



European Network of
Transmission System Operators
for Electricity

REMEDIAL ACTION SCHEDULE PROFILE SPECIFICATION

2025-02-13

APPROVED DOCUMENT
VERSION 2.3.2

1 Copyright notice:

2 **Copyright © ENTSO-E. All Rights Reserved.**

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

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

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

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

17 **NOTE CONCERNING WORDING USED IN THIS DOCUMENT**

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

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

32

33

Revision History

Version	Date	Paragraph	Comments
1.0.0	2021-03-22		Document for SOC approval
2.0.0	2021-10-12		For CIM EG review. Association to OverlappingZone was added to the profile.
2.0.0	2022-02-16		SOC approved.
2.1.0	2022-09-21		SOC approved.
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	2025-01-04		For CIM WG review. ICTC approval on 13 February 2025.

34	CONTENTS		
35	Copyright notice:.....		2
36	Revision History.....		3
37	CONTENTS		4
38	1 Introduction		8
39	2 Application profile specification		8
40	2.1 Version information		8
41	2.2 Constraints naming convention		8
42	2.3 Profile constraints		9
43	2.4 Metadata.....		13
44	2.4.1 Constraints		13
45	2.4.2 Reference metadata		13
46	3 Detailed Profile Specification		13
47	3.1 General.....		13
48	3.2 (abstract,NC) BaseIrregularTimeSeries		14
49	3.3 (abstract,NC) BaseTimeSeries		14
50	3.4 (abstract) Contingency root class		15
51	3.5 (NC) EventSchedule		15
52	3.6 (NC) EventTimePoint root class		15
53	3.7 (abstract,NC) GenericValueSchedule		15
54	3.8 (NC) GenericValueTimePoint root class		16
55	3.9 (abstract,NC) GridStateAlteration		16
56	3.10 (NC) GridStateIntensitySchedule		17
57	3.11 (abstract) IdentifiedObject root class		17
58	3.12 (abstract,NC) RemedialAction		18
59	3.13 (NC) RemedialActionCost root class		18
60	3.14 (NC) RemedialActionSchedule		19
61	3.15 (NC) RemedialActionScheduleAcceptance root class		19
62	3.16 (abstract,NC) Region root class		20
63	3.17 (abstract,NC) SecurityCoordinator root class		20
64	3.18 (abstract,NC) SystemOperator root class		20
65	3.19 (NC) CostSettledKind enumeration		20
66	3.20 (NC) RemedialActionScheduleAcceptanceKind enumeration		20
67	3.21 (NC) TimeSeriesInterpolationKind enumeration		21
68	3.22 (NC) RemedialActionScheduleStatusKind enumeration		21
69	3.23 UnitMultiplier enumeration		21
70	3.24 UnitSymbol enumeration		22
71	3.25 (NC) ValueOffsetKind enumeration		23
72	3.26 ActivePower datatype		23
73	3.27 RealEnergy datatype.....		23
74	3.28 Seconds datatype		24
75	3.29 Boolean primitive		24
76	3.30 DateTime primitive		24
77	3.31 Duration primitive.....		24

78	3.32	Decimal primitive	24
79	3.33	Float primitive	24
80	3.34	String primitive	24
81	3.35	(NC) PowerSchedule	24
82	3.36	(NC) PowerTimePoint root class	24
83	3.37	ReactivePower datatype	25
84	3.38	Integer primitive	25
85	3.39	(abstract,NC) PowerBidSchedule	25
86	3.40	(abstract,NC) AvailabilityRemedialAction	26
87	3.41	(NC,Description) SchemeRemedialAction	26
88	3.42	(NC,Description) AvailabilitySchedule	26
89	3.43	Currency enumeration	27
90	3.44	(NC) ProposingRemedialActionScheduleShare root class	31
91	3.45	(NC) RemedialActionScheduleGroup	32
92	3.46	(abstract,NC) PowerScheduleAction	32
93	3.47	(NC) CountertradeScheduleAction	32
94	3.48	(NC) RedispatchScheduleAction	33
95	3.49	(NC,Description) RemedialActionSchemeSchedule	34
96	3.50	(NC) RemedialActionScheduleDependency root class	34
97	3.51	(NC) RemedialActionScheduleDependencyKind enumeration	35
98		Annex A (informative): Sample data	36
99	A.1	General	36
100	A.2	Sample instance data	36
101			
102		List of figures	
103		Figure 1 – Other diagram	
104		RemedialActionScheduleProfile::RemedialActionScheduleProfile	14
105			
106		List of tables	
107		Table 1 – Attributes of RemedialActionScheduleProfile::BaselrregularTimeSeries	14
108		Table 2 – Attributes of RemedialActionScheduleProfile::BaseTimeSeries	14
109		Table 3 – Attributes of RemedialActionScheduleProfile::EventSchedule	15
110		Table 4 – Association ends of RemedialActionScheduleProfile::EventSchedule with	
111		other classes	15
112		Table 5 – Attributes of RemedialActionScheduleProfile::EventTimePoint	15
113		Table 6 – Association ends of RemedialActionScheduleProfile::EventTimePoint with	
114		other classes	15
115		Table 7 – Attributes of RemedialActionScheduleProfile::GenericValueSchedule	16
116		Table 8 – Association ends of RemedialActionScheduleProfile::GenericValueSchedule	
117		with other classes	16
118		Table 9 – Attributes of RemedialActionScheduleProfile::GenericValueTimePoint	16
119		Table 10 – Association ends of	
120		RemedialActionScheduleProfile::GenericValueTimePoint with other classes	16

