European Network of
Transmission System Operators
for Electricity

STATE INSTRUCTION SCHEDULE PROFILE SPECIFICATION

2023-05-10

APPROVED DOCUMENT
VERSION 2.2

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	Release	Date	Paragraph	Comments
2	2	2023-03-24		For review.
2	2	2023-05-10		ICTC approved.

CONTENTS

Copyright notice:....2
Revision History....3
CONTENTS....4
1 Introduction....8
2 Application profile specification....8
2.1 Version information....8
2.2 Constraints naming convention....8
2.3 Profile constraints....9
2.4 Metadata....11
2.4.1 Constraints....11
2.4.2 Reference metadata....11
3 Detailed Profile Specification....11
3.1 General....11
3.2 (abstract,NC) AssessedElement....14
3.3 (NC) AssessedElementSchedule....14
3.4 (NC) AssessedElementTimePoint root class....15
3.5 (abstract,NC) AssessedElementWithContingency root class....16
3.6 (abstract,NC) AssessedElementWithRemedialAction root class....16
3.7 (NC) AvailabilityTimePoint root class....16
3.8 (abstract,NC) BaseIrregularTimeSeries....16
3.9 (abstract,NC) BaseTimeSeries....17
3.10 (abstract) ConductingEquipment....17
3.11 (abstract) Contingency root class....17
3.12 (NC) ContingencySchedule....17
3.13 (NC) ContingencyTimePoint root class....18
3.14 (abstract,NC) ContingencyWithRemedialAction root class....18
3.15 (NC) EnablingTimePoint root class....18
3.16 (abstract,NC) EnergyBlockOrder root class....19
3.17 (abstract) EnergyConnection....19
3.18 (abstract) EnergyConsumer root class....19
3.19 (abstract,NC) EnergyGroup root class....19
3.20 (abstract) Equipment....19
3.21 (abstract) GeneratingUnit root class....19
3.22 (NC) GenericAvailableSchedule....19
3.23 (NC) GenericEnablingSchedule....20
3.24 (NC) GenericValueSchedule....21
3.25 (NC) GenericValueTimePoint root class....21
3.26 (NC) GLSKBidActionDistribution....21
3.27 (NC) GLSKSchedule....22
3.28 (abstract,NC) GridStateAlteration....23
3.29 (NC) GridStateIntensitySchedule....23
3.30 (abstract) HydroPump root class....24
3.31 (abstract) IdentifiedObject root class....24

3.32 (NC) ParticipationFactorTimePoint root class 24
3.33 (NC) PowerBidDependency root class 25
3.34 (NC) PowerBidSchedule 25
3.35 (NC) PowerBidScheduleTimePoint root class 26
3.36 (abstract) PowerElectronicsUnit root class 27
3.37 (abstract,NC) PowerRemedialAction root class 27
3.38 (abstract) PowerSystemResource 27
3.39 (abstract,NC) RangeConstraint 27
3.40 (abstract,NC) RemedialAction 27
3.41 (abstract,NC) RemedialActionDependency root class 28
3.42 (abstract,NC) ScheduleResource root class 28
3.43 (NC) BaseTimeSeriesKind enumeration 28
3.44 (NC) BidDirectionKind enumeration 28
3.45 Currency enumeration 29
3.46 (NC) PowerBidDependencyKind enumeration 33
3.47 (NC) TimeSeriesInterpolationKind enumeration 33
3.48 UnitMultiplier enumeration 33
3.49 UnitSymbol enumeration 34
3.50 (NC) ValueOffsetKind enumeration 34
3.51 ActivePower datatype 35
3.52 PerCent datatype 35
3.53 RealEnergy datatype 35
3.54 Seconds datatype 35
3.55 Boolean primitive 36
3.56 DateTime primitive 36
3.57 Decimal primitive 36
3.58 Duration primitive 36
3.59 Float primitive 36
3.60 Integer primitive 36
3.61 String primitive 36
Annex A (informative): Sample data 37
A.1 General 37
A.2 Sample instance data 37

List of figures

Figure 1 – Class diagram StateInstructionScheduleProfile::GLSKSchedule 12
Figure 2 – Class diagram StateInstructionScheduleProfile::Core 13
Figure 3 – Class diagram StateInstructionScheduleProfile::SecuritySchedule 13
Figure 4 – Class diagram StateInstructionScheduleProfile::PowerBidSchedule 14

List of tables

Table 1 – Attributes of StateInstructionScheduleProfile::AssessedElement 14
Table 2 – Attributes of StateInstructionScheduleProfile::AssessedElementSchedule 14

