



European Network of
Transmission System Operators
for Electricity

STATE INSTRUCTION SCHEDULE PROFILE SPECIFICATION

2025-09-11

ICTC APPROVED
VERSION 2.4.0

Copyright notice:

Copyright © ENTSO-E. All Rights Reserved.

This document and its whole translations may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, except for literal and whole translation into languages other than English and under all circumstances, the copyright notice or references to ENTSO-E may not be removed.

This document and the information contained herein is provided on an "as is" basis.

ENTSO-E DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

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.

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.
2.3.2-alpha	2024-01-04		For CIM WG review.
2.4.0	2025-09-11		ICTC approval.

34 CONTENTS

35	Copyright notice:.....	2
36	Revision History.....	3
37	CONTENTS	4
38	1 Introduction	12
39	2 Application profile specification	12
40	2.1 Version information	12
41	2.2 Constraints naming convention	12
42	2.3 Profile constraints	13
43	2.4 Metadata.....	17
44	2.4.1 Constraints	17
45	2.4.2 Reference metadata	17
46	3 Detailed Profile Specification	17
47	3.1 General.....	17
48	3.2 (abstract,NC) BaseIrregularTimeSeries	23
49	3.3 (abstract,NC) BaseTimeSeries	23
50	3.4 (abstract) ConductingEquipment	24
51	3.5 (abstract) EnergyConnection	24
52	3.6 (abstract) EnergyConsumer root class	24
53	3.7 (abstract) Equipment.....	24
54	3.8 (abstract) GeneratingUnit root class	25
55	3.9 (NC) PowerShiftKeySchedule.....	25
56	3.10 (abstract) HydroPump root class	26
57	3.11 (abstract) IdentifiedObject root class	26
58	3.12 (abstract) PowerElectronicsUnit root class	26
59	3.13 (abstract) PowerSystemResource	26
60	3.14 (abstract,NC) ScheduleResource root class	27
61	3.15 (NC) BaseTimeSeriesKind enumeration	27
62	3.16 (NC) TimeSeriesInterpolationKind enumeration	27
63	3.17 UnitMultiplier enumeration	27
64	3.18 UnitSymbol enumeration	28
65	3.19 Seconds datatype	29
66	3.20 DateTime primitive	29
67	3.21 Float primitive	29
68	3.22 String primitive	29
69	3.23 (NC) ParticipationFactorTimePoint root class	29
70	3.24 (NC) ContingencySchedule	30
71	3.25 (NC) ContingencyTimePoint root class	30
72	3.26 (abstract) Contingency root class	31
73	3.27 (NC) AssessedElementSchedule	31
74	3.28 (NC) AssessedElementTimePoint root class	31
75	3.29 (abstract,NC) AssessedElement.....	33
76	3.30 (NC) GenericValueSchedule	33
77	3.31 (abstract,NC) RangeConstraint	34

78	3.32	(NC) GenericValueTimePoint root class	34
79	3.33	(NC) GenericAvailableSchedule	34
80	3.34	(NC) GenericEnablingSchedule	35
81	3.35	(abstract,NC) RemedialAction	36
82	3.36	(NC) EnablingTimePoint root class	36
83	3.37	(NC) AvailabilityTimePoint root class	36
84	3.38	(NC) PowerBidDependency root class	37
85	3.39	(NC) PowerBidSchedule	39
86	3.40	(NC) PowerBidScheduleTimePoint root class	40
87	3.41	(abstract,NC) AssessedElementWithRemedialAction root class	41
88	3.42	(abstract,NC) AssessedElementWithContingency root class	41
89	3.43	Duration primitive	41
90	3.44	Boolean primitive	41
91	3.45	RealEnergy datatype	41
92	3.46	(NC) BidDirectionKind enumeration	41
93	3.47	Currency enumeration	42
94	3.48	ActivePower datatype	46
95	3.49	PerCent datatype	46
96	3.50	Decimal primitive	46
97	3.51	Integer primitive	46
98	3.52	(abstract,NC) RemedialActionDependency root class	46
99	3.53	(NC) PowerBidDependencyKind enumeration	47
100	3.54	(abstract,NC) ContingencyWithRemedialAction root class	47
101	3.55	(NC) GridStateIntensitySchedule	47
102	3.56	(NC) ValueOffsetKind enumeration	48
103	3.57	(NC) PowerRemedialActionSchedule	48
104	3.58	(NC) PowerRemedialActionTimePoint root class	48
105	3.59	(abstract,NC) DCPole root class	49
106	3.60	(NC) PowerShiftKeyDistribution	49
107	3.61	(abstract,NC) PowerShiftKeyStrategy root class	49
108	3.62	(abstract) EquivalentInjection root class	50
109	3.63	(NC) UnitCostSchedule	50
110	3.64	(NC) UnitCostTimePoint root class	50
111	3.65	(NC) RemedialActionSchemeSchedule	51
112	3.66	(abstract,NC) RemedialActionScheme root class	51
113	3.67	(NC) RemedialActionSchemeTimePoint root class	51
114	3.68	(abstract,NC) PowerRemedialAction root class	52
115	3.69	(NC) RemedialActionGroupSchedule	52
116	3.70	(NC) RemedialActionGroupTimePoint root class	52
117	3.71	(abstract,NC) RemedialActionGroup root class	53
118	3.72	(NC) GridStateAlterationSchedule	53
119	3.73	(NC) GridStateAlterationTimePoint root class	53
120	3.74	(abstract,NC) GridStateAlteration root class	54
121	3.75	(abstract,NC) BaseRegularIntervalSchedule	54
122	3.76	Season	55
123	3.77	(NC) HourPattern	55

124	3.78	(NC) HourPeriod root class	55
125	3.79	(NC) AssessedElementRegularSchedule	56
126	3.80	(NC) AssessedElementRegularTimePoint root class	57
127	3.81	(NC) DayOfWeekKind enumeration	58
128	3.82	(NC) PeakKind enumeration	58
129	3.83	(NC) EnergyScenarioKind enumeration	58
130	3.84	(abstract,NC) EnergyGroup root class	59
131	3.85	(abstract,NC) EnergyBlockOrder root class	59
132	3.86	(abstract,NC) AutomationFunction root class	59
133	3.87	(abstract,NC) FunctionBlock root class	59
134	3.88	(NC) VoltageAngleSchedule	59
135	3.89	(NC) VoltageAngleTimePoint root class	60
136	3.90	(abstract,NC) VoltageAngleLimit root class	60
137	3.91	(NC) InfeedLimitSchedule	60
138	3.92	(abstract,NC) InfeedLimit root class	61
139	3.93	(NC) InfeedLimitTimePoint root class	61
140	3.94	(abstract,NC) PowerTransferCorridor root class	61
141	3.95	Time primitive	61
142	3.96	CurrentFlow datatype	61
143	3.97	MonthDay primitive	61
144	3.98	AngleDegrees datatype	61
145	3.99	(abstract,NC) CrossBorderRelevance root class	62
146	3.100	(NC) SecuredExclusionReasonKind enumeration	62
147	3.101	(abstract) EnergySource root class	62
148	3.102	(abstract) ExternalNetworkInjection root class	62
149	3.103	(NC) StageTriggerSchedule	62
150	3.104	(NC) StageTriggerTimePoint root class	63
151	3.105	(abstract,NC) StageTrigger root class	63
152	3.106	(NC) GenericSequenceSchedule	63
153	3.107	(NC) SequenceTimePoint root class	64
154	3.108	(NC) ReferenceValueKind enumeration	64
155	3.109	(NC) BaseCaseCurrentLimitSchedule	64
156	3.110	(abstract,NC) BaseCaseCurrentLimit root class	65
157	3.111	(NC) BaseCaseCurrentLimitTimePoint root class	65
158	3.112	(NC) FuelStorageSchedule	65
159	3.113	(NC) FuelStorageTimePoint root class	66
160	3.114	(NC) ReservoirSchedule	66
161	3.115	(NC) ReservoirTimePoint root class	67
162	3.116	(abstract,NC) FuelStorage root class	67
163	3.117	(abstract) Reservoir root class	67
164	Annex A (informative): Sample data		68
165	A.1	General	68
166	A.2	Sample instance data	68
167			
168	List of figures		

169	Figure 1 – Class diagram StateInstructionScheduleProfile::PowerShiftKeySchedule	18
170	Figure 2 – Class diagram StateInstructionScheduleProfile::Core	19
171	Figure 3 – Class diagram StateInstructionScheduleProfile::SecuritySchedule_One	20
172	Figure 4 – Class diagram StateInstructionScheduleProfile::PowerBidSchedule	20
173	Figure 5 – Class diagram StateInstructionScheduleProfile::RegularSecuritySchedule	21
174	Figure 6 – Class diagram StateInstructionScheduleProfile::SecuritySchedule_Two	22
175	Figure 7 – Class diagram StateInstructionScheduleProfile::GenericEnablingSchedule	22
176	Figure 8 – Class diagram StateInstructionScheduleProfile::GenericSequenceSchedule	23
177		
178	List of tables	
179	Table 1 – Attributes of StateInstructionScheduleProfile::BaseIrregularTimeSeries	23
180	Table 2 – Attributes of StateInstructionScheduleProfile::BaseTimeSeries	24
181	Table 3 – Attributes of StateInstructionScheduleProfile::ConductingEquipment	24
182	Table 4 – Attributes of StateInstructionScheduleProfile::EnergyConnection	24
183	Table 5 – Attributes of StateInstructionScheduleProfile::Equipment	25
184	Table 6 – Attributes of StateInstructionScheduleProfile::PowerShiftKeySchedule	25
185	Table 7 – Association ends of	
186	StateInstructionScheduleProfile::PowerShiftKeySchedule with other classes	25
187	Table 8 – Attributes of StateInstructionScheduleProfile::IdentifiedObject	26
188	Table 9 – Attributes of StateInstructionScheduleProfile::PowerSystemResource	26
189	Table 10 – Literals of StateInstructionScheduleProfile::BaseTimeSeriesKind	27
190	Table 11 – Literals of StateInstructionScheduleProfile::TimeSeriesInterpolationKind	27
191	Table 12 – Literals of StateInstructionScheduleProfile::UnitMultiplier	28
192	Table 13 – Literals of StateInstructionScheduleProfile::UnitSymbol	28
193	Table 14 – Attributes of StateInstructionScheduleProfile::Seconds	29
194	Table 15 – Attributes of StateInstructionScheduleProfile::ParticipationFactorTimePoint	29
195	Table 16 – Association ends of	
196	StateInstructionScheduleProfile::ParticipationFactorTimePoint with other classes	30
197	Table 17 – Attributes of StateInstructionScheduleProfile::ContingencySchedule	30
198	Table 18 – Association ends of StateInstructionScheduleProfile::ContingencySchedule	
199	with other classes	30
200	Table 19 – Attributes of StateInstructionScheduleProfile::ContingencyTimePoint	30
201	Table 20 – Association ends of StateInstructionScheduleProfile::ContingencyTimePoint	
202	with other classes	31
203	Table 21 – Attributes of StateInstructionScheduleProfile::AssessedElementSchedule	31
204	Table 22 – Association ends of	
205	StateInstructionScheduleProfile::AssessedElementSchedule with other classes	31
206	Table 23 – Attributes of StateInstructionScheduleProfile::AssessedElementTimePoint	31
207	Table 24 – Association ends of	
208	StateInstructionScheduleProfile::AssessedElementTimePoint with other classes	33
209	Table 25 – Attributes of StateInstructionScheduleProfile::AssessedElement	33