121	Table 11 – Attributes of RemedialActionScheduleProfile::GridStateAlteration	16
122	Table 12 – Attributes of RemedialActionScheduleProfile::GridStateIntensitySchedule	17
123	Table 13 – Association ends of	
124	RemedialActionScheduleProfile::GridStateIntensitySchedule with other classes	17
125	Table 14 – Attributes of RemedialActionScheduleProfile::IdentifiedObject	17
126	Table 15 – Attributes of RemedialActionScheduleProfile::RemedialAction	18
127	Table 16 – Attributes of RemedialActionScheduleProfile::RemedialActionCost	18
128	Table 17 – Association ends of RemedialActionScheduleProfile::RemedialActionCost	
129	with other classes	19
130	Table 18 – Attributes of RemedialActionScheduleProfile::RemedialActionSchedule	19
131	Table 19 – Association ends of	
132	RemedialActionScheduleProfile::RemedialActionSchedule with other classes	19
133	Table 20 – Attributes of	
134	RemedialActionScheduleProfile::RemedialActionScheduleAcceptance	20
135	Table 21 – Association ends of	
136	RemedialActionScheduleProfile::RemedialActionScheduleAcceptance with other	
137	classes	20
138	Table 22 – Literals of RemedialActionScheduleProfile::CostSettledKind	20
139	Table 23 – Literals of	
140	RemedialActionScheduleProfile::RemedialActionScheduleAcceptanceKind	21
141	Table 24 – Literals of RemedialActionScheduleProfile::TimeSeriesInterpolationKind	21
142	Table 25 – Literals of	
143	RemedialActionScheduleProfile::RemedialActionScheduleStatusKind	21
144	Table 26 – Literals of RemedialActionScheduleProfile::UnitMultiplier	22
145	Table 27 – Literals of RemedialActionScheduleProfile::UnitSymbol	22
146	Table 28 – Literals of RemedialActionScheduleProfile::ValueOffsetKind	23
147	Table 29 – Attributes of RemedialActionScheduleProfile::ActivePower	23
148	Table 30 – Attributes of RemedialActionScheduleProfile::RealEnergy	23
149	Table 31 – Attributes of RemedialActionScheduleProfile::Seconds	24
150	Table 32 – Attributes of RemedialActionScheduleProfile::PowerSchedule	24
151	Table 33 – Attributes of RemedialActionScheduleProfile::PowerTimePoint	25
152	Table 34 – Association ends of RemedialActionScheduleProfile::PowerTimePoint with	
153	other classes	25
154	Table 35 – Attributes of RemedialActionScheduleProfile::ReactivePower	25
155	Table 36 – Attributes of RemedialActionScheduleProfile::PowerBidSchedule	26
156	Table 37 – Attributes of RemedialActionScheduleProfile::AvailabilityRemedialAction	26
157	Table 38 – Attributes of RemedialActionScheduleProfile::SchemeRemedialAction	26
158	Table 39 – Attributes of RemedialActionScheduleProfile::AvailabilitySchedule	26
159	Table 40 – Association ends of RemedialActionScheduleProfile::AvailabilitySchedule	
160	with other classes	27
161	Table 41 – Literals of RemedialActionScheduleProfile::Currency	27
162	Table 42 – Attributes of	
163	RemedialActionScheduleProfile::ProposingRemedialActionScheduleShare	31

164	Table 43 – Association ends of	
165	RemedialActionScheduleProfile::ProposingRemedialActionScheduleShare with other	
166	classes	31
167	Table 44 – Attributes of	
168	RemedialActionScheduleProfile::RemedialActionScheduleGroup	32
169	Table 45 – Attributes of RemedialActionScheduleProfile::PowerScheduleAction	32
170	Table 46 – Association ends of RemedialActionScheduleProfile::PowerScheduleAction	
171	with other classes	32
172	Table 47 – Attributes of RemedialActionScheduleProfile::CountertradeScheduleAction	33
173	Table 48 – Association ends of	
174	RemedialActionScheduleProfile::CountertradeScheduleAction with other classes	33
175	Table 49 – Attributes of RemedialActionScheduleProfile::RedispatchScheduleAction	33
176	Table 50 – Association ends of	
177	RemedialActionScheduleProfile::RedispatchScheduleAction with other classes	33
178	Table 51 – Attributes of	
179	RemedialActionScheduleProfile::RemedialActionSchemeSchedule	34
180	Table 52 – Association ends of	
181	RemedialActionScheduleProfile::RemedialActionSchemeSchedule with other classes	34
182	Table 53 – Attributes of	
183	RemedialActionScheduleProfile::RemedialActionScheduleDependency	34
184	Table 54 – Association ends of	
185	RemedialActionScheduleProfile::RemedialActionScheduleDependency with other	
186	classes	35
187	Table 55 – Literals of	
188	RemedialActionScheduleProfile::RemedialActionScheduleDependencyKind	35
189		

190 1 Introduction

191 The remedial action schedule profile is a profile to exchange a list of proposed, agreed, rejected,
192 etc. remedial action schedules.

193 Each grid state alteration defined as part of an available remedial action (by the remedial action
194 profile) gets a schedule for the parameter that should be modified when the remedial action
195 schedule is agreed and ordered. The remedial action schedule profile allows for several data
196 exchanges:

- 197 - List of remedial action schedules as output from a security analysis
- 198 - An exchange of the status of the remedial action
- 199 - An exchange of the agreements per TSO.