Table 3 – Association ends of StateInstructionScheduleProfile::AssessedElementSchedule with other classes 15
Table 4 – Attributes of StateInstructionScheduleProfile::AssessedElementTimePoint 15
Table 5 – Association ends of StateInstructionScheduleProfile::AssessedElementTimePoint with other classes 16
Table 6 – Attributes of StateInstructionScheduleProfile::AvailabilityTimePoint 16
Table 7 – Association ends of StateInstructionScheduleProfile::AvailabilityTimePoint with other classes 16
Table 8 – Attributes of StateInstructionScheduleProfile::BaseIrregularTimeSeries 16
Table 9 – Attributes of StateInstructionScheduleProfile::BaseTimeSeries 17
Table 10 – Attributes of StateInstructionScheduleProfile::ConductingEquipment 17
Table 11 – Attributes of StateInstructionScheduleProfile::ContingencySchedule 17
Table 12 – Association ends of StateInstructionScheduleProfile::ContingencySchedule with other classes 18
Table 13 – Attributes of StateInstructionScheduleProfile::ContingencyTimePoint 18
Table 14 – Association ends of StateInstructionScheduleProfile::ContingencyTimePoint with other classes 18
Table 15 – Attributes of StateInstructionScheduleProfile::EnablingTimePoint 18
Table 16 – Association ends of StateInstructionScheduleProfile::EnablingTimePoint with other classes 18
Table 17 – Attributes of StateInstructionScheduleProfile::EnergyConnection 19
Table 18 – Attributes of StateInstructionScheduleProfile::Equipment 19
Table 19 – Attributes of StateInstructionScheduleProfile::GenericAvailableSchedule 20
Table 20 – Association ends of StateInstructionScheduleProfile::GenericAvailableSchedule with other classes 20
Table 21 – Attributes of StateInstructionScheduleProfile::GenericEnablingSchedule 20
Table 22 – Association ends of StateInstructionScheduleProfile::GenericEnablingSchedule with other classes 20
Table 23 – Attributes of StateInstructionScheduleProfile::GenericValueSchedule 21
Table 24 – Association ends of StateInstructionScheduleProfile::GenericValueSchedule with other classes 21
Table 25 – Attributes of StateInstructionScheduleProfile::GenericValueTimePoint 21
Table 26 – Association ends of StateInstructionScheduleProfile::GenericValueTimePoint with other classes 21
Table 27 – Attributes of StateInstructionScheduleProfile::GLSKBidActionDistribution 22
Table 28 – Association ends of StateInstructionScheduleProfile::GLSKBidActionDistribution with other classes 22
Table 29 – Attributes of StateInstructionScheduleProfile::GLSKSchedule 22
Table 30 – Association ends of StateInstructionScheduleProfile::GLSKSchedule with other classes 22
Table 31 – Attributes of StateInstructionScheduleProfile::GridStateAlteration 23
Table 32 – Attributes of StateInstructionScheduleProfile::GridStateIntensitySchedule 23
Table 33 – Association ends of StateInstructionScheduleProfile::GridStateIntensitySchedule with other classes 23

Table 34 – Attributes of StateInstructionScheduleProfile::IdentifiedObject 24
Table 35 – Attributes of StateInstructionScheduleProfile::ParticipationFactorTimePoint 24
Table 36 – Association ends of StateInstructionScheduleProfile::ParticipationFactorTimePoint with other classes 24
Table 37 – Attributes of StateInstructionScheduleProfile::PowerBidDependency 25
Table 38 – Association ends of StateInstructionScheduleProfile::PowerBidDependency with other classes 25
Table 39 – Attributes of StateInstructionScheduleProfile::PowerBidSchedule 25
Table 40 – Association ends of StateInstructionScheduleProfile::PowerBidSchedule with other classes 26
Table 41 – Attributes of StateInstructionScheduleProfile::PowerBidScheduleTimePoint 26
Table 42 – Association ends of StateInstructionScheduleProfile::PowerBidScheduleTimePoint with other classes 27
Table 43 – Attributes of StateInstructionScheduleProfile::PowerSystemResource 27
Table 44 – Attributes of StateInstructionScheduleProfile::RangeConstraint 27
Table 45 – Attributes of StateInstructionScheduleProfile::RemedialAction 27
Table 46 – Association ends of StateInstructionScheduleProfile::ScheduleResource with other classes 28
Table 47 – Literals of StateInstructionScheduleProfile::BaseTimeSeriesKind 28
Table 48 – Literals of StateInstructionScheduleProfile::BidDirectionKind 28
Table 49 – Literals of StateInstructionScheduleProfile::Currency 29
Table 50 – Literals of StateInstructionScheduleProfile::PowerBidDependencyKind 33
Table 51 – Literals of StateInstructionScheduleProfile::TimeSeriesInterpolationKind 33
Table 52 – Literals of StateInstructionScheduleProfile::UnitMultiplier 34
Table 53 – Literals of StateInstructionScheduleProfile::UnitSymbol 34
Table 54 – Literals of StateInstructionScheduleProfile::ValueOffsetKind 35
Table 55 – Attributes of StateInstructionScheduleProfile::ActivePower 35
Table 56 – Attributes of StateInstructionScheduleProfile::PerCent 35
Table 57 – Attributes of StateInstructionScheduleProfile::RealEnergy 35
Table 58 – Attributes of StateInstructionScheduleProfile::Seconds 36

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 CIM100_CGMES31v01_501-20v02_NC22v95_MM10v01.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: http://entsoe.eu/ns/CIM/StateInstructionSchedule-EU/2.2
- Version info: 2.2.0
- Prior version:
- 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://CGMES-30v25_501-20v01.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.

