is only used when an assessed element is considered conservative for a region.<br>The allowed value range is [0,100]. |
| maxFlow | 0..1 | ActivePower | (NC) Maximum flow on an a conducting equipment or a collection of conducting equipment forming a power transfer corridor. For assessed elements that is becomes critical due to contingency, this value represents the maximum flow with remedial action taken into consideration. |
| enabled | 0..1 | Boolean | (NC) It identifies if the assessed element is enabled. True means enabled, False means disabled. |
| virtualPositiveMargin | 0..1 | PerCent | (NC) A margin defined only for scanned AssessedElement (If AssessedElement.ScannedForRegion is present) in order to represent the influence of available remedial action which is not cross-border relevant remedial action.<br>The margin is modifying the limits used for the assessment whatever the limit it is (e.g. PATL, TATL).This symbolizes a remedial action that can be applied internally by the System Operator. It will be resolved by the System Operator and not by the optimization of remedial actions. The attribute shall be a positive value.<br>The allowed value range is [0,100]. |
| scannedThresholdMargin | 0..1 | PerCent | (NC) Threshold percentage that a scanned element can be overloaded, on a given element, on top of any overload prior to optimisation (default= 5%). e.g. Initial loading of the element is 110%, with a 5% scanned threshold margin, the new maximum is 115% of the limit (e.g. PATL, TATL, etc).<br>The allowed value range is [0,100]. |

Table 75 shows all association ends of AssessedElementRegularTimePoint with other classes.

**Table 75 – Association ends of StateInstructionScheduleProfile::AssessedElementRegularTimePoint with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | AssessedElementRegularSchedule | 0..1 | AssessedElementRegularSchedule | (NC) Assessed element regular schedule which has assessed element regular time points. |

## 3.71 (NC) DayOfWeekKind enumeration

The kind of day to be included in a regular schedule.
Table 76 shows all literals of DayOfWeekKind.

**Table 76 – Literals of StateInstructionScheduleProfile::DayOfWeekKind**

| literal | value | description |
|---|---|---|
| monday | | Monday as the day of the week. |
| tuesday | | Tuesday as the day of the week. |
| wednesday | | Wednesday as the day of the week. |
| thursday | | Thursday as the day of the week. |
| friday | | Friday as the day of the week. |
| saturday | | Saturday as the day of the week. |
| sunday | | Sunday as the day of the week. |
| weekday | | A day of the week other than Sunday or Saturday. |
| weekend | | A day of the week which is Sunday or Saturday. |
| all | | All days of the week. |
| holiday | | |
| bridgeDay | | A day that is a gap between two distinguished days e.g holiday and weekend that leads to an abnormal scheduling behavior. e.g. if Ascension day falls on a Thursday, then Friday would be a bridge day due to the schedule will not have a normal Friday consumption and production. |

## 3.72 (NC) PeakKind enumeration

Kind of time period with similar intensity.
Table 77 shows all literals of PeakKind.

**Table 77 – Literals of StateInstructionScheduleProfile::PeakKind**

| literal | value | description |
|---|---|---|
| offPeak | | Off-peak refer to periods of lower demand for a particular service or commodity. |
| onPeak | | Off-peak refer to periods of higher demand for a particular service or commodity. |

## 3.73 (NC) EnergyDemandKind enumeration

Kind of energy demand.
Table 78 shows all literals of EnergyDemandKind.

**Table 78 – Literals of StateInstructionScheduleProfile::EnergyDemandKind**

| literal | value | description |
|---|---|---|
| consumption | | |
| production | | |
| storage | | |
| export | | |
| import | | |

## 3.74 (abstract,NC) EnergyGroup root class

An energy group is an aggregation of energy components which have the same energy characteristic, e.g. fuel type and technology. It can be used to allocate energy.

### 3.75 (abstract,NC) EnergyBlockOrder root class

The energy block order is a block (an amount) of energy that forms the sequence of orders that are going to be distributed to an energy block component.