210	Table 26 – Attributes of StateInstructionScheduleProfile::GenericValueSchedule	33
211	Table 27 – Association ends of	
212	StateInstructionScheduleProfile::GenericValueSchedule with other classes	34
213	Table 28 – Attributes of StateInstructionScheduleProfile::RangeConstraint	34
214	Table 29 – Attributes of StateInstructionScheduleProfile::GenericValueTimePoint	34
215	Table 30 – Association ends of	
216	StateInstructionScheduleProfile::GenericValueTimePoint with other classes	34
217	Table 31 – Attributes of StateInstructionScheduleProfile::GenericAvailableSchedule	35
218	Table 32 – Association ends of	
219	StateInstructionScheduleProfile::GenericAvailableSchedule with other classes	35
220	Table 33 – Attributes of StateInstructionScheduleProfile::GenericEnablingSchedule	35
221	Table 34 – Association ends of	
222	StateInstructionScheduleProfile::GenericEnablingSchedule with other classes	35
223	Table 35 – Attributes of StateInstructionScheduleProfile::RemedialAction	36
224	Table 36 – Attributes of StateInstructionScheduleProfile::EnablingTimePoint	36
225	Table 37 – Association ends of StateInstructionScheduleProfile::EnablingTimePoint	
226	with other classes	36
227	Table 38 – Attributes of StateInstructionScheduleProfile::AvailabilityTimePoint	37
228	Table 39 – Association ends of StateInstructionScheduleProfile::AvailabilityTimePoint	
229	with other classes	37
230	Table 40 – Attributes of StateInstructionScheduleProfile::PowerBidDependency	37
231	Table 41 – Association ends of StateInstructionScheduleProfile::PowerBidDependency	
232	with other classes	39
233	Table 42 – Attributes of StateInstructionScheduleProfile::PowerBidSchedule	39
234	Table 43 – Association ends of StateInstructionScheduleProfile::PowerBidSchedule	
235	with other classes	40
236	Table 44 – Attributes of StateInstructionScheduleProfile::PowerBidScheduleTimePoint	40
237	Table 45 – Association ends of	
238	StateInstructionScheduleProfile::PowerBidScheduleTimePoint with other classes	41
239	Table 46 – Attributes of StateInstructionScheduleProfile::RealEnergy	41
240	Table 47 – Literals of StateInstructionScheduleProfile::BidDirectionKind	41
241	Table 48 – Literals of StateInstructionScheduleProfile::Currency	42
242	Table 49 – Attributes of StateInstructionScheduleProfile::ActivePower	46
243	Table 50 – Attributes of StateInstructionScheduleProfile::PerCent	46
244	Table 51 – Literals of StateInstructionScheduleProfile::PowerBidDependencyKind	47
245	Table 52 – Attributes of StateInstructionScheduleProfile::GridStateIntensitySchedule	47
246	Table 53 – Association ends of	
247	StateInstructionScheduleProfile::GridStateIntensitySchedule with other classes	47
248	Table 54 – Literals of StateInstructionScheduleProfile::ValueOffsetKind	48
249	Table 55 – Attributes of	
250	StateInstructionScheduleProfile::PowerRemedialActionSchedule	48
251	Table 56 – Association ends of	
252	StateInstructionScheduleProfile::PowerRemedialActionSchedule with other classes	48

253	Table 57 – Attributes of	
254	StateInstructionScheduleProfile::PowerRemedialActionTimePoint	49
255	Table 58 – Association ends of	
256	StateInstructionScheduleProfile::PowerRemedialActionTimePoint with other classes	49
257	Table 59 – Attributes of StateInstructionScheduleProfile::PowerShiftKeyDistribution	49
258	Table 60 – Association ends of	
259	StateInstructionScheduleProfile::PowerShiftKeyDistribution with other classes	49
260	Table 61 – Attributes of StateInstructionScheduleProfile::UnitCostSchedule	50
261	Table 62 – Association ends of StateInstructionScheduleProfile::UnitCostSchedule with	
262	other classes	50
263	Table 63 – Attributes of StateInstructionScheduleProfile::UnitCostTimePoint	50
264	Table 64 – Association ends of StateInstructionScheduleProfile::UnitCostTimePoint	
265	with other classes	51
266	Table 65 – Attributes of	
267	StateInstructionScheduleProfile::RemedialActionSchemeSchedule	51
268	Table 66 – Association ends of	
269	StateInstructionScheduleProfile::RemedialActionSchemeSchedule with other classes	51
270	Table 67 – Attributes of	
271	StateInstructionScheduleProfile::RemedialActionSchemeTimePoint	51
272	Table 68 – Association ends of	
273	StateInstructionScheduleProfile::RemedialActionSchemeTimePoint with other classes	52
274	Table 69 – Attributes of	
275	StateInstructionScheduleProfile::RemedialActionGroupSchedule	52
276	Table 70 – Association ends of	
277	StateInstructionScheduleProfile::RemedialActionGroupSchedule with other classes	52
278	Table 71 – Attributes of	
279	StateInstructionScheduleProfile::RemedialActionGroupTimePoint	53
280	Table 72 – Association ends of	
281	StateInstructionScheduleProfile::RemedialActionGroupTimePoint with other classes	53
282	Table 73 – Attributes of StateInstructionScheduleProfile::GridStateAlterationSchedule	53
283	Table 74 – Association ends of	
284	StateInstructionScheduleProfile::GridStateAlterationSchedule with other classes	53
285	Table 75 – Attributes of StateInstructionScheduleProfile::GridStateAlterationTimePoint	54
286	Table 76 – Association ends of	
287	StateInstructionScheduleProfile::GridStateAlterationTimePoint with other classes	54
288	Table 77 – Attributes of StateInstructionScheduleProfile::BaseRegularIntervalSchedule	54
289	Table 78 – Association ends of	
290	StateInstructionScheduleProfile::BaseRegularIntervalSchedule with other classes	55
291	Table 79 – Attributes of StateInstructionScheduleProfile::Season	55
292	Table 80 – Attributes of StateInstructionScheduleProfile::HourPattern	55
293	Table 81 – Attributes of StateInstructionScheduleProfile::HourPeriod	56
294	Table 82 – Association ends of StateInstructionScheduleProfile::HourPeriod with other	
295	classes	56
296	Table 83 – Attributes of	
297	StateInstructionScheduleProfile::AssessedElementRegularSchedule	56

298	Table 84 – Association ends of	
299	StateInstructionScheduleProfile::AssessedElementRegularSchedule with other classes	57
300	Table 85 – Attributes of	
301	StateInstructionScheduleProfile::AssessedElementRegularTimePoint	57
302	Table 86 – Association ends of	
303	StateInstructionScheduleProfile::AssessedElementRegularTimePoint with other	
304	classes	58
305	Table 87 – Literals of StateInstructionScheduleProfile::DayOfWeekKind	58
306	Table 88 – Literals of StateInstructionScheduleProfile::PeakKind	58
307	Table 89 – Literals of StateInstructionScheduleProfile::EnergyScenarioKind	59
308	Table 90 – Attributes of StateInstructionScheduleProfile::VoltageAngleSchedule	59
309	Table 91 – Association ends of StateInstructionScheduleProfile::VoltageAngleSchedule	
310	with other classes	59
311	Table 92 – Attributes of StateInstructionScheduleProfile::VoltageAngleTimePoint	60
312	Table 93 – Association ends of	
313	StateInstructionScheduleProfile::VoltageAngleTimePoint with other classes	60
314	Table 94 – Attributes of StateInstructionScheduleProfile::InfeedLimitSchedule	60
315	Table 95 – Association ends of StateInstructionScheduleProfile::InfeedLimitSchedule	
316	with other classes	60
317	Table 96 – Attributes of StateInstructionScheduleProfile::InfeedLimitTimePoint	61
318	Table 97 – Association ends of StateInstructionScheduleProfile::InfeedLimitTimePoint	
319	with other classes	61
320	Table 98 – Attributes of StateInstructionScheduleProfile::CurrentFlow	61
321	Table 99 – Attributes of StateInstructionScheduleProfile::AngleDegrees	62
322	Table 100 – Literals of StateInstructionScheduleProfile::SecuredExclusionReasonKind	62
323	Table 101 – Attributes of StateInstructionScheduleProfile::StageTriggerSchedule	62
324	Table 102 – Association ends of	
325	StateInstructionScheduleProfile::StageTriggerSchedule with other classes	63
326	Table 103 – Attributes of StateInstructionScheduleProfile::StageTriggerTimePoint	63
327	Table 104 – Association ends of	
328	StateInstructionScheduleProfile::StageTriggerTimePoint with other classes	63
329	Table 105 – Attributes of StateInstructionScheduleProfile::GenericSequenceSchedule	63
330	Table 106 – Association ends of	
331	StateInstructionScheduleProfile::GenericSequenceSchedule with other classes	64
332	Table 107 – Attributes of StateInstructionScheduleProfile::SequenceTimePoint	64
333	Table 108 – Association ends of StateInstructionScheduleProfile::SequenceTimePoint	
334	with other classes	64
335	Table 109 – Literals of StateInstructionScheduleProfile::ReferenceValueKind	64
336	Table 110 – Attributes of	
337	StateInstructionScheduleProfile::BaseCaseCurrentLimitSchedule	64
338	Table 111 – Association ends of	
339	StateInstructionScheduleProfile::BaseCaseCurrentLimitSchedule with other classes	65
340	Table 112 – Attributes of	
341	StateInstructionScheduleProfile::BaseCaseCurrentLimitTimePoint	65