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 (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.

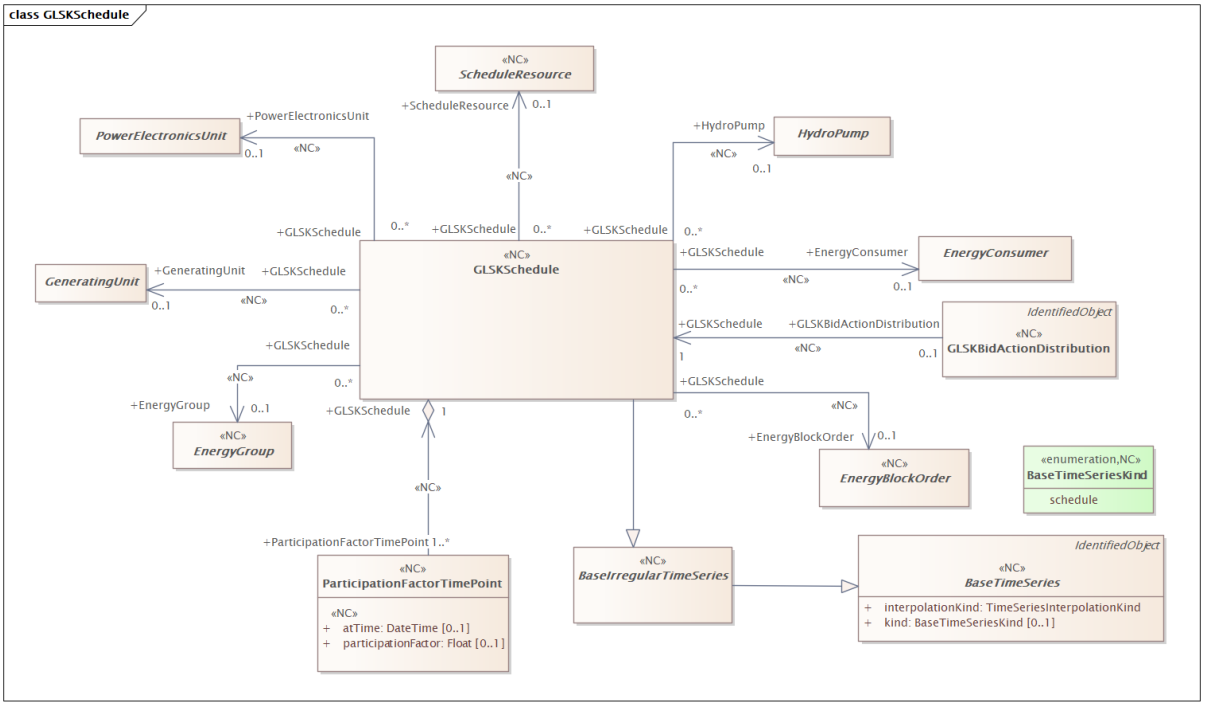

Figure 1 – Class diagram StateInstructionScheduleProfile::GLSKSchedule

Figure 1: The diagram shows generation and load shift keys related classes.