### 3.76 (abstract,NC) AutomationFunction root class

Automation function is a collection of functional block or other automation function that can be executed as a work cycle program as part of an automated system.

### 3.77 (abstract,NC) FunctionBlock root class

Function block is a function described as a set of elementary blocks. The blocks describe the function between input variables and output variables.

### 3.78 (NC) VoltageAngleSchedule

Inheritance path = BaseIrregularTimeSeries : BaseTimeSeries : IdentifiedObject
The schedule for a voltage angle.
Table 79 shows all attributes of VoltageAngleSchedule.

**Table 79 – Attributes of StateInstructionScheduleProfile::VoltageAngleSchedule**

| name | mult | type | description |
|---|---|---|---|
| interpolationKind | 1..1 | TimeSeriesInterpolationKind | (NC) inherited from: BaseTimeSeries |
| timeSeriesKind | 0..1 | BaseTimeSeriesKind | (NC) inherited from: BaseTimeSeries |
| actionMethod | 0..1 | String | (NC) inherited from: BaseTimeSeries |
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

Table 80 shows all association ends of VoltageAngleSchedule with other classes.

**Table 80 – Association ends of StateInstructionScheduleProfile::VoltageAngleSchedule with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | VoltageAngleLimit | 1..1 | VoltageAngleLimit | (NC) Voltage angle limit which has voltage angle schedules. |

### 3.79 (NC) VoltageAngleTimePoint root class

Voltage angle at a given point in time.
Table 81 shows all attributes of VoltageAngleTimePoint.

**Table 81 – Attributes of StateInstructionScheduleProfile::VoltageAngleTimePoint**

| name | mult | type | description |
|---|---|---|---|
| atTime | 1..1 | DateTime | (NC) The time the data is valid for. |
| value | 1..1 | AngleDegrees | (NC) The difference in angle degrees between referenced by the association end OperationalLimitSet.Terminal and the Terminal referenced by the association end VoltageAngleLimit.AngleReferenceTerminal. The value shall be positive (greater than zero). |

Table 82 shows all association ends of VoltageAngleTimePoint with other classes.

**Table 82 – Association ends of StateInstructionScheduleProfile::VoltageAngleTimePoint with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 1..* | VoltageAngleSchedule | 1..1 | VoltageAngleSchedule | (NC) The voltage angle schedule that has time point. |

### 3.80 (abstract,NC) VoltageAngleLimit root class

Voltage angle limit between two terminals. The association end OperationalLimitSet.Terminal defines one end and the host of the limit. The association end VoltageAngleLimit.AngleReferenceTerminal defines the reference terminal.

### 3.81 (NC) InfeedLimitSchedule

Inheritance path = BaseIrregularTimeSeries : BaseTimeSeries : IdentifiedObject

The schedule for an infeed limit.

Table 83 shows all attributes of InfeedLimitSchedule.

**Table 83 – Attributes of StateInstructionScheduleProfile::InfeedLimitSchedule**

| name | mult | type | description |
|---|---|---|---|
| interpolationKind | 1..1 | TimeSeriesInterpolationKind | (NC) inherited from: BaseTimeSeries |
| timeSeriesKind | 0..1 | BaseTimeSeriesKind | (NC) inherited from: BaseTimeSeries |
| actionMethod | 0..1 | String | (NC) inherited from: BaseTimeSeries |
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

Table 84 shows all association ends of InfeedLimitSchedule with other classes.

**Table 84 – Association ends of StateInstructionScheduleProfile::InfeedLimitSchedule with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | InfeedLimit | 1..1 | InfeedLimit | (NC) Infeed limit which has infeed limit schedules. |

### 3.82 (abstract,NC) InfeedLimit root class

Infeed limit set constraints fed in to the network by two or more terminals.

### 3.83 (NC) InfeedLimitTimePoint root class

Infeed limit at a given point in time.

Table 85 shows all attributes of InfeedLimitTimePoint.