200

201 2 Application profile specification

202 2.1 Version information

203 The content is generated from UML model file CIM17-2_CGMES31v01_PROF-
204 20v02_NC23v69_MS10v01_DES10v01.eap.

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

- 206 - Title: Remedial Action Schedule Vocabulary
- 207 - Keyword: RAS
- 208 - Description: This vocabulary is describing the remedial action schedule profile.
- 209 - Version IRI: <https://ap-voc.cim4.eu/RemedialActionSchedule/2.3>
- 210 - Version info: 2.3.2
- 211 - Prior version: <http://entsoe.eu/ns/CIM/RemedialActionSchedule-EU/2.2>
- 212 - Conforms to: urn:iso:std:iec:61970-600-2:ed-1|urn:iso:std:iec:61970-301:ed-
213 7:amd1|file://iec61970cim17v40_iec61968cim13v13a_iec62325cim03v17a.eap|urn:iso:
214 std:iec:61970-401:draft:ed-1|urn:iso:std:iec:61970-501:draft:ed-
215 2|file://CIM100_CGMES31v01_501-20v02_NC23v62_MM10v01.eap
- 216 - Identifier: urn:uuid:6e90c546-3c6c-471b-8040-e05037081c59

217

218 2.2 Constraints naming convention

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

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

222 where

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

224 rule.Standard: the number of the standard e.g. 301 for 61970-301, 456 for 61970-456, 13 for
225 61968-13. 61970-600 specific constraints refer to 600 although they are related to one or
226 combination of the 61970-450 series profiles. For NC profiles, NC is used.

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

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

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

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

234 2.3 Profile constraints

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

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

- 240 • C:452:ALL:NA:datatypes

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

- 243 • R:452:ALL:NA:exchange

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

- 246 • R:452:ALL:NA:exchange1

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

- 252 • R:452:ALL:NA:exchange2

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

- 261 • R:452:ALL:NA:exchange3

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

- 265 • R:452:ALL:NA:exchange4

- 266 From the standpoint of the model import used by a data recipient, the document
267 describes a subset of the CIM that importing software shall be able to interpret in order
268 to import exported models. Data providers are free to exceed the minimum requirements
269 described herein as long as their resulting data files are compliant with the CIM Schema
270 and the conventions established in Clause 5. The document, therefore, describes
271 additional classes and class data that, although not required, exporters will, in all
272 likelihood, choose to include in their data files. The additional classes and data are
273 labelled as required (cardinality 1..1) or as optional (cardinality 0..1) to distinguish them
274 from their required counterparts. Please note, however, that data importers could
275 potentially receive data containing instances of any and all classes described by the
276 CIM Schema.
- 277 • R:452:ALL:NA:cardinality
- 278 The cardinality defined in the CIM model shall be followed, unless a more restrictive
279 cardinality is explicitly defined in this document. For instance, the cardinality on the
280 association between VoltageLevel and BaseVoltage indicates that a VoltageLevel shall
281 be associated with one and only one BaseVoltage, but a BaseVoltage can be associated
282 with zero to many VoltageLevels.
- 283 • R:452:ALL:NA:associations
- 284 Associations between classes referenced in this document and classes not referenced
285 here are not required regardless of cardinality.
- 286 • R:452:ALL:IdentifiedObject.name:rule
- 287 The attribute “name” inherited by many classes from the abstract class IdentifiedObject
288 is not required to be unique. It must be a human readable identifier without additional
289 embedded information that would need to be parsed. The attribute is used for purposes
290 such as User Interface and data exchange debugging. The MRID defined in the data
291 exchange format is the only unique and persistent identifier used for this data exchange.
292 The attribute IdentifiedObject.name is, however, always required for CoreEquipment
293 profile and Short Circuit profile.
- 294 • R:452:ALL:IdentifiedObject.description:rule
- 295 The attribute “description” inherited by many classes from the abstract class
296 IdentifiedObject must contain human readable text without additional embedded
297 information that would need to be parsed.
- 298 • R:452:ALL:NA:uniqueIdentifier
- 299 All IdentifiedObject-s shall have a persistent and globally unique identifier (Master
300 Resource Identifier - mRID).
- 301 • R:452:ALL:NA:unitMultiplier
- 302 For exchange of attributes defined using CIM Data Types (ActivePower, Susceptance,
303 etc.) a unit multiplier of 1 is used if the UnitMultiplier specified in this document is “none”.
- 304 • C:452:ALL:IdentifiedObject.name:stringLength
- 305 The string IdentifiedObject.name has a maximum of 128 characters.
- 306 • C:452:ALL:IdentifiedObject.description:stringLength
- 307 The string IdentifiedObject.description is maximum 256 characters.

308 • C:452:ALL:NA:float

309 An attribute that is defined as float (e.g. has a type Float or a type which is a Datatype
310 with .value attribute of type Float) shall support ISO/IEC 60559:2020 for floating-point
311 arithmetic using single precision floating point. A single precision float supports 7
312 significant digits where the significant digits are described as an integer, or a decimal
313 number with 6 decimal digits. Two float values are equal when the significant with 7
314 digits are identical, e.g. 1234567 is equal 1.234567E6 and so are 1.2345678 and
315 1.234567E0.

316 • R:NC:ALL:Region:reference

317 The reference to the Region is normally a reference to the capacity calculation region,
318 which is identified by “Y” EIC code of the capacity calculation region.

319 • R:NC:ALL:SystemOperator:reference

320 The reference to the System Operator is normally identified by “X” EIC code of TSO.

321 • C:NC:RAS:RemedialActionSchedule:proposingEntity

322 The RemedialActionSchedule shall have a proposing entity either the Security
323 Coordinator (RemedialActionSchedule.ProposingEntity) or at least one
324 ProposingRemedialActionScheduleShare.