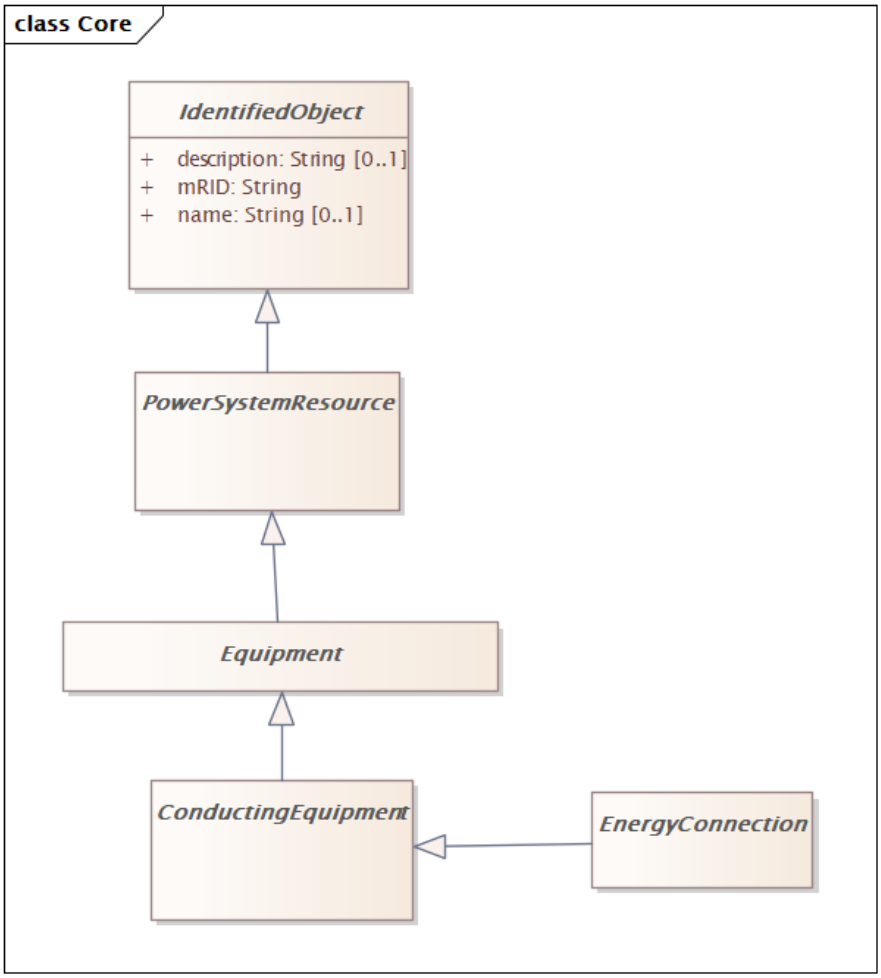

Figure 2 – Class diagram StateInstructionScheduleProfile::Core

Figure 2: The diagram shows classes from Base CIM used in the security schedule profile.

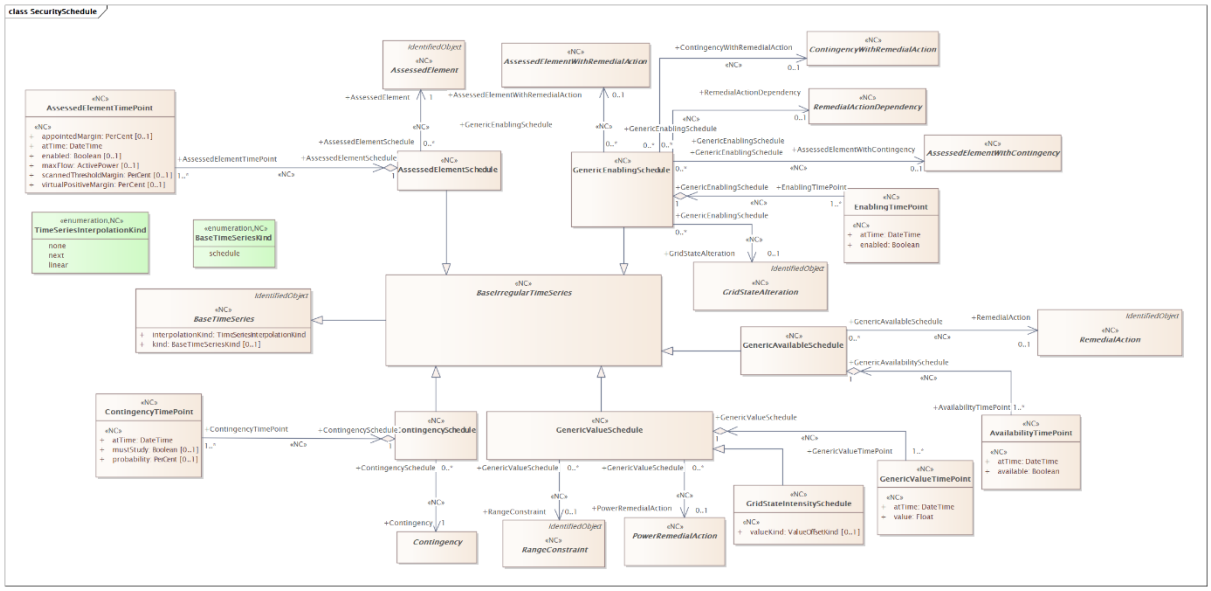

Figure 3 – Class diagram StateInstructionScheduleProfile::SecuritySchedule

Figure 3: The diagram shows security schedule related classes.