342	Table 113 – Association ends of	
343	StateInstructionScheduleProfile::BaseCaseCurrentLimitTimePoint with other classes	65
344	Table 114 – Attributes of StateInstructionScheduleProfile::FuelStorageSchedule	65
345	Table 115 – Association ends of	
346	StateInstructionScheduleProfile::FuelStorageSchedule with other classes	66
347	Table 116 – Attributes of StateInstructionScheduleProfile::FuelStorageTimePoint	66
348	Table 117 – Association ends of	
349	StateInstructionScheduleProfile::FuelStorageTimePoint with other classes	66
350	Table 118 – Attributes of StateInstructionScheduleProfile::ReservoirSchedule	66
351	Table 119 – Association ends of StateInstructionScheduleProfile::ReservoirSchedule	
352	with other classes	67
353	Table 120 – Attributes of StateInstructionScheduleProfile::ReservoirTimePoint	67
354	Table 121 – Association ends of StateInstructionScheduleProfile::ReservoirTimePoint	
355	with other classes	67
356		

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_NC24v18_MS10v03_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/2.4>
- Version info: 2.4.0
- Prior version: <https://ap-voc.cim4.eu/StateInstructionSchedule/2.3>
- 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:///CIM17-2_CGMES31v01_PROF-20v02_NC24v18_MS10v03_DES10v01.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

393 2.3 Profile constraints

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

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

- 399 • C:452:ALL:NA:datatypes

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

- 402 • R:452:ALL:NA:exchange

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

- 405 • R:452:ALL:NA:exchange1

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

- 411 • R:452:ALL:NA:exchange2

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

- 420 • R:452:ALL:NA:exchange3

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

- 424 • R:452:ALL:NA:exchange4

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

- 436 • R:452:ALL:NA:cardinality

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

- 442 • R:452:ALL:NA:associations

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

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

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

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

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

- 457 • R:452:ALL:NA:uniqueIdentifier

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

- 460 • R:452:ALL:NA:unitMultiplier

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

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

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

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

466 The string IdentifiedObject.description is maximum 256 characters.

- 467 • C:452:ALL:NA:float

468 An attribute that is defined as float (e.g. has a type Float or a type which is a Datatype
469 with .value attribute of type Float) shall support ISO/IEC 60559:2020 for floating-point
470 arithmetic using single precision floating point. A single precision float supports 7
471 significant digits where the significant digits are described as an integer, or a decimal
472 number with 6 decimal digits. Two float values are equal when the significant with 7
473 digits are identical, e.g. 1234567 is equal 1.234567E6 and so are 1.2345678 and
474 1.234567E0.

- 475 • R:NC:ALL:NA:serialization

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

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

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

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

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

499 It should be taken into account that this approach deviates from MVAL5 (IEC 61970-
 500 600-1:2021), which creates multiple inheritance at serialization. For instance, with this
 501 more explicit exchange the serialization of the association between abstract class
 502 Equipment and abstract class Circuit for a PowerTransformer will be serialized as
 503 follows:

504 ○ for association
 505 <cim:PowerTransformer rdf:about="_c328f787-bc17-47ad-a59f-6ba7133340d0">
 506 <nc:Equipment.Circuit rdf:resource="#_9ced16ac-d076-4ef9-a241-a998a579e77b"/>
 507 </cim:PowerTransformer>
 508 ○ for attribute
 509 <cim:ACLineSegment rdf:about="_04f681aa-6999-4fb3-9775-acaa5eb7ceff">
 510 <cim:Equipment.inService>true</cim:Equipment.inService>
 511 </cim:ACLineSegment>

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

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

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

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

- 520 PowerShiftKeySchedule can be used as a part of a bid pattern. In that case, the
521 schedules are following the bid time.
- 522 • C:NC:SIS:PowerShiftKeySchedule.ParticipationFactorTimePoint:dependency
- 523 The multiplicity of the association end
524 PowerShiftKeySchedule.ParticipationFactorTimePoint shall be 1..n for all cases except
525 if the PowerShiftKeySchedule is associated with PowerShiftKeyDistribution.
- 526 • C:NC:SIS:PowerShiftKeySchedule:associations
527 PowerShiftKeySchedule shall include one and only one association to
528 ScheduleResource, PowerElectronicUnit, GeneratingUnit, DCPole, EnergyGroup,
529 EquivalentInjection, EnergyConsumer, EnergySource, ExternalNetworkInjection and
530 HydroPump.
- 531 • C:NC:SIS:PowerShiftKeyDistribution:associations
532 PowerShiftKeyDistribution shall include one and only one association to either
533 PowerBidSchedule or PowerShiftKeyStrategy.
- 534 • C:NC:SIS:GenericSequenceSchedule:associations
535 Association end GenericSequenceSchedule.EnergyBlockOrder is required.
- 536 • C:NC:SIS:PowerBidSchedule:associations
537 PowerBidSchedule shall include one and only one association to either
538 ScheduleResource or PowerRemedialAction.
- 539 • C:NC:SIS:PowerBidScheduleTimePoint:attributes
540 PowerBidScheduleTimePoint can have the following combinations of optional attributes:
541 1) activationCost, 2) shutdownCost or 3) any of the following p, minimumActivationP,
542 price, reservePrice, stepIncrementP.
- 543 • C:NC:SIS:PowerBidDependency.finishToFinishLagKind:required
544 If PowerBidDependency.finishToFinishLag is present
545 PowerBidDependency.finishToFinishLagKind is required.
- 546 • C:NC:SIS:PowerBidDependency.finishToStartLagKind:required
547 If PowerBidDependency.finishToStartLag is present
548 PowerBidDependency.finishToStartLagKind is required.
- 549 • C:NC:SIS:PowerBidDependency.overlapKind:required
550 If PowerBidDependency.overlap is present PowerBidDependency.overlapKind is
551 required.
- 552 • C:NC:SIS:PowerBidDependency.startToStartLagKind:required
553 If PowerBidDependency.startToStartLag is present
554 PowerBidDependency.startToStartLagKind is required.
- 555 • C:NC:SIS:PowerBidDependency.kind:exclusive
556 If PowerBidDependency.kind is set to PowerBidDependencyKind.exclusive, the
557 following attributes shall not be defined: PowerBidDependency.overlap,
558 PowerBidDependency.startToStartLag, PowerBidDependency.finishToStartLag,
559 PowerBidDependency.finishToFinishLag.
- 560
- 561
- 562

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.

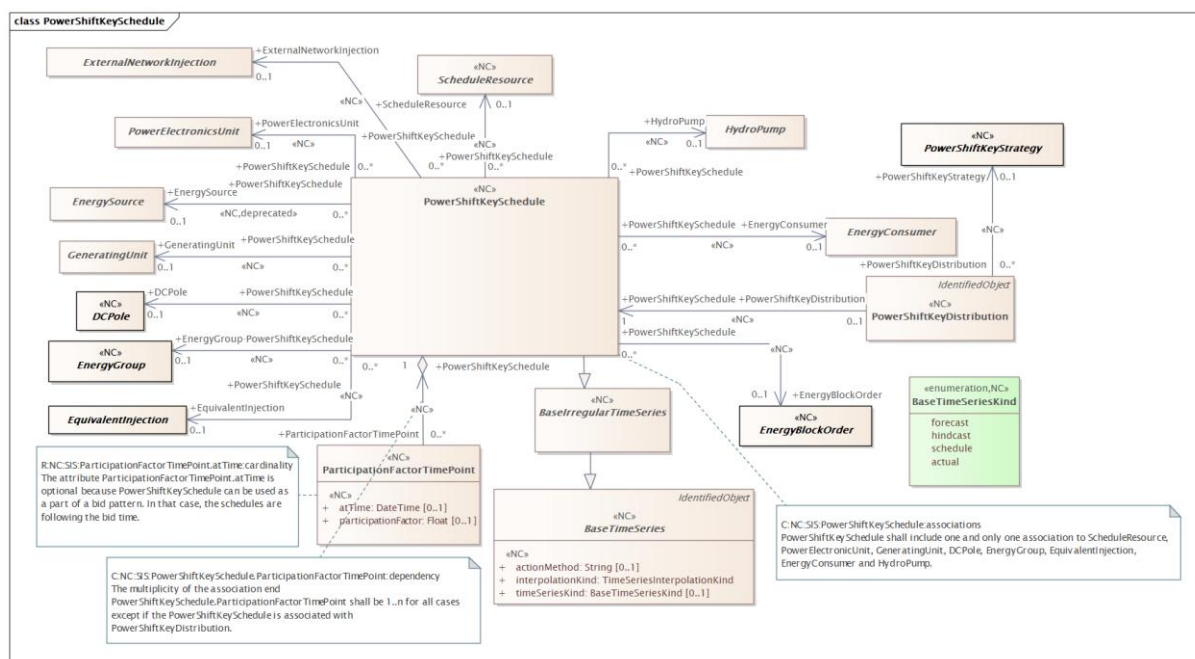


Figure 1 – Class diagram StateInstructionScheduleProfile::PowerShiftKeySchedule

Figure 1: The diagram shows power shift keys related classes.