325 • R:NC:ALL:NA:serialization

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

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

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

- 336 ○ Case 1: An association end refers to an abstract class
- 337 ○ Case 2: An abstract class (stereotyped with “Description”) has an association
338 (direction to another class)
- 339 ○ Case 3: An abstract class (not stereotyped with “Description”) has an
340 association (direction to another class)
- 341 ○ Case 4: An abstract class has attributes and subclasses are not in the profile

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

349 It should be taken into account that this approach deviates from MVAL5 (IEC 61970-
350 600-1:2021), which creates multiple inheritance at serialization. For instance, with this
351 more explicit exchange the serialization of the association between abstract class
352 Equipment and abstract class Circuit for a PowerTransformer will be serialized as
353 follows:

354 ○ for association
 355 <cim:PowerTransformer rdf:about="_c328f787-bc17-47ad-a59f-6ba7133340d0">
 356 <nc:Equipment.Circuit rdf:resource="#_9ced16ac-d076-4ef9-a241-a998a579e77b"/>
 357 </cim:PowerTransformer>

358 ○ for attribute
 359 <cim:ACLineSegment rdf:about="_04f681aa-6999-4fb3-9775-acia5eb7ceff">
 360 <cim:Equipment.inService>true</cim:Equipment.inService>
 361 </cim:ACLineSegment>

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

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

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

- 368 • C:NC:RAS:RemedialActionScheduleGroup.GroupDependency:cardinality

369 A RemedialActionScheduleGroup shall have at least 3
 370 RemedialActionScheduleDependency referenced.

- 371 • C:NC:RAS:RemedialActionScheduleGroup:dependency

372 A RemedialActionScheduleGroup shall have at least one
 373 RemedialActionScheduleDependency with association end
 374 RemedialActionScheduleDependency.ProposedRemedialActionSchedule and at least
 375 one RemedialActionScheduleDependency with association end
 376 RemedialActionScheduleDependency.ReplacedRemedialActionSchedule.

- 377 • C:NC:RAS:RemedialActionScheduleDependency.kind:cardinality

378 RemedialActionScheduleDependency.kind is required for all
 379 RemedialActionScheduleDependency objects part of a
 380 RemedialActionScheduleGroup.

- 381 • C:NC:RAS:RemedialActionScheduleDependency.kind:applicability

382 RemedialActionScheduleDependency.kind shall not be provided for
 383 RemedialActionScheduleDependency objects that are not part of a
 384 RemedialActionScheduleGroup.

- 385 • C:NC:RAS:RemedialActionScheduleDependency:associations

386 When a RemedialActionScheduleDependency is part of a
 387 RemedialActionScheduleGroup, it is required to provide either
 388 RemedialActionScheduleDependency.ProposedRemedialActionSchedule or
 389 RemedialActionScheduleDependency.ReplacedRemedialActionSchedule (i.e., not both
 390 associations).

391 When a RemedialActionScheduleDependency is not part of a
392 RemedialActionScheduleGroup, it is required to provide both
393 RemedialActionScheduleDependency.ProposedRemedialActionSchedule and
394 RemedialActionScheduleDependency.ReplacedRemedialActionSchedule.

395

396 2.4 Metadata

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

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

406 2.4.1 Constraints

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

- 409 • R:NC:ALL:wasAttributedTo:usage

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

412

413 2.4.2 Reference metadata

414 The header defined for this profile requires availability of a set of reference metadata. For
415 instance, the attribute prov:wasGeneratedBy requires a reference to an activity which produced
416 the model or the related process. The activities are defined as reference metadata and their
417 identifiers are referenced from the header to enable the receiving entity to retrieve the “static”
418 (reference) information that is not modified frequently. This approach imposes a requirement
419 that both the sending entity and the receiving entity have access to a unique version of the
420 reference metadata. Therefore, each business process shall define which reference metadata
421 is used and where it is located.

422 3 Detailed Profile Specification

423 3.1 General

424 This package contains remedial action schedule profile.

440

441 **3.4 (abstract) Contingency root class**

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

443 **3.5 (NC) EventSchedule**444 Inheritance path = [BaseIrregularTimeSeries](#) : [BaseTimeSeries](#) : [IdentifiedObject](#)

445 Time series represent irregular event described by event points in time.

446 Table 3 shows all attributes of EventSchedule.

447

Table 3 – Attributes of RemedialActionScheduleProfile::EventSchedule

name	mult	type	description
interpolationKind	1..1	TimeSeriesInterpolationKind	(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

448

449 Table 4 shows all association ends of EventSchedule with other classes.

450 **Table 4 – Association ends of RemedialActionScheduleProfile::EventSchedule with**
451 **other classes**

mult from	name	mult to	type	description
0..1	RemedialActionSchedule	0..1	RemedialActionSchedule	Remedial action schedule is the event that is validity for the given time series.

452

453 **3.6 (NC) EventTimePoint root class**

454 Event valid for a given point in time.

455 Table 5 shows all attributes of EventTimePoint.

456

Table 5 – Attributes of RemedialActionScheduleProfile::EventTimePoint

name	mult	type	description
atTime	0..1	DateTime	(NC) The time the data is valid for.
isActive	0..1	Boolean	(NC) True, if the event is occurring (Active) at this time point. Otherwise false.

457

458 Table 6 shows all association ends of EventTimePoint with other classes.

459 **Table 6 – Association ends of RemedialActionScheduleProfile::EventTimePoint with**
460 **other classes**

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

461

462 **3.7 (abstract,NC) GenericValueSchedule**463 Inheritance path = [BaseIrregularTimeSeries](#) : [BaseTimeSeries](#) : [IdentifiedObject](#)

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

466 Table 7 shows all attributes of GenericValueSchedule.

467 **Table 7 – Attributes of RemedialActionScheduleProfile::GenericValueSchedule**

name	mult	type	description
interpolationKind	1..1	TimeSeriesInterpolationKind	(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

468

469 Table 8 shows all association ends of GenericValueSchedule with other classes.

470 **Table 8 – Association ends of RemedialActionScheduleProfile::GenericValueSchedule**
471 **with other classes**

mult from	name	mult to	type	description
0..*	RemedialActionSchedule	0..1	RemedialActionSchedule	(NC) Remedial action schedule which has generic value schedules.