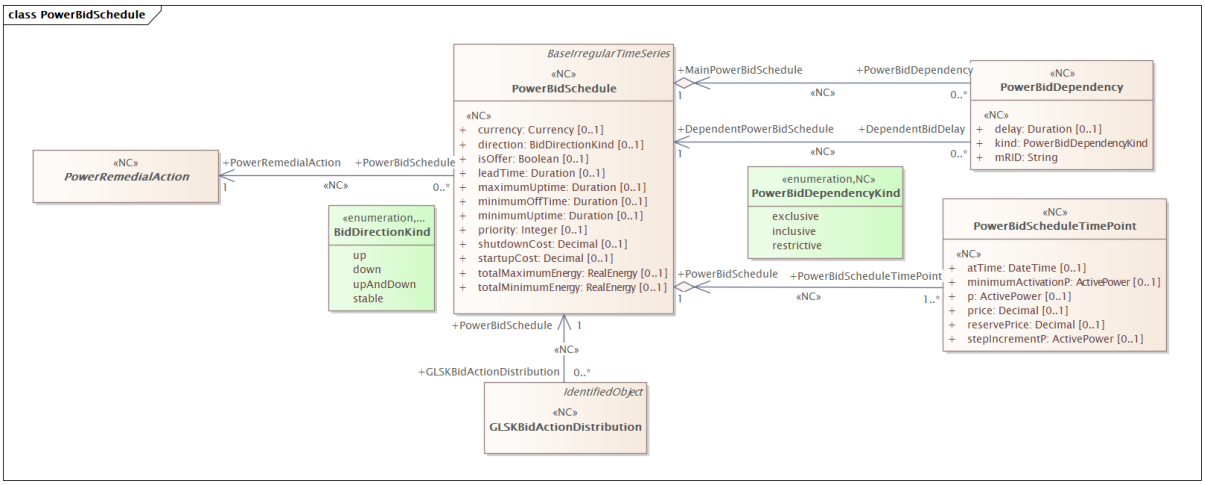

Figure 4 – Class diagram StateInstructionScheduleProfile::PowerBidSchedule

Figure 4: The diagram shows power bid schedule related classes. The power bid schedule part of the security schedule profile shall be used only for Coordinated Security Analysis (CSA) 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.

3.2 (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 1 shows all attributes of AssessedElement.

Table 1 – 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

3.3 (NC) AssessedElementSchedule

Inheritance path = BaseIrregularTimeSeries : BaseTimeSeries : IdentifiedObject
The schedule for Assessed Element.
Table 2 shows all attributes of AssessedElementSchedule.

Table 2 – Attributes of StateInstructionScheduleProfile::AssessedElementSchedule

name	mult	type	description
interpolationKind	1..1	TimeSeriesInterpolationKind	inherited from: BaseTimeSeries
kind	0..1	BaseTimeSeriesKind	inherited from: BaseTimeSeries
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

Table 3 shows all association ends of AssessedElementSchedule with other classes.

Table 3 – Association ends of 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.

3.4 (NC) AssessedElementTimePoint root class

Assessed element instruction value at a given point in time.

Table 4 shows all attributes of AssessedElementTimePoint.

Table 4 – Attributes of StateInstructionScheduleProfile::AssessedElementTimePoint

name	mult	type	description
atTime	1..1	DateTime	(NC) The time the data is valid for.
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. It is only used when an assessed element is considered conservative for a region. 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. 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. 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).

name	mult	type	description
			The allowed value range is [0,100].

Table 5 shows all association ends of AssessedElementTimePoint with other classes.

Table 5 – 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.

3.5 (abstract,NC) AssessedElementWithContingency root class

Combination of an assessed element and a contingency.

3.6 (abstract,NC) AssessedElementWithRemedialAction root class

Combination of an assessed element and a remedial action

3.7 (NC) AvailabilityTimePoint root class

Availability instruction value at a given point in time.
Table 6 shows all attributes of AvailabilityTimePoint.

Table 6 – 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 7 shows all association ends of AvailabilityTimePoint with other classes.

Table 7 – 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.

3.8 (abstract,NC) BaseIrregularTimeSeries

Inheritance path = BaseTimeSeries : IdentifiedObject
Time series that has irregular points in time.
Table 8 shows all attributes of BaseIrregularTimeSeries.

Table 8 – Attributes of StateInstructionScheduleProfile::BaseIrregularTimeSeries

name	mult	type	description
interpolationKind	1..1	TimeSeriesInterpolationKind	inherited from: BaseTimeSeries
kind	0..1	BaseTimeSeriesKind	inherited from: BaseTimeSeries
description	0..1	String	inherited from: IdentifiedObject
mRID	1..1	String	inherited from: IdentifiedObject
name	0..1	String	inherited from: IdentifiedObject

3.9 (abstract,NC) BaseTimeSeries

Inheritance path = IdentifiedObject
Time series of values at points in time.
Table 9 shows all attributes of BaseTimeSeries.

Table 9 – Attributes of StateInstructionScheduleProfile::BaseTimeSeries

name	mult	type	description
interpolationKind	1..1	TimeSeriesInterpolationKind	Kind of interpolation done between time point.
kind	0..1	BaseTimeSeriesKind	Kind of base time series.
description	0..1	String	inherited from: IdentifiedObject
mRID	1..1	String	inherited from: IdentifiedObject
name	0..1	String	inherited from: IdentifiedObject

3.10 (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 10 shows all attributes of ConductingEquipment.