**Table 85 – Attributes of StateInstructionScheduleProfile::InfeedLimitTimePoint**

| name | mult | type | description |
|---|---|---|---|
| atTime | 1..1 | DateTime | (NC) The time the data is valid for. |
| valueW | 0..1 | ActivePower | (NC) Value of active power limit. The attribute shall be a positive value or zero. |
| valueA | 0..1 | CurrentFlow | (NC) Value of current limit. The attribute shall be a positive value or zero. |

Table 86 shows all association ends of InfeedLimitTimePoint with other classes.

**Table 86 – Association ends of StateInstructionScheduleProfile::InfeedLimitTimePoint with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 1..* | InfeedLimitSchedule | 1..1 | InfeedLimitSchedule | (NC) Infeed limit schedule that has time point. |

### 3.84 (abstract,NC) PowerTransferCorridor root class

A power transfer corridor is defined as a set of circuits (transmission lines or transformers) separating two portions of the power system, or a subset of circuits exposed to a substantial portion of the transmission exchange between two parts of the system.

### 3.85 (NC) RemedialActionSchemeSchedule

Inheritance path = BaseIrregularTimeSeries : BaseTimeSeries : IdentifiedObject
The schedule for a remedial action scheme.
Table 87 shows all attributes of RemedialActionSchemeSchedule.

**Table 87 – Attributes of StateInstructionScheduleProfile::RemedialActionSchemeSchedule**

| name | mult | type | description |
|---|---|---|---|
| interpolationKind | 1..1 | TimeSeriesInterpolationKind | (NC) inherited from: BaseTimeSeries |
| timeSeriesKind | 0..1 | BaseTimeSeriesKind | (NC) inherited from: BaseTimeSeries |
| actionMethod | 0..1 | String | (NC) inherited from: BaseTimeSeries |
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

Table 88 shows all association ends of RemedialActionSchemeSchedule with other classes.

**Table 88 – Association ends of StateInstructionScheduleProfile::RemedialActionSchemeSchedule with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | ArmedRemedialAction | 1..1 | RemedialActionScheme | (NC) Armed remedial action for a remedial action scheme. |

### 3.86 (abstract,NC) RemedialActionScheme root class

Remedial Action Scheme (RAS), Special Protection Schemes (SPS), System Protection Schemes (SPS) or System Integrity Protection Schemes (SIPS).
A Remedial Action Scheme consists of one or more stages that can trigger and execute a protection action.

### 3.87 (NC) RemedialActionSchemeTimePoint root class

Remedial action scheme at a given point in time.
Table 89 shows all attributes of RemedialActionSchemeTimePoint.

**Table 89 – Attributes of StateInstructionScheduleProfile::RemedialActionSchemeTimePoint**

| name | mult | type | description |
|---|---|---|---|
| armed | 0..1 | Boolean | (NC) Defines the arming status of the remedial action scheme. It is set by operation or by signal. |
| atTime | 1..1 | DateTime | (NC) The time the data is valid for. |
| inService | 0..1 | Boolean | (NC) Specifies the availability of the Remedial Action Scheme (RAS). If true, the RAS is available for contingency processing. If false, the RAS is treated by contingency processing as if it is not in the model. |

Table 90 shows all association ends of RemedialActionSchemeTimePoint with other classes.

**Table 90 – Association ends of StateInstructionScheduleProfile::RemedialActionSchemeTimePoint with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 1..* | RemedialActionScheme Schedule | 1..1 | RemedialActionScheme Schedule | (NC) Remedial action scheme schedule that has time point. |

### 3.88 (abstract,NC) PowerRemedialAction root class

Energy remedial action describes actions to rearrange power schedules.

### 3.89 (NC) RemedialActionGroupSchedule

Inheritance path = BaseIrregularTimeSeries : BaseTimeSeries : IdentifiedObject
The schedule for a remedial action group.
Table 91 shows all attributes of RemedialActionGroupSchedule.