472

473 **3.8 (NC) GenericValueTimePoint root class**

474 Generic value for a given point in time.

475 Table 9 shows all attributes of GenericValueTimePoint.

476 **Table 9 – Attributes of RemedialActionScheduleProfile::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.

477

478 Table 10 shows all association ends of GenericValueTimePoint with other classes.

479 **Table 10 – Association ends of RemedialActionScheduleProfile::GenericValueTimePoint**
480 **with other classes**

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

481

482 **3.9 (abstract,NC) GridStateAlteration**483 Inheritance path = [IdentifiedObject](#)

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

486 Table 11 shows all attributes of GridStateAlteration.

487 **Table 11 – Attributes of RemedialActionScheduleProfile::GridStateAlteration**

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

488

489 **3.10 (NC) GridStateIntensitySchedule**

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

492 Defines the intensity applied for a given grid state alteration. It is primarily used in exchanges
493 related to the remedial action schedule. The value provided by the schedule replaces the value
494 of the attribute to which the schedule refers to.

495 Table 12 shows all attributes of GridStateIntensitySchedule.

496 **Table 12 – Attributes of RemedialActionScheduleProfile::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
description	0..1	String	inherited from: IdentifiedObject
mRID	1..1	String	inherited from: IdentifiedObject
name	0..1	String	inherited from: IdentifiedObject

497

498 Table 13 shows all association ends of GridStateIntensitySchedule with other classes.

499 **Table 13 – Association ends of**
500 **RemedialActionScheduleProfile::GridStateIntensitySchedule with other classes**

mult from	name	mult to	type	description
0..*	GridStateAlteration	1..1	GridStateAlteration	(NC) The grid state alteration which has intensity.
0..*	RemedialActionSchedule	0..1	RemedialActionSchedule	(NC) inherited from: GenericValueSchedule

501

502 **3.11 (abstract) IdentifiedObject root class**

503 This is a root class to provide common identification for all classes needing identification and
504 naming attributes.

505 Table 14 shows all attributes of IdentifiedObject.

506 **Table 14 – Attributes of RemedialActionScheduleProfile::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	mult	type	description
name	0..1	String	The name is any free human readable and possibly non unique text naming the object.

507

508 **3.12 (abstract,NC) RemedialAction**509 Inheritance path = [IdentifiedObject](#)510 Remedial action describes one or more actions that can be performed on a given power system
511 model situation to eliminate one or more identified breaches of constraints. The remedial action
512 can be costly, and have a cost characteristic, or non costly.

513 Table 15 shows all attributes of RemedialAction.

514 **Table 15 – Attributes of RemedialActionScheduleProfile::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

515

516 **3.13 (NC) RemedialActionCost root class**

517 Remedial action cost is the total cost itemised cost by category and type for the remedial action.

518 Table 16 shows all attributes of RemedialActionCost.

519 **Table 16 – Attributes of RemedialActionScheduleProfile::RemedialActionCost**

name	mult	type	description
costAllocationTime	0..1	DateTime	(NC) Cost allocation time is the time the cost shall be allocated.
kind	1..1	CostSettledKind	(NC) Remedial action cost category related to the confirmation of the cost in regards to changes.
operationalCost	0..1	Decimal	(NC) Operational cost is the total cost directly related to operate the unit according to the remedial action, e.g. fuel cost.
opportunityCost	0..1	Decimal	(NC) Opportunity cost is the total cost of potential earning that is missed due to performing the remedial action.
otherCost	0..1	Decimal	(NC) Other cost is the total cost that cannot be directly allocated to any of the other items.
processingFee	0..1	Decimal	(NC) Processing fee is the total cost for processing the remedial action.
savedFuelCost	0..1	Decimal	(NC) Saved fuel cost is the total saving due to not consuming the expected fuel as part of the remedial action.
shutdownCost	0..1	Decimal	(NC) Shutdown cost is the total cost for shutting down a unit as part of the remedial action.
startupCost	0..1	Decimal	(NC) Start-up cost is the total cost for activating the remedial action, e.g. if a generator needs to be started before it can perform the remedial action.
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.

name	mult	type	description
			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.

520

521 Table 17 shows all association ends of RemedialActionCost with other classes.

522 **Table 17 – Association ends of RemedialActionScheduleProfile::RemedialActionCost**
523 **with other classes**

mult from	name	mult to	type	description
0..*	RemedialActionSchedule	1..1	RemedialActionSchedule	(NC) Remedial action schedule for which this remedial action cost relates to.

524

525 **3.14 (NC) RemedialActionSchedule**526 Inheritance path = [IdentifiedObject](#)

527 A schedule for a determined remedial action.

528 Table 18 shows all attributes of RemedialActionSchedule.

529 **Table 18 – Attributes of RemedialActionScheduleProfile::RemedialActionSchedule**

name	mult	type	description
statusKind	1..1	RemedialActionScheduleStatusKind	(NC) Indicates the status kind for the remedial action schedule.
statusReason	0..1	String	(NC) Description of reasoning for the status. For instance, in case of rejected remedial action, the reason for this rejection is described here.
totalCostCurrency	0..1	Currency	(NC) The currency of the total cost.
description	0..1	String	inherited from: IdentifiedObject
mRID	1..1	String	inherited from: IdentifiedObject
name	0..1	String	inherited from: IdentifiedObject

530

531 Table 19 shows all association ends of RemedialActionSchedule with other classes.

532 **Table 19 – Association ends of**
533 **RemedialActionScheduleProfile::RemedialActionSchedule with other classes**

mult from	name	mult to	type	description
0..*	Contingency	0..1	Contingency	(NC) The contingency for a curative remedial action schedule.
0..*	RemedialAction	1..1	RemedialAction	(NC) The remedial action that has a remedial action schedule associated.
0..*	ProposingEntity	0..1	SecurityCoordinator	(NC) The security coordinator that is proposing this remedial action schedule.
0..*	AssignedRegion	0..1	Region	(NC) The assigned region for this remedial action schedule.