Table 10 – 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

3.11 (abstract) Contingency root class

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

3.12 (NC) ContingencySchedule

Inheritance path = BaseIrregularTimeSeries : BaseTimeSeries : IdentifiedObject
The schedule for Contingency.
Table 11 shows all attributes of ContingencySchedule.

Table 11 – Attributes of StateInstructionScheduleProfile::ContingencySchedule

name	mult	type	description
interpolationKind	1..1	TimeSeriesInterpolationKind	inherited from: BaseTimeSeries
kind	0..1	BaseTimeSeriesKind	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 12 shows all association ends of ContingencySchedule with other classes.

Table 12 – 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.

3.13 (NC) ContingencyTimePoint root class

Contingency instruction value at a given point in time.
Table 13 shows all attributes of ContingencyTimePoint.

Table 13 – 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 14 shows all association ends of ContingencyTimePoint with other classes.

Table 14 – Association ends of StateInstructionScheduleProfile::ContingencyTimePoint with other classes

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

3.14 (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).

3.15 (NC) EnablingTimePoint root class

Enabling instruction value at a given point in time.
Table 15 shows all attributes of EnablingTimePoint.

Table 15 – 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 16 shows all association ends of EnablingTimePoint with other classes.

Table 16 – 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.

3.16 (abstract,NC) EnergyBlockOrder root class

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

3.17 (abstract) EnergyConnection

Inheritance path = ConductingEquipment : Equipment : PowerSystemResource : IdentifiedObject
A connection of energy generation or consumption on the power system model.
Table 17 shows all attributes of EnergyConnection.

Table 17 – 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

3.18 (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.

3.19 (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.

3.20 (abstract) Equipment

Inheritance path = PowerSystemResource : IdentifiedObject
The parts of a power system that are physical devices, electronic or mechanical.
Table 18 shows all attributes of Equipment.

Table 18 – 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

3.21 (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.

3.22 (NC) GenericAvailableSchedule

Inheritance path = BaseIrregularTimeSeries : BaseTimeSeries : IdentifiedObject
The schedule for the availability of elements.
Table 19 shows all attributes of GenericAvailableSchedule.

Table 19 – Attributes of StateInstructionScheduleProfile::GenericAvailableSchedule

name	mult	type	description
interpolationKind	1..1	TimeSeriesInterpolationKind	inherited from: BaseTimeSeries
kind	0..1	BaseTimeSeriesKind	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 20 shows all association ends of GenericAvailableSchedule with other classes.

Table 20 – 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.

3.23 (NC) GenericEnablingSchedule

Inheritance path = BaseIrregularTimeSeries : BaseTimeSeries : IdentifiedObject

The schedule for the enabling of elements.

Table 21 shows all attributes of GenericEnablingSchedule.

Table 21 – Attributes of StateInstructionScheduleProfile::GenericEnablingSchedule

name	mult	type	description
interpolationKind	1..1	TimeSeriesInterpolationKind	inherited from: BaseTimeSeries
kind	0..1	BaseTimeSeriesKind	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 22 shows all association ends of GenericEnablingSchedule with other classes.

Table 22 – 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..*	GridStateAlteration	0..1	GridStateAlteration	(NC) Grid state alteration which has enabling schedules.
0..*	RemedialActionDependency	0..1	RemedialActionDependency	(NC) Remedial action dependency which has enabling schedules.

3.24 (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 23 shows all attributes of GenericValueSchedule.

Table 23 – Attributes of StateInstructionScheduleProfile::GenericValueSchedule

name	mult	type	description
interpolationKind	1..1	TimeSeriesInterpolationKind	inherited from: BaseTimeSeries
kind	0..1	BaseTimeSeriesKind	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 24 shows all association ends of GenericValueSchedule with other classes.

Table 24 – 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.
0..*	PowerRemedialAction	0..1	PowerRemedialAction	(NC) Power remedial action for the generic value schedule.

3.25 (NC) GenericValueTimePoint root class

Generic value for a given point in time.
Table 25 shows all attributes of GenericValueTimePoint.

Table 25 – 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 26 shows all association ends of GenericValueTimePoint with other classes.

Table 26 – 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.

3.26 (NC) GLSKBidActionDistribution

Inheritance path = IdentifiedObject
Distribution of the bid action on the generator and load shift keys.

Table 27 shows all attributes of GLSKBidActionDistribution.

Table 27 – Attributes of StateInstructionScheduleProfile::GLSKBidActionDistribution

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 28 shows all association ends of GLSKBidActionDistribution with other classes.