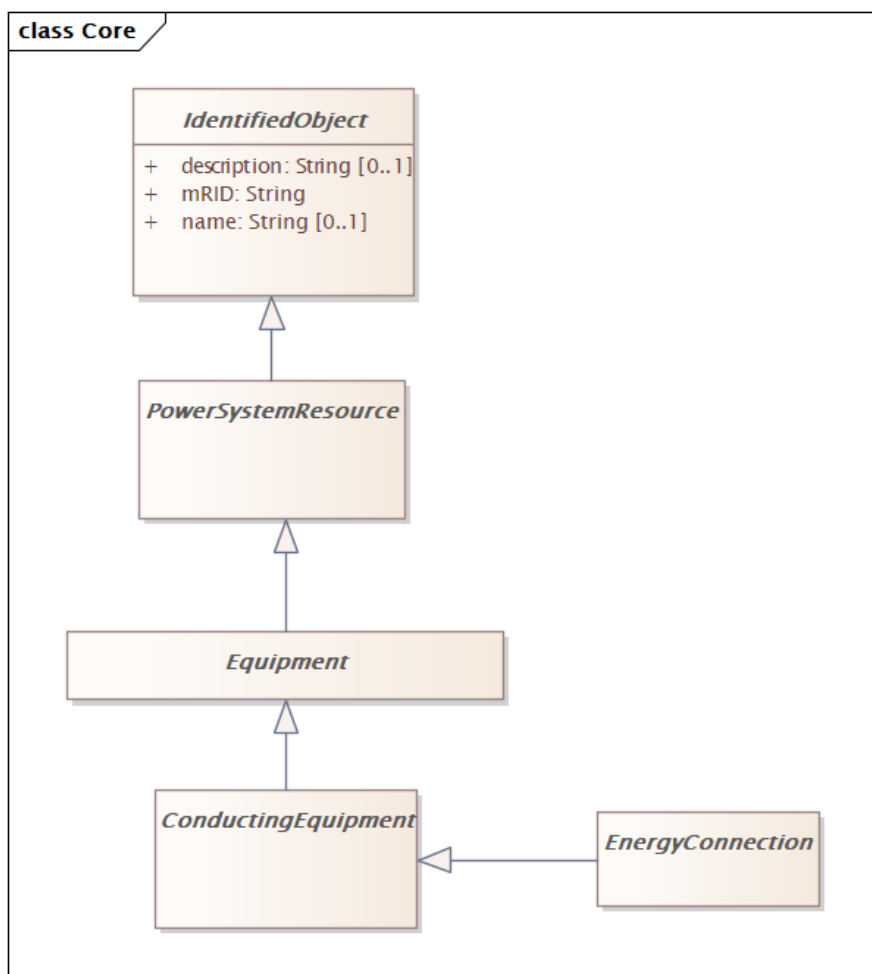


Figure 2 – Class diagram StateInstructionScheduleProfile::Core

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

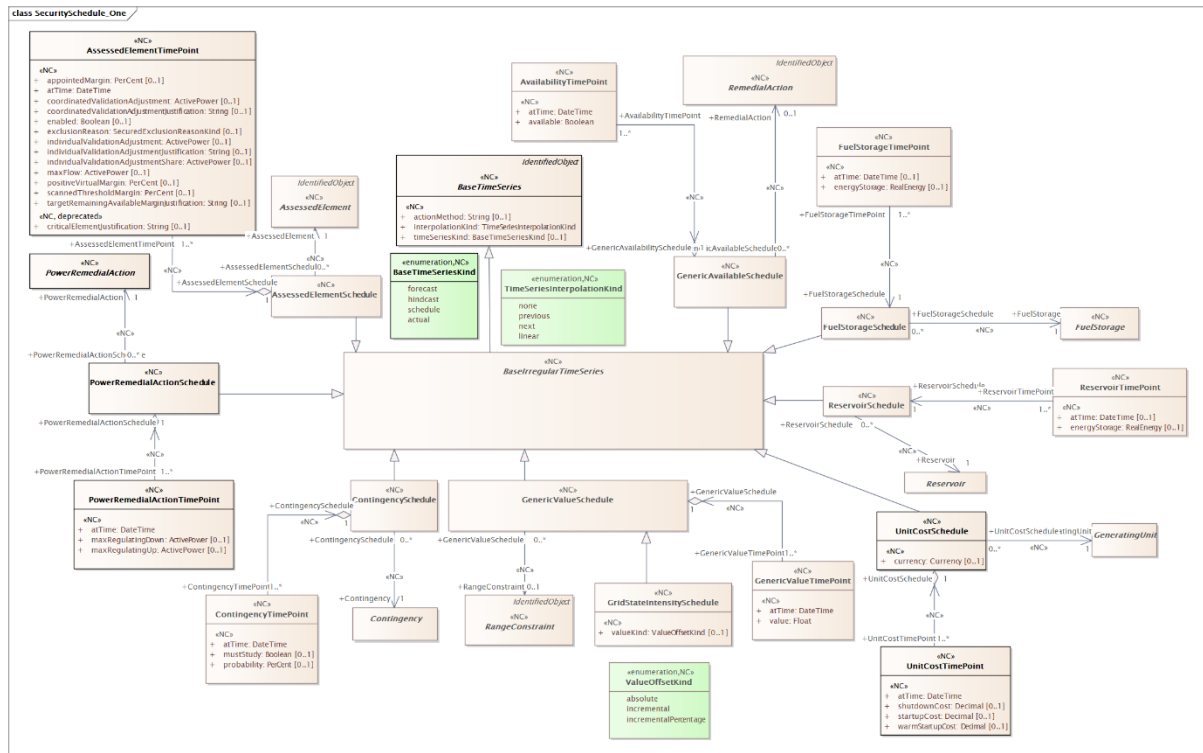


Figure 3 – Class diagram StateInstructionScheduleProfile::SecuritySchedule_One

Figure 3: The diagram shows security schedule related classes.

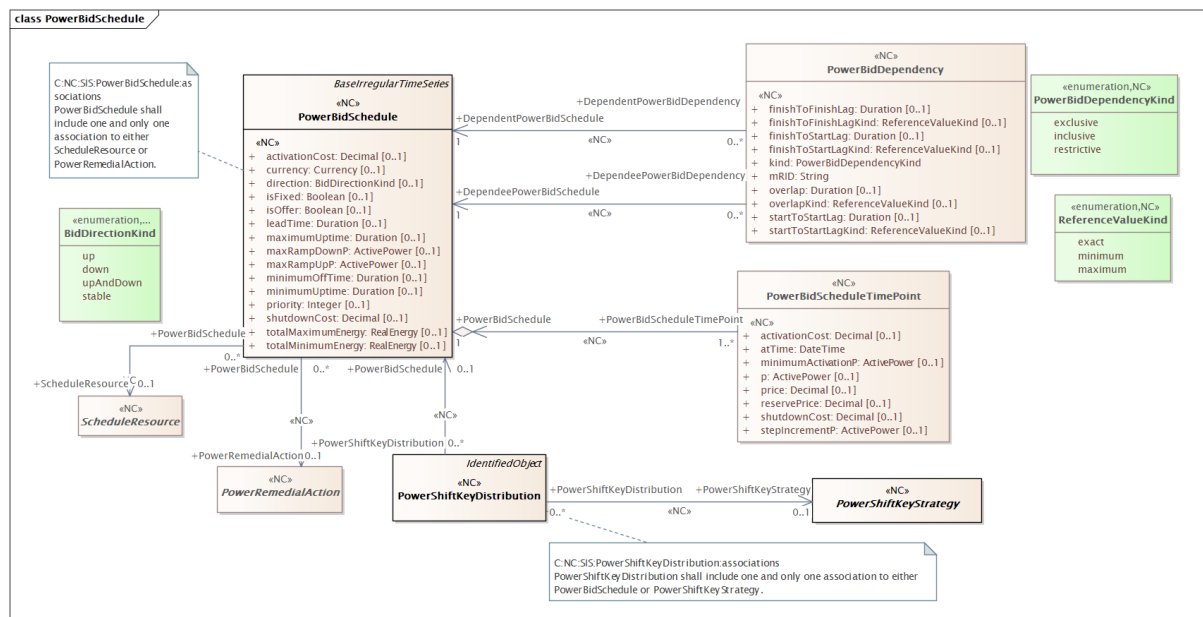


Figure 4 – Class diagram StateInstructionScheduleProfile::PowerBidSchedule

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.

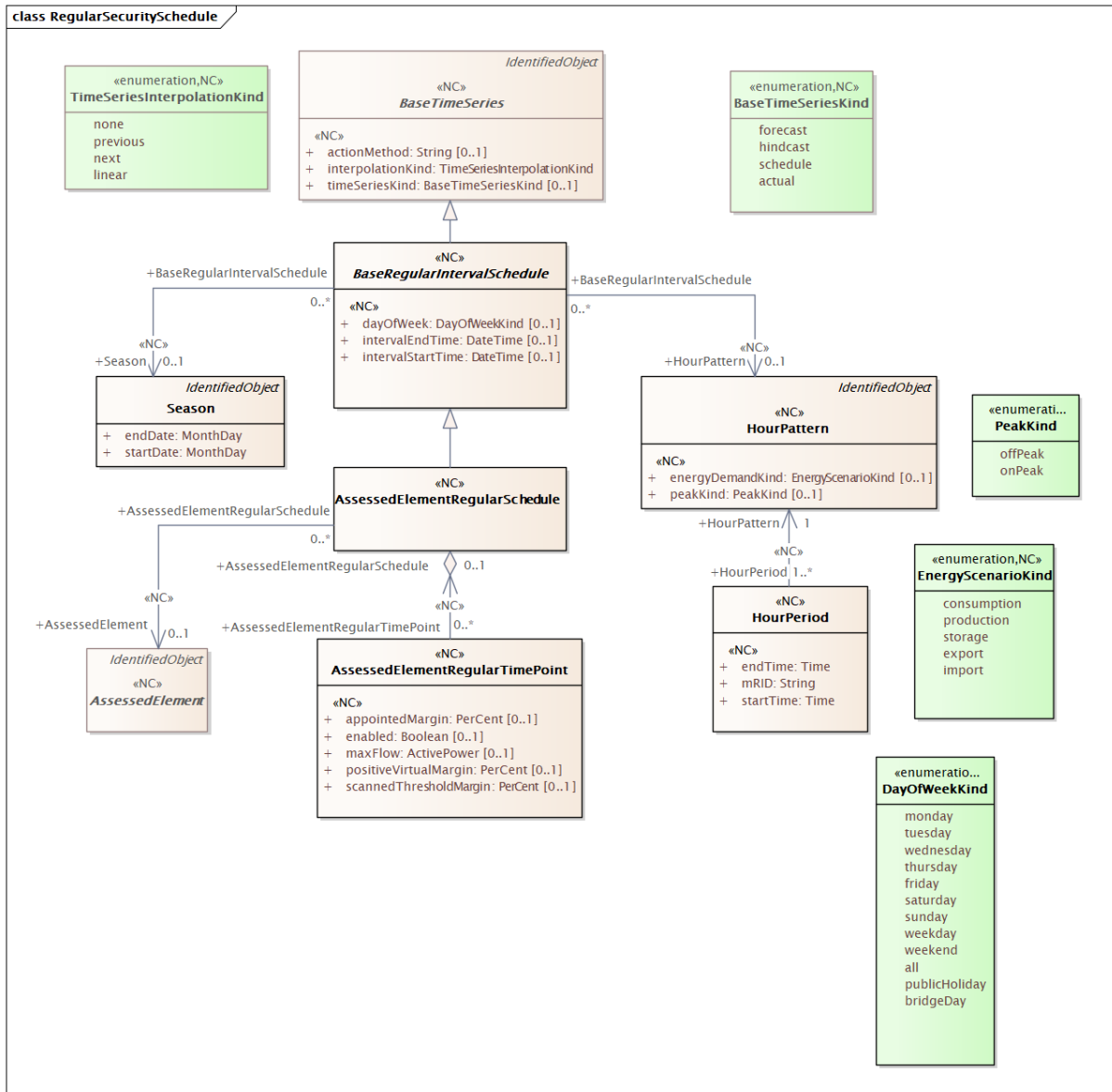


Figure 5 – Class diagram StateInstructionScheduleProfile::RegularSecuritySchedule