534

535 **3.15 (NC) RemedialActionScheduleAcceptance root class**

536 It identifies if the remedial action schedule is accepted for a given system operator.

537 Table 20 shows all attributes of RemedialActionScheduleAcceptance.

538
539**Table 20 – Attributes of RemedialActionScheduleProfile::RemedialActionScheduleAcceptance**

name	mult	type	description
kind	1..1	RemedialActionScheduleAcceptanceKind	(NC) The kind of the remedial action acceptance.
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.

540
541

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

542
543
544**Table 21 – Association ends of RemedialActionScheduleProfile::RemedialActionScheduleAcceptance with other classes**

mult from	name	mult to	type	description
0..*	RemedialActionSchedule	1..1	RemedialActionSchedule	(NC) A remedial action schedule for which a remedial action schedule acceptance is reported.
0..*	RespondingEntity	1..1	SystemOperator	(NC) An entity for which a remedial action schedule acceptances are reported.

545

3.16 (abstract,NC) Region root class

547 A region where the system operator belongs to.

3.17 (abstract,NC) SecurityCoordinator root class

549 A role that coordinates the relevant remedial actions and their optimisation to ensure efficient
550 use to achieve required operational security of the power system.

3.18 (abstract,NC) SystemOperator root class

552 System operator.

3.19 (NC) CostSettledKind enumeration

554 Kind describing how settled the cost is in regards to changes.

555 Table 22 shows all literals of CostSettledKind.

556

Table 22 – Literals of RemedialActionScheduleProfile::CostSettledKind

literal	value	description
indicative		Indicative cost.
provisional		Provisional cost.
final		Final cost. For instance, the cost is not expected to be changed on a later stage.

557

3.20 (NC) RemedialActionScheduleAcceptanceKind enumeration

559 The kind of acceptance for a remedial action schedule.

560 Table 23 shows all literals of RemedialActionScheduleAcceptanceKind.

561
562**Table 23 – Literals of RemedialActionScheduleProfile::RemedialActionScheduleAcceptanceKind**

literal	value	description
accepted		The acceptance of remedial action schedule is concluded and accepted.
refused		The acceptance of the remedial action schedule is concluded and refused.
waiting		The acceptance of the remedial action schedule is waiting (in progress).
timeout		The acceptance of the remedial action schedule was not completed due to timeout.

563

3.21 (NC) TimeSeriesInterpolationKind enumeration

565 Kinds of interpolation of values between two time point.

566 Table 24 shows all literals of TimeSeriesInterpolationKind.

Table 24 – Literals of RemedialActionScheduleProfile::TimeSeriesInterpolationKind

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

568

3.22 (NC) RemedialActionScheduleStatusKind enumeration

570 Remedial action schedule status kinds.

571 Table 25 shows all literals of RemedialActionScheduleStatusKind.

572
573**Table 25 – Literals of RemedialActionScheduleProfile::RemedialActionScheduleStatusKind**

literal	value	description
proposed		Proposed remedial action schedule.
agreed		Agreed remedial action schedule.
rejected		Rejected remedial action schedule.
ordered		Ordered remedial action schedule.
previouslyAgreed		Previously agreed remedial action schedule.
notUsed		Not used remedial action schedule.
agreementValidated		The agreement is validated for the remedial action schedule.
rejectionValidated		The rejection is validated for the remedial action schedule.
implemented		An ordered remedial action is implemented.
activated		Activated remedial action schedule.

574

3.23 UnitMultiplier enumeration

576 The unit multipliers defined for the CIM. When applied to unit symbols, the unit symbol is
 577 treated as a derived unit. Regardless of the contents of the unit symbol text, the unit symbol
 578 shall be treated as if it were a single-character unit symbol. Unit symbols should not contain
 579 multipliers, and it should be left to the multiplier to define the multiple for an entire data type.

580 For example, if a unit symbol is "m2Pers" and the multiplier is "k", then the value is $k(m^{**2}/s)$,
581 and the multiplier applies to the entire final value, not to any individual part of the value. This
582 can be conceptualized by substituting a derived unit symbol for the unit type. If one imagines
583 that the symbol "P" represents the derived unit "m2Pers", then applying the multiplier "k" can
584 be conceptualized simply as "kP".

585 For example, the SI unit for mass is "kg" and not "g". If the unit symbol is defined as "kg", then
586 the multiplier is applied to "kg" as a whole and does not replace the "k" in front of the "g". In
587 this case, the multiplier of "m" would be used with the unit symbol of "kg" to represent one gram.
588 As a text string, this violates the instructions in IEC 80000-1. However, because the unit symbol
589 in CIM is treated as a derived unit instead of as an SI unit, it makes more sense to conceptualize
590 the "kg" as if it were replaced by one of the proposed replacements for the SI mass symbol. If
591 one imagines that the "kg" were replaced by a symbol "P", then it is easier to conceptualize the
592 multiplier "m" as creating the proper unit "mP", and not the forbidden unit "mkg".