Table 28 – Association ends of StateInstructionScheduleProfile::GLSKBidActionDistribution with other classes

mult from	name	mult to	type	description
0..1	GLSKSchedule	1..1	GLSKSchedule	(NC) GLSK schedule in GLSK bid action distribution.
0..*	PowerBidSchedule	1..1	PowerBidSchedule	(NC) Power bid schedule for the given distribution.

3.27 (NC) GLSKSchedule

Inheritance path = BaseIrregularTimeSeries : BaseTimeSeries : IdentifiedObject
The schedule for Generation and Load Shift Keys (GLSK).
Table 29 shows all attributes of GLSKSchedule.

Table 29 – Attributes of StateInstructionScheduleProfile::GLSKSchedule

name	mult	type	description
interpolationKind	1..1	TimeSeriesInterpolationKind	inherited from: BaseTimeSeries
kind	0..1	BaseTimeSeriesKind	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 30 shows all association ends of GLSKSchedule with other classes.

Table 30 – Association ends of StateInstructionScheduleProfile::GLSKSchedule with other classes

mult from	name	mult to	type	description
0..*	EnergyBlockOrder	0..1	EnergyBlockOrder	(NC) A EnergyBlockOrder which has a GLSK Schedule.
0..*	EnergyConsumer	0..1	EnergyConsumer	(NC) The EnergyConsumer that has a GLSK schedule.
0..*	EnergyGroup	0..1	EnergyGroup	(NC) The EnergyGroup which has a GLSK Schedule.
0..*	GeneratingUnit	0..1	GeneratingUnit	(NC) The Generating Unit which a GLSK Schedule.
0..*	ScheduleResource	0..1	ScheduleResource	(NC) The Schedule Resource that has a GLSK schedule.

mult from	name	mult to	type	description
0..*	HydroPump	0..1	HydroPump	(NC) The Hydro Pump which has a GLSK schedule.
0..*	PowerElectronicsUnit	0..1	PowerElectronicsUnit	(NC) The Power Electronics Unit which has a GLSK schedule.

3.28 (abstract,NC) GridStateAlteration

Inheritance path = IdentifiedObject

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

Table 31 shows all attributes of GridStateAlteration.

Table 31 – Attributes of StateInstructionScheduleProfile::GridStateAlteration

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

3.29 (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 32 shows all attributes of GridStateIntensitySchedule.

Table 32 – Attributes of StateInstructionScheduleProfile::GridStateIntensitySchedule

name	mult	type	description
valueKind	0..1	ValueOffsetKind	(NC) The kind of value1 and value2 of the associated IrregularIntervalSchedule.
interpolationKind	1..1	TimeSeriesInterpolationKind	inherited from: BaseTimeSeries
kind	0..1	BaseTimeSeriesKind	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 GridStateIntensitySchedule with other classes.

Table 33 – Association ends of StateInstructionScheduleProfile::GridStateIntensitySchedule with other classes

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

3.30 (abstract) HydroPump root class

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

3.31 (abstract) IdentifiedObject root class

This is a root class to provide common identification for all classes needing identification and naming attributes.
Table 34 shows all attributes of IdentifiedObject.

Table 34 – 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. 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.

3.32 (NC) ParticipationFactorTimePoint root class

Participation factor for a given point in time.
Table 35 shows all attributes of ParticipationFactorTimePoint.

Table 35 – 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. 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). In the case of priority strategy, the item with the lowest number gets allocated energy first.

Table 36 shows all association ends of ParticipationFactorTimePoint with other classes.

Table 36 – Association ends of StateInstructionScheduleProfile::ParticipationFactorTimePoint with other classes

mult from	name	mult to	type	description
1..*	GLSKSchedule	1..1	GLSKSchedule	(NC) The GLSK schedule which belongs to the participation factor timepoint.

3.33 (NC) PowerBidDependency root class

Dependency between the related power bids.
Table 37 shows all attributes of PowerBidDependency.

Table 37 – 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. 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 38 shows all association ends of PowerBidDependency with other classes.

Table 38 – 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.

3.34 (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.
Table 39 shows all attributes of PowerBidSchedule.

Table 39 – Attributes of StateInstructionScheduleProfile::PowerBidSchedule

name	mult	type	description
isOffer	0..1	Boolean	(NC) Indicates if the energy 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.

name	mult	type	description
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.
startupCost	0..1	Decimal	(NC) Total startup cost incurred for all the units involved in the bid. This overrides any cost on the specific unit.
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.
interpolationKind	1..1	TimeSeriesInterpolationKind	inherited from: BaseTimeSeries
kind	0..1	BaseTimeSeriesKind	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 40 shows all association ends of PowerBidSchedule with other classes.

Table 40 – 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	1..1	PowerRemedialAction	(NC) Power remedial action for which the bid is given.