**Table 91 – Attributes of StateInstructionScheduleProfile::RemedialActionGroupSchedule**

| name | mult | type | description |
|---|---|---|---|
| interpolationKind | 1..1 | TimeSeriesInterpolation Kind | (NC) inherited from: BaseTimeSeries |
| timeSeriesKind | 0..1 | BaseTimeSeriesKind | (NC) inherited from: BaseTimeSeries |
| actionMethod | 0..1 | String | (NC) inherited from: BaseTimeSeries |
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

Table 92 shows all association ends of RemedialActionGroupSchedule with other classes.

**Table 92 – Association ends of StateInstructionScheduleProfile::RemedialActionGroupSchedule with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | RemedialActionGroup | 1..1 | RemedialActionGroup | (NC) Remedial action group which has remedial action group schedules. |

### 3.90 (NC) RemedialActionGroupTimePoint root class

Remedial action group at a given point in time.
Table 93 shows all attributes of RemedialActionGroupTimePoint.

**Table 93 – Attributes of StateInstructionScheduleProfile::RemedialActionGroupTimePoint**

| name | mult | type | description |
|---|---|---|---|
| atTime | 1..1 | DateTime | (NC) The time the data is valid for. |
| maxRegulatingDown | 0..1 | ActivePower | (NC) Maximum net amount of active power that the group of remedial actions can regulate down. |
| maxRegulatingUp | 0..1 | ActivePower | (NC) Maximum net amount of active power that the group of remedial actions can regulate up. |

Table 94 shows all association ends of RemedialActionGroupTimePoint with other classes.

**Table 94 – Association ends of StateInstructionScheduleProfile::RemedialActionGroupTimePoint with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 1..* | RemedialActionGroupSchedule | 1..1 | RemedialActionGroupSchedule | (NC) Remedial action group schedule that has time point. |

### 3.91 (abstract,NC) RemedialActionGroup root class

Grouping of remedial actions that can be operated together.

### 3.92 (NC) GridStateAlterationSchedule

Inheritance path = BaseIrregularTimeSeries : BaseTimeSeries : IdentifiedObject
Schedule for a grid state alteration.
Table 95 shows all attributes of GridStateAlterationSchedule.

**Table 95 – Attributes of StateInstructionScheduleProfile::GridStateAlterationSchedule**

| name | mult | type | description |
|---|---|---|---|
| interpolationKind | 1..1 | TimeSeriesInterpolationKind | (NC) inherited from: BaseTimeSeries |
| timeSeriesKind | 0..1 | BaseTimeSeriesKind | (NC) inherited from: BaseTimeSeries |
| actionMethod | 0..1 | String | (NC) inherited from: BaseTimeSeries |
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

Table 96 shows all association ends of GridStateAlterationSchedule with other classes.

**Table 96 – Association ends of StateInstructionScheduleProfile::GridStateAlterationSchedule with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | GridStateAlteration | 1..1 | GridStateAlteration | (NC) Grid state alteration which has grid state alteration schedules. |

## 3.93 (NC) GridStateAlterationTimePoint root class

Grid state alteration at a given point in time.
Table 97 shows all attributes of GridStateAlterationTimePoint.

**Table 97 – Attributes of StateInstructionScheduleProfile::GridStateAlterationTimePoint**

| name | mult | type | description |
|---|---|---|---|
| atTime | 1..1 | DateTime | (NC) The time the data is valid for. |
| enabled | 0..1 | Boolean | (NC) The status of the GridStateAlteration set by an operation or by a signal resulting from a control action. |
| participationFactor | 0..1 | PerCent | (NC) Participation factor describing the entity part of the active power provided by a collection of entities (e.g. an active power forecast to a collection of entities is divided to each of the member entity according to the participation factor). Must be a positive value.<br>In the case of a sharing strategy, the distribution is following entities value (V) equals aggregated value (T) divided by sum of participation factors (PF), i.e. V=T/sum(PF).<br>In the case of priority strategy, the item with the lowest number gets allocated energy first.<br>e.g. If 0 this grid alteration does not participate. The sum of all participation factors for all grid state alterations associated with same remedial action shall be equal to 100%. |

Table 98 shows all association ends of GridStateAlterationTimePoint with other classes.