593 Table 26 shows all literals of UnitMultiplier.

594 **Table 26 – Literals of RemedialActionScheduleProfile::UnitMultiplier**

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

595

596 3.24 UnitSymbol enumeration

597 The derived units defined for usage in the CIM. In some cases, the derived unit is equal to an
598 SI unit. Whenever possible, the standard derived symbol is used instead of the formula for the
599 derived unit. For example, the unit symbol Farad is defined as "F" instead of "CPerV". In cases
600 where a standard symbol does not exist for a derived unit, the formula for the unit is used as
601 the unit symbol. For example, density does not have a standard symbol and so it is represented
602 as "kgPerm3". With the exception of the "kg", which is an SI unit, the unit symbols do not contain
603 multipliers and therefore represent the base derived unit to which a multiplier can be applied as
604 a whole.

605 Every unit symbol is treated as an unparseable text as if it were a single-letter symbol. The
606 meaning of each unit symbol is defined by the accompanying descriptive text and not by the
607 text contents of the unit symbol.

608 To allow the widest possible range of serializations without requiring special character handling,
609 several substitutions are made which deviate from the format described in IEC 80000-1. The
610 division symbol "/" is replaced by the letters "Per". Exponents are written in plain text after the
611 unit as "m3" instead of being formatted as "m" with a superscript of 3 or introducing a symbol
612 as in "m^3". The degree symbol "°" is replaced with the letters "deg". Any clarification of the
613 meaning for a substitution is included in the description for the unit symbol.

614 Non-SI units are included in list of unit symbols to allow sources of data to be correctly labelled
615 with their non-SI units (for example, a GPS sensor that is reporting numbers that represent feet
616 instead of meters). This allows software to use the unit symbol information correctly convert
617 and scale the raw data of those sources into SI-based units.

618 The integer values are used for harmonization with IEC 61850.

619 Table 27 shows all literals of UnitSymbol.

620 **Table 27 – Literals of RemedialActionScheduleProfile::UnitSymbol**

literal	value	description
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 $VI\cos(\phi)$), is expressed in Watts. See also apparent power and reactive power.
VAr	63	Reactive power in volt amperes reactive. The "reactive" or "imaginary" component of electrical

literal	value	description
		power (VIsin(phi)). (See also real power and apparent power). Note: Different meter designs use different methods to arrive at their results. Some meters may compute reactive power as an arithmetic value, while others compute the value vectorially. The data consumer should determine the method in use and the suitability of the measurement for the intended purpose.
Wh	72	Real energy in watt hours.

621

622 **3.25 (NC) ValueOffsetKind enumeration**

623 The kind of the value offset.

624 Table 28 shows all literals of ValueOffsetKind.

625

Table 28 – Literals of RemedialActionScheduleProfile::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.

626

627 **3.26 ActivePower datatype**628 Product of RMS value of the voltage and the RMS value of the in-phase component of the
629 current.

630 Table 29 shows all attributes of ActivePower.

631

Table 29 – Attributes of RemedialActionScheduleProfile::ActivePower

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

632

633 **3.27 RealEnergy datatype**

634 Real electrical energy.

635 Table 30 shows all attributes of RealEnergy.

636

Table 30 – Attributes of RemedialActionScheduleProfile::RealEnergy

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

637

638 3.28 Seconds datatype

639 Time, in seconds.

640 Table 31 shows all attributes of Seconds.

641 **Table 31 – Attributes of RemedialActionScheduleProfile::Seconds**

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

642

643 3.29 Boolean primitive

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

645 3.30 DateTime primitive

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

650 3.31 Duration primitive

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

656 3.32 Decimal primitive

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

658 3.33 Float primitive

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

660 3.34 String primitive

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

663 3.35 (NC) PowerSchedule664 Inheritance path = [BaseIrregularTimeSeries](#) : [BaseTimeSeries](#) : [IdentifiedObject](#)

665 Time series represent irregular power, active and reactive, values at given points in time.

666 Table 32 shows all attributes of PowerSchedule.

667 **Table 32 – Attributes of RemedialActionScheduleProfile::PowerSchedule**

name	mult	type	description
interpolationKind	1..1	TimeSeriesInterpolationKind	(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

668

669 3.36 (NC) PowerTimePoint root class

670 Power, active and reactive, value at a given point in time.

671 Table 33 shows all attributes of PowerTimePoint.

672

Table 33 – Attributes of RemedialActionScheduleProfile::PowerTimePoint

name	mult	type	description
atTime	1..1	DateTime	(NC) The time the data is valid for.
p	0..1	ActivePower	(NC) Active power injection. Load sign convention is used, i.e. positive sign means flow out from a node.
q	0..1	ReactivePower	(NC) Reactive power injection. Load sign convention is used, i.e. positive sign means flow out from a node.
price	0..1	Decimal	(NC) Price for the scheduled active power per unit of active power. e.g. per MW.
activatedP	0..1	ActivePower	(NC) Active power activated as part of redispatch. Negative number means that the value is scheduling down. Positive number means that the value is scheduling up.
activatedPrice	0..1	Decimal	(NC) Price for the activated active power per unit e.g. per MW.

673

674

Table 34 shows all association ends of PowerTimePoint with other classes.

675

676

Table 34 – Association ends of RemedialActionScheduleProfile::PowerTimePoint with other classes

mult from	name	mult to	type	description
0..*	PowerSchedule	1..1	PowerSchedule	(NC) Time series the time point values belongs to.