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

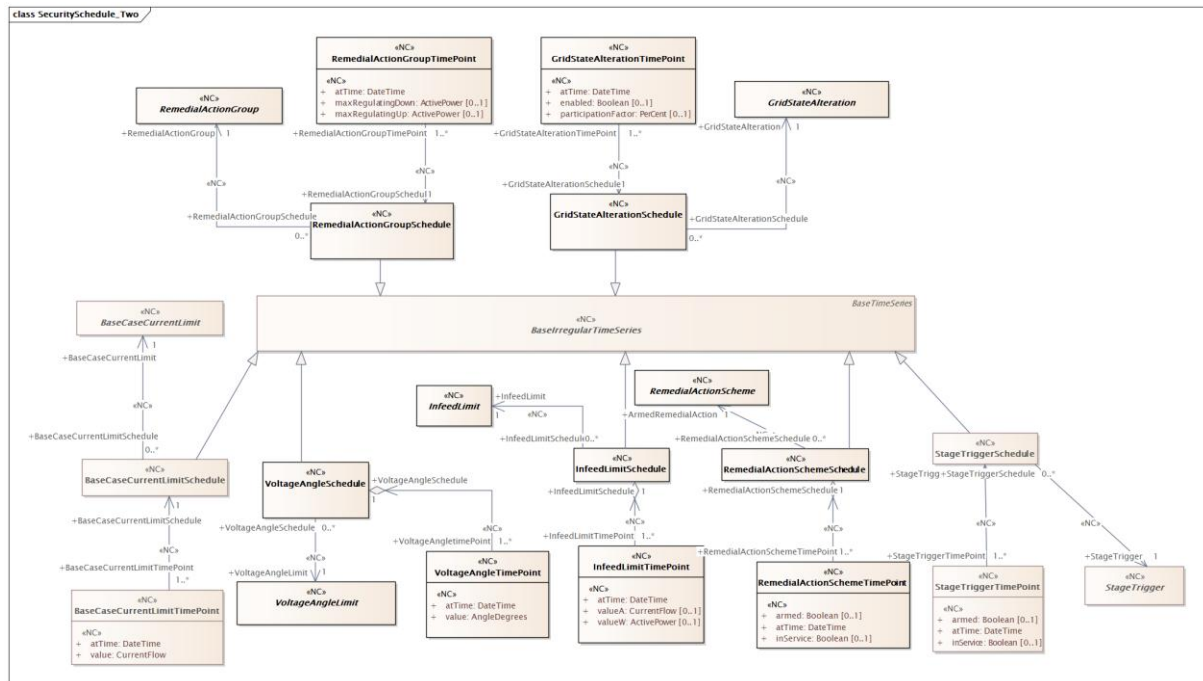


Figure 6 – Class diagram StateInstructionScheduleProfile::SecuritySchedule_Two

Figure 6: The diagram shows security schedule related classes.

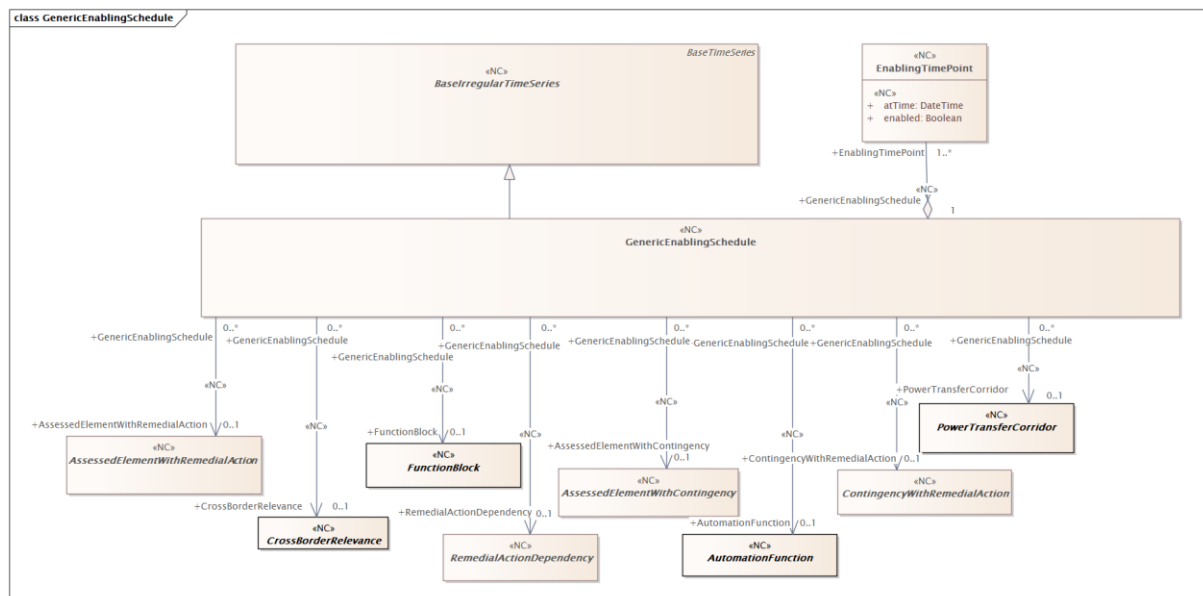


Figure 7 – Class diagram StateInstructionScheduleProfile::GenericEnablingSchedule

Figure 7: The diagram shows generic enabling schedule related classes.

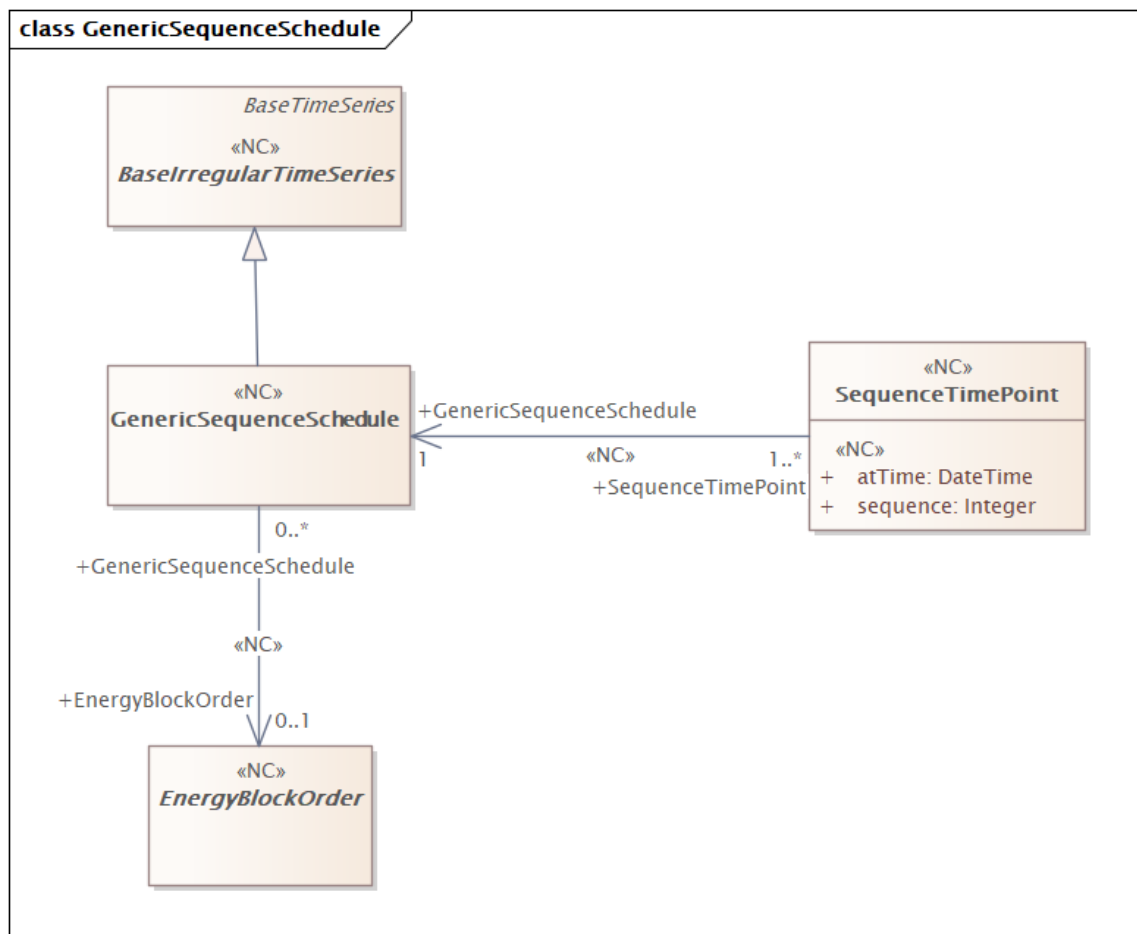


Figure 8 – Class diagram StateInstructionScheduleProfile::GenericSequenceSchedule

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.