**Table 98 – Association ends of StateInstructionScheduleProfile::GridStateAlterationTimePoint with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 1..* | GridStateAlterationSchedule | 1..1 | GridStateAlterationSchedule | (NC) Grid state alteration schedule that has time point. |

## 3.94 (abstract,NC) GridStateAlteration root class

Grid state alteration is a change of values describing state (operating point) of one element in the grid model compared to the base case.

## 3.95 Time primitive

Time as "hh:mm:ss.sss", which conforms with ISO 8601. UTC time zone is specified as "hh:mm:ss.sssZ". A local timezone relative UTC is specified as "hh:mm:ss.sss±hh:mm". The second component (shown here as "ss.sss") could have any number of digits in its fractional part to allow any kind of precision beyond seconds.

## 3.96 CurrentFlow datatype

Electrical current with sign convention: positive flow is out of the conducting equipment into the connectivity node. Can be both AC and DC.
Table 99 shows all attributes of CurrentFlow.

**Table 99 – Attributes of StateInstructionScheduleProfile::CurrentFlow**

| name | mult | type | description |
|---|---|---|---|
| multiplier | 0..1 | UnitMultiplier | |
| unit | 0..1 | UnitSymbol | (const=A) |

| name | mult | type | description |
|---|---|---|---|
| value | 0..1 | Float | |

### 3.97 MonthDay primitive

MonthDay format as "--mm-dd", which conforms with XSD data type gMonthDay.

### 3.98 Money datatype

Amount of money.
Table 100 shows all attributes of Money.

**Table 100 – Attributes of StateInstructionScheduleProfile::Money**

| name | mult | type | description |
|---|---|---|---|
| multiplier | 0..1 | UnitMultiplier | |
| unit | 0..1 | Currency | |
| value | 0..1 | Decimal | |

### 3.99 AngleDegrees datatype

Measurement of angle in degrees.
Table 101 shows all attributes of AngleDegrees.

**Table 101 – Attributes of StateInstructionScheduleProfile::AngleDegrees**

| name | mult | type | description |
|---|---|---|---|
| value | 0..1 | Float | |
| unit | 0..1 | UnitSymbol | (const=deg) |
| multiplier | 0..1 | UnitMultiplier | (const=none) |

### 3.100 (abstract,NC) CrossBorderRelevance root class

Combination of an assessed element and one or more bidding zone border that are affected by the assessment.

### 3.101 (NC) SecuredExclusionReasonKind enumeration

The kind of secured exclusion reason.
Table 102 shows all literals of SecuredExclusionReasonKind.

**Table 102 – Literals of StateInstructionScheduleProfile::SecuredExclusionReasonKind**

| literal | value | description |
|---|---|---|
| systemOperator | | The network element that is going to be assessed is excluded for being secured by the system operator. |
| capacityCalculationRegion | | The network element that is going to be assessed is excluded for being secured by the capacity calculation region. |
| nonNativeCapacityCalculationRegion | | The network element that is going to be assessed is excluded for being secured for the native capacity calculation region since it would be secured for a non native capacity calculation region. |

### 3.102 (abstract) EnergySource root class

A generic equivalent for an energy supplier on a transmission or distribution voltage level.

### 3.103 (abstract) ExternalNetworkInjection root class

This class represents the external network and it is used for IEC 60909 calculations.

### 3.104 (NC) StageTriggerSchedule

Inheritance path = BaseIrregularTimeSeries : BaseTimeSeries : IdentifiedObject
Schedule for a stage trigger.
Table 103 shows all attributes of StageTriggerSchedule.

**Table 103 – Attributes of StateInstructionScheduleProfile::StageTriggerSchedule**

| name | mult | type | description |
|---|---|---|---|
| interpolationKind | 1..1 | TimeSeriesInterpolationKind | (NC) inherited from: BaseTimeSeries |
| timeSeriesKind | 0..1 | BaseTimeSeriesKind | (NC) inherited from: BaseTimeSeries |
| actionMethod | 0..1 | String | (NC) inherited from: BaseTimeSeries |
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

Table 104 shows all association ends of StageTriggerSchedule with other classes.