3.35 (NC) PowerBidScheduleTimePoint root class

Time series represent irregular power, active and reactive, values at given points in time.
Table 41 shows all attributes of PowerBidScheduleTimePoint.

Table 41 – Attributes of StateInstructionScheduleProfile::PowerBidScheduleTimePoint

name	mult	type	description
atTime	0..1	DateTime	(NC) The time the data is valid for.
price	0..1	Decimal	(NC) Quantity given in the time points.
p	0..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 42 shows all association ends of PowerBidScheduleTimePoint with other classes.

608 **Table 42 – Association ends of**
609 **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.

610

611 ### 3.36 (abstract) PowerElectronicsUnit root class

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

614 ### 3.37 (abstract,NC) PowerRemedialAction root class

615 Energy remedial action describes actions to rearrange power schedules.

616 ### 3.38 (abstract) PowerSystemResource

617 Inheritance path = IdentifiedObject
618 A power system resource (PSR) can be an item of equipment such as a switch, an equipment
619 container containing many individual items of equipment such as a substation, or an
620 organisational entity such as sub-control area. Power system resources can have
621 measurements associated.
622 Table 43 shows all attributes of PowerSystemResource.

623 **Table 43 – Attributes of StateInstructionScheduleProfile::PowerSystemResource**

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

624

625 ### 3.39 (abstract,NC) RangeConstraint

626 Inheritance path = IdentifiedObject
627 Defines the rage constraint.
628 Table 44 shows all attributes of RangeConstraint.

629 **Table 44 – 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

630

631 ### 3.40 (abstract,NC) RemedialAction

632 Inheritance path = IdentifiedObject
633 Remedial action describes one or more actions that can be performed on a given power system
634 model situation to eliminate one or more identified breaches of constraints. The remedial action
635 can be costly, and have a cost characteristic, or non costly.
636 Table 45 shows all attributes of RemedialAction.

637 **Table 45 – Attributes of StateInstructionScheduleProfile::RemedialAction**

name	mult	type	description
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

3.41 (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.

3.42 (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 46 shows all association ends of ScheduleResource with other classes.

Table 46 – 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.

3.43 (NC) BaseTimeSeriesKind enumeration

Kind of time series.
Table 47 shows all literals of BaseTimeSeriesKind.

Table 47 – 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.

3.44 (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.

literal	value	description
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.

668

669 **3.45 Currency enumeration**

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

672 **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.
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.

literal	value	description
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.
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.

literal	value	description
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.
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.

literal	value	description
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.
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.

literal	value	description
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.

3.46 (NC) PowerBidDependencyKind enumeration

Kind of power bid dependency.
Table 50 shows all literals of PowerBidDependencyKind.

Table 50 – 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.

3.47 (NC) TimeSeriesInterpolationKind enumeration

Kinds of interpolation of values between two time point.
Table 51 shows all literals of TimeSeriesInterpolationKind.

Table 51 – Literals of StateInstructionScheduleProfile::TimeSeriesInterpolationKind

literal	value	description
none		No interpolation is applied.
next		The value between two time points is set to next value.
linear		Linear interpolation is applied for values between two time points.

3.48 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 52 shows all literals of UnitMultiplier.

Table 52 – Literals of StateInstructionScheduleProfile::UnitMultiplier

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

3.49 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 53 shows all literals of UnitSymbol.

Table 53 – 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 (I^2R or VIcos(phi)), is expressed in Watts. See also apparent power and reactive power.
Wh	72	Real energy in watt hours.

3.50 (NC) ValueOffsetKind enumeration

The kind of the value offset.
Table 54 shows all literals of ValueOffsetKind.

Table 54 – 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.

3.51 ActivePower datatype

Product of RMS value of the voltage and the RMS value of the in-phase component of the current.
Table 55 shows all attributes of ActivePower.

Table 55 – Attributes of StateInstructionScheduleProfile::ActivePower

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

3.52 PerCent datatype

Percentage on a defined base. For example, specify as 100 to indicate at the defined base.
Table 56 shows all attributes of PerCent.

Table 56 – 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)
multiplier	0..1	UnitMultiplier	(const=none)

3.53 RealEnergy datatype

Real electrical energy.
Table 57 shows all attributes of RealEnergy.

Table 57 – Attributes of StateInstructionScheduleProfile::RealEnergy

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

3.54 Seconds datatype

Time, in seconds.

Table 58 shows all attributes of Seconds.

Table 58 – 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)

3.55 Boolean primitive

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

3.56 DateTime primitive

Date and time as "yyyy-mm-ddThh:mm:ss.sss", which conforms with ISO 8601. UTC time zone is specified as "yyyy-mm-ddThh:mm:ss.sssZ". A local timezone relative UTC is specified as "yyyy-mm-ddThh: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.

3.57 Decimal primitive

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

3.58 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.

3.59 Float primitive

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

3.60 Integer primitive

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

3.61 String primitive

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

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.