630 Table 2 shows all attributes of BaseTimeSeries.

631 **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

632

633 3.4 (abstract) ConductingEquipment

634 Inheritance path = [Equipment](#) : [PowerSystemResource](#) : [IdentifiedObject](#)

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

637 Table 3 shows all attributes of ConductingEquipment.

638 **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

639

640 3.5 (abstract) EnergyConnection

641 Inheritance path = [ConductingEquipment](#) : [Equipment](#) : [PowerSystemResource](#) :
642 [IdentifiedObject](#)

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

644 Table 4 shows all attributes of EnergyConnection.

645 **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

646

647 3.6 (abstract) EnergyConsumer root class

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

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

652 3.7 (abstract) Equipment

653 Inheritance path = [PowerSystemResource](#) : [IdentifiedObject](#)

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

655 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 power 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 PowerElectronicsUnit.

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. 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.

3.15 (NC) BaseTimeSeriesKind enumeration

Kind of time series.

Table 10 shows all literals of BaseTimeSeriesKind.

Table 10 – 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 11 shows all literals of TimeSeriesInterpolationKind.

Table 11 – 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.
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 "P" represents the derived unit "m2Pers", then applying the multiplier "k" can be conceptualized simply as "kP".

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 "P", then it is easier to conceptualize the multiplier "m" as creating the proper unit "mP", and not the forbidden unit "mkg".

Table 12 shows all literals of UnitMultiplier.

Table 12 – 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 13 shows all literals of UnitSymbol.

Table 13 – 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 $V\cos(\phi)$), is expressed in Watts. See also apparent power and reactive power.

literal	value	description
Wh	72	Real energy in watt hours.

3.19 Seconds datatype

Time, in seconds.

Table 14 shows all attributes of Seconds.

Table 14 – 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)

3.20 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.

3.21 Float primitive

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

3.22 String primitive

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

3.23 (NC) ParticipationFactorTimePoint root class

Participation factor for a given point in time.

Table 15 shows all attributes of ParticipationFactorTimePoint.

Table 15 – 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. $V=T/\text{sum}(PF)$. In the case of priority strategy, the item with the lowest number gets allocated energy first.

Table 16 shows all association ends of ParticipationFactorTimePoint with other classes.

Table 16 – 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 17 shows all attributes of ContingencySchedule.

Table 17 – 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 18 shows all association ends of ContingencySchedule with other classes.

Table 18 – 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 19 shows all attributes of ContingencyTimePoint.

Table 19 – 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 20 shows all association ends of ContingencyTimePoint with other classes.

Table 20 – 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.

3.26 (abstract) Contingency root class

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

3.27 (NC) AssessedElementSchedule

Inheritance path = [BaseIrregularTimeSeries](#) : [BaseTimeSeries](#) : [IdentifiedObject](#)

Schedule for assessed element.

Table 21 shows all attributes of AssessedElementSchedule.

Table 21 – 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

Table 22 shows all association ends of AssessedElementSchedule with other classes.

Table 22 – 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.

3.28 (NC) AssessedElementTimePoint root class

Assessed element instruction value at a given point in time.

Table 23 shows all attributes of AssessedElementTimePoint.

Table 23 – 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. It is only used when an assessed element is considered conservative for a region. 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.
positiveVirtualMargin	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). 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.
criticalElementJustification	0..1	String	(deprecated,NC) Justification indicating the kind of critical element. 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 24 shows all association ends of AssessedElementTimePoint with other classes.

Table 24 – 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 25 shows all attributes of AssessedElement.

Table 25 – 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 26 shows all attributes of GenericValueSchedule.

Table 26 – 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 27 shows all association ends of GenericValueSchedule with other classes.

Table 27 – 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 range constraint.

Table 28 shows all attributes of RangeConstraint.

Table 28 – 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 29 shows all attributes of GenericValueTimePoint.

Table 29 – 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 30 shows all association ends of GenericValueTimePoint with other classes.

Table 30 – 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 31 shows all attributes of GenericAvailableSchedule.

Table 31 – 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 32 shows all association ends of GenericAvailableSchedule with other classes.

Table 32 – 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 33 shows all attributes of GenericEnablingSchedule.

Table 33 – 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 34 shows all association ends of GenericEnablingSchedule with other classes.

Table 34 – 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.

874

875 **3.35 (abstract,NC) RemedialAction**876 Inheritance path = [IdentifiedObject](#)

877 Remedial action describes one or more actions that can be performed on a given power system
 878 model situation to eliminate one or more identified breaches of constraints. The remedial action
 879 can be costly, and have a cost characteristic, or non costly.

880 Table 35 shows all attributes of RemedialAction.

881 **Table 35 – 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

882

883 **3.36 (NC) EnablingTimePoint root class**

884 Enabling instruction value at a given point in time.

885 Table 36 shows all attributes of EnablingTimePoint.

886 **Table 36 – 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.

887

888 Table 37 shows all association ends of EnablingTimePoint with other classes.

889 **Table 37 – Association ends of StateInstructionScheduleProfile::EnablingTimePoint**
 890 **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.

891

892 **3.37 (NC) AvailabilityTimePoint root class**

893 Availability instruction value at a given point in time.

894 Table 38 shows all attributes of AvailabilityTimePoint.

Table 38 – 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 39 shows all association ends of AvailabilityTimePoint with other classes.

Table 39 – 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 40 shows all attributes of PowerBidDependency.

Table 40 – Attributes of StateInstructionScheduleProfile::PowerBidDependency

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. 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.
kind	1..1	PowerBidDependencyKind	(NC) Type of dependency between bids. It indicates the activation dependency between dependee (parent) and dependent (child) bids. It does not indicate any running condition between dependee and dependent bids. If PowerBidDependencyKind equals to restrictive, a combination of the following attributes PowerBidDependency.startToStartLag, PowerBidDependency.finishToStartLag, PowerBidDependency.finishToFinishLag, or PowerBidDependency.overlap shall be used.
finishToFinishLag	0..1	Duration	(NC) Time between the deactivation of the dependee (parent) bids and the deactivation of dependent (child) bids. It cannot be used when the PowerBidDependency.kind is set to exclusive. Depending on the finishToFinishLagKind the value has the following meaning: - exact: the indicated lag should be exactly respected. - minimum: the indicated lag should be the minimum respected one, and a greater or equal lag for the deactivation of the dependent (child) after the deactivation of the dependee (parent) is allowed. - maximum: the indicated lag is the maximum that can occur, and a less or equal lag for the

name	mult	type	description
			deactivation of the dependent (child) after the deactivation of the dependee (parent) is allowed.
finishToFinishLagKind	0..1	ReferenceValueKind	(NC) It indicates the finishToFinishLag kind between the deactivation of the dependee (parent) bids and the deactivation of dependent (child) bids. It is required if PowerBidDependency.finishToFinishLag is used.
finishToStartLag	0..1	Duration	(NC) Time between the deactivation of the dependee (parent) bids and the activation of dependent (child) bids. It cannot be used when the PowerBidDependency.kind is set to exclusive. Depending on the finishToStartLagKind the value has the following meaning: - exact: the indicated lag should be exactly respected. - minimum: the indicated lag should be the minimum respected one, and a greater or equal lag for the activation of the dependent (child) after the deactivation of the dependee (parent) is allowed. - maximum: the indicated lag is the maximum that can occur, and a less or equal lag for the activation of the dependent (child) after the deactivation of the dependee (parent) is allowed.
finishToStartLagKind	0..1	ReferenceValueKind	(NC) It indicates the finishToStartLag kind between the deactivation of the dependee (parent) bids and the activation of dependent (child) bids. It is required if PowerBidDependency.finishToStartLag is used.
overlap	0..1	Duration	(NC) It indicates the duration the dependee (parent) and dependent (child) bids are active (run) at the same time. It cannot be used when the PowerBidDependency.kind is set to exclusive. Depending on the overlapKind the value has the following meaning: - exact: the dependee (parent) and the dependent (child) should be active (running) for the exact (equal) specified duration. - minimum: the dependee (parent) and the dependent (child) should be active (running) at least (greater than or equal) for the specified duration. - maximum: the dependee (parent) and the dependent (child) should be active (running) at the most (less than or equal) for the specified duration.
overlapKind	0..1	ReferenceValueKind	(NC) It indicates the overlap kind for the duration the parent (dependee) and child (dependent) bids are active at the same time. It is required when PowerBidDependency.overlap is used.
startToStartLag	0..1	Duration	(NC) Time between the activation of dependee (parent) bids and the activation of dependent (child) bids. It cannot be used when the PowerBidDependency.kind is set to exclusive. Depending on the startToStartLagKind the value has the following meaning: - exact - the indicated lag should be exactly respected.

name	mult	type	description
			- minimum - the indicated lag should be the minimum respected one and a higher or equal lag for the activation of the dependent (child) after the activation of the dependee (parent) is allowed. - maximum - the indicated lag is the maximum that can occur, and a smaller or equal lag for the activation of the dependent (child) after the activation of the dependee (parent) is allowed.
startToStartLagKind	0..1	ReferenceValueKind	(NC) Indicates the startToStartLag kind between the activation of the dependee (parent) bid and the activation of dependent (child) bid. It is required if PowerBidDependency.startToStartLag is used.

Table 41 shows all association ends of PowerBidDependency with other classes.

Table 41 – Association ends of StateInstructionScheduleProfile::PowerBidDependency with other classes

mult from	name	mult to	type	description
0..*	DependentPowerBidSchedule	1..1	PowerBidSchedule	(NC) The dependent (child) power bid schedule for this dependency.
0..*	DependeePowerBidSchedule	1..1	PowerBidSchedule	(NC) The dependee (parent) power bid schedule for this dependency.

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.

Table 42 shows all attributes of PowerBidSchedule.

Table 42 – 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.

name	mult	type	description
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

Table 43 shows all association ends of PowerBidSchedule with other classes.

Table 43 – 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 44 shows all attributes of PowerBidScheduleTimePoint.

Table 44 – 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) Price given in the time point.
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.

name	mult	type	description
activationCost	0..1	Decimal	(NC) Cost to activate the bid in the given time point.
shutdownCost	0..1	Decimal	(NC) Total shutdown cost incurred for all the units involved in the bid given in the time point. This overrides any cost on the specific unit.

Table 45 shows all association ends of PowerBidScheduleTimePoint with other classes.

**Table 45 – 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 46 shows all attributes of RealEnergy.

Table 46 – 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 47 shows all literals of BidDirectionKind.

Table 47 – 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.