**Table 104 – Association ends of StateInstructionScheduleProfile::StageTriggerSchedule with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | StageTrigger | 1..1 | StageTrigger | (NC) Stage trigger which has stage trigger schedules. |

### 3.105 (NC) StageTriggerTimePoint root class

Stage trigger values at a given point in time.
Table 105 shows all attributes of StageTriggerTimePoint.

**Table 105 – Attributes of StateInstructionScheduleProfile::StageTriggerTimePoint**

| name | mult | type | description |
|---|---|---|---|
| armed | 0..1 | Boolean | (NC) Defines the arming status of the remedial action scheme. It is set by operation or by signal. |
| atTime | 1..1 | DateTime | (NC) The time the data is valid for. |
| inService | 0..1 | Boolean | (NC) Specifies the availability of the Remedial Action Scheme (RAS). If true, the RAS is available for contingency processing. If false, the RAS is treated by contingency processing as if it is not in the model. |

Table 106 shows all association ends of StageTriggerTimePoint with other classes.

**Table 106 – Association ends of StateInstructionScheduleProfile::StageTriggerTimePoint with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 1..* | StageTriggerSchedule | 1..1 | StageTriggerSchedule | (NC) Stage trigger schedule that has time point. |

### 3.106 (abstract,NC) StageTrigger root class

Stage that is triggered either by TriggerCondition or by gate condition within a stage.

### 3.107 (NC) GenericSequenceSchedule

Inheritance path = BaseIrregularTimeSeries : BaseTimeSeries : IdentifiedObject
Schedule for sequence.
Table 107 shows all attributes of GenericSequenceSchedule.

**Table 107 – Attributes of StateInstructionScheduleProfile::GenericSequenceSchedule**

| name | mult | type | description |
|---|---|---|---|
| interpolationKind | 1..1 | TimeSeriesInterpolationKind | (NC) inherited from: BaseTimeSeries |
| timeSeriesKind | 0..1 | BaseTimeSeriesKind | (NC) inherited from: BaseTimeSeries |
| actionMethod | 0..1 | String | (NC) inherited from: BaseTimeSeries |
| description | 0..1 | String | inherited from: IdentifiedObject |
| mRID | 1..1 | String | inherited from: IdentifiedObject |
| name | 0..1 | String | inherited from: IdentifiedObject |

Table 108 shows all association ends of GenericSequenceSchedule with other classes.

**Table 108 – Association ends of StateInstructionScheduleProfile::GenericSequenceSchedule with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 0..* | EnergyBlockOrder | 0..1 | EnergyBlockOrder | (NC) Energy block order which has generic sequence schedules. |

### 3.108 (NC) SequenceTimePoint root class

Sequence at a given point in time.
Table 109 shows all attributes of SequenceTimePoint.

**Table 109 – Attributes of StateInstructionScheduleProfile::SequenceTimePoint**

| name | mult | type | description |
|---|---|---|---|
| atTime | 1..1 | DateTime | (NC) The time the data is valid for. |
| sequence | 1..1 | Integer | (NC) Sequence needs to be ordered by the scheduling area. It has to be unique by the scheduling area. |

Table 110 shows all association ends of SequenceTimePoint with other classes.

**Table 110 – Association ends of StateInstructionScheduleProfile::SequenceTimePoint with other classes**

| mult from | name | mult to | type | description |
|---|---|---|---|---|
| 1..* | GenericSequenceSchedule | 1..1 | GenericSequenceSchedule | (NC) The sequence schedule which belongs to the sequence timepoint. |

1299

# Annex A (informative): Sample data

## A.1 General

This Annex is designed to illustrate the profile by using fragments of sample data. It is not meant to be a complete set of examples covering all possibilities of using the profile. Defining a complete set of test data is considered a separate activity to be performed for the purpose of setting up interoperability testing and conformity related to this profile.

## A.2 Sample instance data

Test data files are available in the CIM EG SharePoint.