952

953 **3.47 Currency enumeration**

954 Monetary currencies. ISO 4217 standard including 3-character currency code.

955 Table 48 shows all literals of Currency.

956

Table 48 – 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.

3.48 ActivePower datatype

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

Table 49 shows all attributes of ActivePower.

Table 49 – 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 50 shows all attributes of PerCent.

Table 50 – 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)

3.50 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 activation dependency.

Table 51 shows all literals of PowerBidDependencyKind.

Table 51 – Literals of StateInstructionScheduleProfile::PowerBidDependencyKind

literal	value	description
exclusive		Bids are exclusive depending on each other. Only dependee (parent) bids can be activated if dependent (child) is activated or vice versa. For instance, dependent (child) must not be activated if the dependee (parent) is activated.
inclusive		Bids are inclusive depending on each other. Both dependee (parent) and dependent (child) bids must be activated, i.e., dependent (child) must be activated if the dependee (parent) is activated.
restrictive		Bids are restrictive depending on each other. The activation of dependent (child) bids is optional if the dependee (parent) is activated.

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 52 shows all attributes of GridStateIntensitySchedule.

Table 52 – 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	(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 53 shows all association ends of GridStateIntensitySchedule with other classes.

Table 53 – Association ends of StateInstructionScheduleProfile::GridStateIntensitySchedule with other classes

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

1005

1006 **3.56 (NC) ValueOffsetKind enumeration**

1007 The kind of the value offset.

1008 Table 54 shows all literals of ValueOffsetKind.

1009

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.

1010

1011 **3.57 (NC) PowerRemedialActionSchedule**1012 Inheritance path = [BaseIrregularTimeSeries](#) : [BaseTimeSeries](#) : [IdentifiedObject](#)

1013 The schedule for a power remedial action.

1014 Table 55 shows all attributes of PowerRemedialActionSchedule.

1015

Table 55 – Attributes of**StateInstructionScheduleProfile::PowerRemedialActionSchedule**

1016

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

1017

1018 Table 56 shows all association ends of PowerRemedialActionSchedule with other classes.

1019 **Table 56 – Association ends of****StateInstructionScheduleProfile::PowerRemedialActionSchedule with other classes**

1020

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

1021

1022 **3.58 (NC) PowerRemedialActionTimePoint root class**

1023 Regulating values at a given point in time.

1024 Table 57 shows all attributes of PowerRemedialActionTimePoint.

**Table 57 – 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 58 shows all association ends of PowerRemedialActionTimePoint with other classes.

**Table 58 – 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 59 shows all attributes of PowerShiftKeyDistribution.

Table 59 – 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 60 shows all association ends of PowerShiftKeyDistribution with other classes.

**Table 60 – 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 61 shows all attributes of UnitCostSchedule.

Table 61 – Attributes of StateInstructionScheduleProfile::UnitCostSchedule

name	mult	type	description
currency	0..1	Currency	(NC) Currency the energy price is given in.
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 62 shows all association ends of UnitCostSchedule with other classes.

Table 62 – 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 63 shows all attributes of UnitCostTimePoint.

Table 63 – Attributes of StateInstructionScheduleProfile::UnitCostTimePoint

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

Table 64 shows all association ends of UnitCostTimePoint with other classes.

Table 64 – 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 (NC) RemedialActionSchemeSchedule

Inheritance path = [BaseIrregularTimeSeries](#) : [BaseTimeSeries](#) : [IdentifiedObject](#)

The schedule for a remedial action scheme.

Table 65 shows all attributes of RemedialActionSchemeSchedule.

Table 65 – 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 66 shows all association ends of RemedialActionSchemeSchedule with other classes.

Table 66 – 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.

3.66 (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.

3.67 (NC) RemedialActionSchemeTimePoint root class

Remedial action scheme at a given point in time.

Table 67 shows all attributes of RemedialActionSchemeTimePoint.

Table 67 – 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.

name	mult	type	description
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 68 shows all association ends of RemedialActionSchemeTimePoint with other classes.

**Table 68 – Association ends of
StateInstructionScheduleProfile::RemedialActionSchemeTimePoint with other classes**

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

3.68 (abstract,NC) PowerRemedialAction root class

Energy remedial action describes actions to rearrange power schedules.

3.69 (NC) RemedialActionGroupSchedule

Inheritance path = [BaseIrregularTimeSeries](#) : [BaseTimeSeries](#) : [IdentifiedObject](#)

The schedule for a remedial action group.

Table 69 shows all attributes of RemedialActionGroupSchedule.

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

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 70 shows all association ends of RemedialActionGroupSchedule with other classes.

**Table 70 – 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.

3.70 (NC) RemedialActionGroupTimePoint root class

Remedial action group at a given point in time.

Table 71 shows all attributes of RemedialActionGroupTimePoint.

**Table 71 – 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 72 shows all association ends of RemedialActionGroupTimePoint with other classes.

**Table 72 – 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.

3.71 (abstract,NC) RemedialActionGroup root class

Grouping of remedial actions that can be operated together.

3.72 (NC) GridStateAlterationSchedule

Inheritance path = [BaseIrregularTimeSeries](#) : [BaseTimeSeries](#) : [IdentifiedObject](#)

Schedule for a grid state alteration.

Table 73 shows all attributes of GridStateAlterationSchedule.

Table 73 – 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 74 shows all association ends of GridStateAlterationSchedule with other classes.

**Table 74 – 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.

3.73 (NC) GridStateAlterationTimePoint root class

Grid state alteration at a given point in time.

Table 75 shows all attributes of GridStateAlterationTimePoint.

1134 **Table 75 – 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. 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. $V=T/\text{sum}(\text{PF})$. In the case of priority strategy, the item with the lowest number gets allocated energy first. 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%.

1135

1136 Table 76 shows all association ends of GridStateAlterationTimePoint with other classes.

1137

1138 **Table 76 – 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.

1139

1140 **3.74 (abstract,NC) GridStateAlteration root class**

1141 Grid state alteration is a change of values describing state (operating point) of one element in
 1142 the grid model compared to the base case.

1143 **3.75 (abstract,NC) BaseRegularIntervalSchedule**1144 Inheritance path = [BaseTimeSeries](#) : [IdentifiedObject](#)

1145 Time series that has regular points in time.

1146 Table 77 shows all attributes of BaseRegularIntervalSchedule.

1147 **Table 77 – 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

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

Table 78 shows all association ends of BaseRegularIntervalSchedule with other classes.

**Table 78 – 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.

3.76 Season

Inheritance path = [IdentifiedObject](#)

A specified time period of the year.

Table 79 shows all attributes of Season.

Table 79 – 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

3.77 (NC) HourPattern

Inheritance path = [IdentifiedObject](#)

Pattern of hourly period in a day with the same kind of intensity.

Table 80 shows all attributes of HourPattern.

Table 80 – 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	EnergyScenarioKind	(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

3.78 (NC) HourPeriod root class

Period of hours in a day.

Table 81 shows all attributes of HourPeriod.

1168

Table 81 – 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. 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.

1169

1170

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

1171

Table 82 – Association ends of StateInstructionScheduleProfile::HourPeriod with other classes

1172

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

1173

1174

3.79 (NC) AssessedElementRegularSchedule

1175

Inheritance path = [BaseRegularIntervalSchedule](#) : [BaseTimeSeries](#) : [IdentifiedObject](#)

1176

Regular schedule for assessed element.

1177

Table 83 shows all attributes of AssessedElementRegularSchedule.

1178

Table 83 – Attributes of

1179

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

1180

1181

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

Table 84 – 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

3.80 (NC) AssessedElementRegularTimePoint root class

Assessed element instruction value at a given point in time.

Table 85 shows all attributes of AssessedElementRegularTimePoint.

Table 85 – 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. 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.
positiveVirtualMargin	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). The allowed value range is [0,100].

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

Table 86 – 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.

3.81 (NC) DayOfWeekKind enumeration

The kind of day to be included in a regular schedule.

Table 87 shows all literals of DayOfWeekKind.

Table 87 – 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		Day of the week other than Sunday or Saturday.
weekend		Day of the week which is Sunday or Saturday.
all		All days of the week.
publicHoliday		Public holiday is an officially designated day when most workplaces and schools are closed.
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.

3.82 (NC) PeakKind enumeration

Kind of time period with similar intensity.

Table 88 shows all literals of PeakKind.

Table 88 – 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.

3.83 (NC) EnergyScenarioKind enumeration

Kind of energy scenario.

1207 Table 89 shows all literals of EnergyScenarioKind.

1208 **Table 89 – Literals of StateInstructionScheduleProfile::EnergyScenarioKind**

literal	value	description
consumption		Scenario focus on consumption of energy.
production		Scenario focus on production of energy.
storage		Scenario focus on storage of energy.
export		Scenario focus on export of energy.
import		Scenario focus on import of energy.

1209

1210 **3.84 (abstract,NC) EnergyGroup root class**

1211 An energy group is an aggregation of energy components which have the same energy
1212 characteristic, e.g. fuel type and technology. It can be used to allocate energy.

1213 **3.85 (abstract,NC) EnergyBlockOrder root class**

1214 The energy block order is a block (an amount) of energy that forms the sequence of orders that
1215 are going to be distributed to an energy block component.

1216 **3.86 (abstract,NC) AutomationFunction root class**

1217 Automation function is a collection of functional block or other automation function that can be
1218 executed as a work cycle program as part of an automated system.

1219 **3.87 (abstract,NC) FunctionBlock root class**

1220 Function block is a function described as a set of elementary blocks. The blocks describe the
1221 function between input variables and output variables.

1222 **3.88 (NC) VoltageAngleSchedule**

1223 Inheritance path = [BaseIrregularTimeSeries](#) : [BaseTimeSeries](#) : [IdentifiedObject](#)

1224 The schedule for a voltage angle.

1225 Table 90 shows all attributes of VoltageAngleSchedule.

1226 **Table 90 – 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

1227

1228 Table 91 shows all association ends of VoltageAngleSchedule with other classes.

1229 **Table 91 – Association ends of StateInstructionScheduleProfile::VoltageAngleSchedule**
1230 **with other classes**

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

1231

3.89 (NC) VoltageAngleTimePoint root class

Voltage angle at a given point in time.

Table 92 shows all attributes of VoltageAngleTimePoint.

Table 92 – 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 93 shows all association ends of VoltageAngleTimePoint with other classes.

Table 93 – 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.

3.90 (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.

3.91 (NC) InfeedLimitSchedule

Inheritance path = [BaseIrregularTimeSeries](#) : [BaseTimeSeries](#) : [IdentifiedObject](#)

The schedule for an infeed limit.

Table 94 shows all attributes of InfeedLimitSchedule.

Table 94 – 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 95 shows all association ends of InfeedLimitSchedule with other classes.

Table 95 – 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 schedule.

1255 3.92 (abstract,NC) InfeedLimit root class

1256 Infeed limit set constraints fed in to the network by two or more terminals.

1257 3.93 (NC) InfeedLimitTimePoint root class

1258 Infeed limit at a given point in time.

1259 Table 96 shows all attributes of InfeedLimitTimePoint.

1260 **Table 96 – 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.

1261

1262 Table 97 shows all association ends of InfeedLimitTimePoint with other classes.

1263 **Table 97 – Association ends of StateInstructionScheduleProfile::InfeedLimitTimePoint**
1264 **with other classes**

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

1265

1266 3.94 (abstract,NC) PowerTransferCorridor root class

1267 A power transfer corridor is defined as a set of circuits (transmission lines or transformers)
1268 separating two portions of the power system, or a subset of circuits exposed to a substantial
1269 portion of the transmission exchange between two parts of the system.

1270 3.95 Time primitive

1271 Time as "hh:mm:ss.sss", which conforms with ISO 8601. UTC time zone is specified as
1272 "hh:mm:ss.sssZ". A local timezone relative UTC is specified as "hh:mm:ss.sss±hh:mm". The
1273 second component (shown here as "ss.sss") could have any number of digits in its fractional
1274 part to allow any kind of precision beyond seconds.

1275 3.96 CurrentFlow datatype

1276 Electrical current with sign convention: positive flow is out of the conducting equipment into the
1277 connectivity node. Can be both AC and DC.

1278 Table 98 shows all attributes of CurrentFlow.

1279 **Table 98 – Attributes of StateInstructionScheduleProfile::CurrentFlow**

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

1280

1281 3.97 MonthDay primitive

1282 MonthDay format as "--mm-dd", which conforms with XSD data type gMonthDay.

1283 3.98 AngleDegrees datatype

1284 Measurement of angle in degrees.

1285 Table 99 shows all attributes of AngleDegrees.

1286 **Table 99 – Attributes of StateInstructionScheduleProfile::AngleDegrees**

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

1287

1288 **3.99 (abstract,NC) CrossBorderRelevance root class**

1289 Combination of an assessed element and one or more bidding zone border that are affected by
1290 the assessment.

1291 **3.100 (NC) SecuredExclusionReasonKind enumeration**

1292 The kind of secured exclusion reason.

1293 Table 100 shows all literals of SecuredExclusionReasonKind.

1294 **Table 100 – 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.

1295

1296 **3.101 (abstract) EnergySource root class**

1297 A generic equivalent for an energy supplier on a transmission or distribution voltage level.

1298 **3.102 (abstract) ExternalNetworkInjection root class**

1299 This class represents the external network and it is used for IEC 60909 calculations.

1300 **3.103 (NC) StageTriggerSchedule**

1301 Inheritance path = [BaseIrregularTimeSeries](#) : [BaseTimeSeries](#) : [IdentifiedObject](#)

1302 Schedule for a stage trigger.

1303 Table 101 shows all attributes of StageTriggerSchedule.

1304 **Table 101 – 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

1305

1306 Table 102 shows all association ends of StageTriggerSchedule with other classes.

Table 102 – 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.

3.104 (NC) StageTriggerTimePoint root class

Stage trigger values at a given point in time.

Table 103 shows all attributes of StageTriggerTimePoint.

Table 103 – 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 104 shows all association ends of StageTriggerTimePoint with other classes.

Table 104 – 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.

3.105 (abstract,NC) StageTrigger root class

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

3.106 (NC) GenericSequenceSchedule

Inheritance path = [BaseIrregularTimeSeries](#) : [BaseTimeSeries](#) : [IdentifiedObject](#)

Schedule for sequence.

Table 105 shows all attributes of GenericSequenceSchedule.

Table 105 – 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 106 shows all association ends of GenericSequenceSchedule with other classes.

**Table 106 – 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.

3.107 (NC) SequenceTimePoint root class

Sequence at a given point in time.

Table 107 shows all attributes of SequenceTimePoint.

Table 107 – 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 108 shows all association ends of SequenceTimePoint with other classes.

Table 108 – 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.

3.108 (NC) ReferenceValueKind enumeration

Kind of reference value.

Table 109 shows all literals of ReferenceValueKind.

Table 109 – Literals of StateInstructionScheduleProfile::ReferenceValueKind

literal	value	description
exact		Exact value applies.
minimum		The value is a minimum.
maximum		The value is a maximum.

3.109 (NC) BaseCaseCurrentLimitSchedule

Inheritance path = [BaseIrregularTimeSeries](#) : [BaseTimeSeries](#) : [IdentifiedObject](#)

The schedule for a base case current limit.

Table 110 shows all attributes of BaseCaseCurrentLimitSchedule.

**Table 110 – Attributes of
StateInstructionScheduleProfile::BaseCaseCurrentLimitSchedule**

name	mult	type	description
interpolationKind	1..1	TimeSeriesInterpolationKind	(NC) inherited from: BaseTimeSeries
timeSeriesKind	0..1	BaseTimeSeriesKind	(NC) inherited from: BaseTimeSeries

name	mult	type	description
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 111 shows all association ends of BaseCaseCurrentLimitSchedule with other classes.

**Table 111 – Association ends of
StateInstructionScheduleProfile::BaseCaseCurrentLimitSchedule with other classes**

mult from	name	mult to	type	description
0..*	BaseCaseCurrentLimit	1..1	BaseCaseCurrentLimit	(NC) Base case current limit which has base case current limit schedule.

3.110 (abstract,NC) BaseCaseCurrentLimit root class

Operational limit on current applicable for base case.

3.111 (NC) BaseCaseCurrentLimitTimePoint root class

Base case current limit at a given point in time.

Table 112 shows all attributes of BaseCaseCurrentLimitTimePoint.

**Table 112 – Attributes of
StateInstructionScheduleProfile::BaseCaseCurrentLimitTimePoint**

name	mult	type	description
atTime	1..1	DateTime	(NC) The time the data is valid for.
value	1..1	CurrentFlow	(NC) Value of current limit. The attribute shall be a positive value or zero.

Table 113 shows all association ends of BaseCaseCurrentLimitTimePoint with other classes.

**Table 113 – Association ends of
StateInstructionScheduleProfile::BaseCaseCurrentLimitTimePoint with other classes**

mult from	name	mult to	type	description
1..*	BaseCaseCurrentLimitSchedule	1..1	BaseCaseCurrentLimitSchedule	(NC) Base case current limit schedule that has time point.

3.112 (NC) FuelStorageSchedule

Inheritance path = [BaseIrregularTimeSeries](#) : [BaseTimeSeries](#) : [IdentifiedObject](#)

Fuel storage schedule for elements.

Table 114 shows all attributes of FuelStorageSchedule.

Table 114 – Attributes of StateInstructionScheduleProfile::FuelStorageSchedule

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

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 115 shows all association ends of FuelStorageSchedule with other classes.

Table 115 – Association ends of StateInstructionScheduleProfile::FuelStorageSchedule with other classes

mult from	name	mult to	type	description
0..*	FuelStorage	1..1	FuelStorage	(NC) Fuel storage of this fuel storage schedule.

3.113 (NC) FuelStorageTimePoint root class

Fuel storage values for a given point in time.

Table 116 shows all attributes of FuelStorageTimePoint.

Table 116 – Attributes of StateInstructionScheduleProfile::FuelStorageTimePoint

name	mult	type	description
atTime	0..1	DateTime	(NC) The time the data is valid for.
energyStorage	0..1	RealEnergy	(NC) Amount of energy available in the storage.

Table 117 shows all association ends of FuelStorageTimePoint with other classes.

Table 117 – Association ends of StateInstructionScheduleProfile::FuelStorageTimePoint with other classes

mult from	name	mult to	type	description
1..*	FuelStorageSchedule	1..1	FuelStorageSchedule	(NC) The fuel storage schedule that has this time point.

3.114 (NC) ReservoirSchedule

Inheritance path = [BaseIrregularTimeSeries](#) : [BaseTimeSeries](#) : [IdentifiedObject](#)

Reservoir schedule for elements.

Table 118 shows all attributes of ReservoirSchedule.

Table 118 – Attributes of StateInstructionScheduleProfile::ReservoirSchedule

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 119 shows all association ends of ReservoirSchedule with other classes.

Table 119 – Association ends of StateInstructionScheduleProfile::ReservoirSchedule with other classes

mult from	name	mult to	type	description
0..*	Reservoir	1..1	Reservoir	(NC) The reservoir that has a schedule.

3.115 (NC) ReservoirTimePoint root class

Reservoir values for a given point in time.

Table 120 shows all attributes of ReservoirTimePoint.

Table 120 – Attributes of StateInstructionScheduleProfile::ReservoirTimePoint

name	mult	type	description
atTime	0..1	DateTime	(NC) The time the data is valid for.
energyStorage	0..1	RealEnergy	(NC) Amount of energy available in the storage.

Table 121 shows all association ends of ReservoirTimePoint with other classes.

Table 121 – Association ends of StateInstructionScheduleProfile::ReservoirTimePoint with other classes

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

3.116 (abstract,NC) FuelStorage root class

Fuel storage. e.g. pile of coal that can be shared between multiple thermal generating units.

3.117 (abstract) Reservoir root class

A water storage facility within a hydro system, including: ponds, lakes, lagoons, and rivers. The storage is usually behind some type of dam.

1414 **Annex A (informative): Sample data**

1415 **A.1 General**

1416 This Annex is designed to illustrate the profile by using fragments of sample data. It is not meant
1417 to be a complete set of examples covering all possibilities of using the profile. Defining a
1418 complete set of test data is considered a separate activity to be performed for the purpose of
1419 setting up interoperability testing and conformity related to this profile.

1420 **A.2 Sample instance data**

1421 Test data files are available in the CIM EG SharePoint.

1422

1423