677

3.37 ReactivePower datatype

679 Product of RMS value of the voltage and the RMS value of the quadrature component of the
680 current.

681 Table 35 shows all attributes of ReactivePower.

682

Table 35 – Attributes of RemedialActionScheduleProfile::ReactivePower

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

683

3.38 Integer primitive

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

686

3.39 (abstract,NC) PowerBidSchedule

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

688 Power bid or offer related to a redispatch or countertrading measures. In the case of market
689 place for economic efficiency of the bids and offers, this is equivalent to BidTimeSeries class
690 in 62325 package.

691 Table 36 shows all attributes of PowerBidSchedule.

692 **Table 36 – Attributes of RemedialActionScheduleProfile::PowerBidSchedule**

name	mult	type	description
interpolationKind	1..1	TimeSeriesInterpolationKind	(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

693

694 **3.40 (abstract,NC) AvailabilityRemedialAction**695 Inheritance path = [RemedialAction](#) : [IdentifiedObject](#)696 Availability remedial action is a remedial action that cancels or reschedules an availability
697 schedule.

698 Table 37 shows all attributes of AvailabilityRemedialAction.

699 **Table 37 – Attributes of RemedialActionScheduleProfile::AvailabilityRemedialAction**

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

700

701 **3.41 (NC,Description) SchemeRemedialAction**702 Inheritance path = [RemedialAction](#) : [IdentifiedObject](#)703 Scheme remedial action is remedial action that involves a scheme that can include conditional
704 logic and stages of grid alteration. The primary remedial action is the arming of these schemes,
705 that will then perform curative remedial action when the condition is met. System Integrity
706 Protection Scheme (SIPS) and Special Protection Scheme (SPS) are example of this.

707 Table 38 shows all attributes of SchemeRemedialAction.

708 **Table 38 – Attributes of RemedialActionScheduleProfile::SchemeRemedialAction**

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

709

710 **3.42 (NC,Description) AvailabilitySchedule**711 Inheritance path = [IdentifiedObject](#)712 A given (un)availability schedule with a given status and cause that include multiple equipment
713 that need to follow the same scheduling periods.

714 Table 39 shows all attributes of AvailabilitySchedule.

715 **Table 39 – Attributes of RemedialActionScheduleProfile::AvailabilitySchedule**

name	mult	type	description
isCancelled	1..1	Boolean	(NC) Defines the cancelling of the availability schedule. True means that is cancelling, False means that it is not cancelling.
priority	1..1	Integer	(NC) Value 0 means ignore priority. 1 means the highest priority, 2 is the second highest priority.

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

716

717

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

718

Table 40 – Association ends of RemedialActionScheduleProfile::AvailabilitySchedule with other classes

719

mult from	name	mult to	type	description
1..*	RemedialAction	1..1	AvailabilityRemedialAction	(NC) Remedial action that is cancelling this availability schedule.

720

721

3.43 Currency enumeration

722

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

723

Table 41 shows all literals of Currency.

724

Table 41 – Literals of RemedialActionScheduleProfile::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.

literal	value	description
BYR	974	Belarusian ruble.
BZD	084	Belize dollar.
CAD	124	Canadian dollar.
CDF	976	Congolese franc.
CHF	756	Swiss franc.
CLF	990	Unidad de Fomento (funds code), Chile.
CLP	152	Chilean peso.
CNY	156	Chinese yuan.
COP	170	Colombian peso.
COU	970	Unidad de Valor Real.
CRC	188	Costa Rican colon.
CUC	931	Cuban convertible peso.
CUP	192	Cuban peso.
CVE	132	Cape Verde escudo.
CZK	203	Czech koruna.
DJF	262	Djiboutian franc.
DKK	208	Danish krone.
DOP	214	Dominican peso.
DZD	012	Algerian dinar.
EEK	233	Estonian kroon.
EGP	818	Egyptian pound.
ERN	232	Eritrean nakfa.
ETB	230	Ethiopian birr.
EUR	978	Euro.
FJD	242	Fiji dollar.
FKP	238	Falkland Islands pound.
GBP	826	Pound sterling.
GEL	981	Georgian lari.
GHS	936	Ghanaian cedi.
GIP	929	Gibraltar pound.
GMD	270	Gambian dalasi.
GNF	324	Guinean franc.
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.

literal	value	description
INR	356	Indian rupee.
IQD	368	Iraqi dinar.
IRR	364	Iranian rial.
ISK	352	Icelandic króna.
JMD	388	Jamaican dollar.
JOD	400	Jordanian dinar.
JPY	392	Japanese yen.
KES	404	Kenyan shilling.
KGS	417	Kyrgyzstani som.
KHR	116	Cambodian riel.
KMF	174	Comoro franc.
KPW	408	North Korean won.
KRW	410	South Korean won.
KWD	414	Kuwaiti dinar.
KYD	136	Cayman Islands dollar.
KZT	398	Kazakhstani tenge.
LAK	418	Lao kip.
LBP	422	Lebanese pound.
LKR	144	Sri Lanka rupee.
LRD	430	Liberian dollar.
LSL	426	Lesotho loti.
LTL	440	Lithuanian litas.
LVL	428	Latvian lats.
LYD	434	Libyan dinar.
MAD	504	Moroccan dirham.
MDL	498	Moldovan leu.
MGA	969	Malagasy ariary.
MKD	807	Macedonian denar.
MMK	104	Myanma kyat.
MNT	496	Mongolian tugrik.
MOP	446	Macanese pataca.
MRO	478	Mauritanian ouguiya.
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.

literal	value	description
NOK	578	Norwegian krone.
NPR	524	Nepalese rupee.
NZD	554	New Zealand dollar.
OMR	512	Omani rial.
PAB	590	Panamanian balboa.
PEN	604	Peruvian nuevo sol.
PGK	598	Papua New Guinean kina.
PHP	608	Philippine peso.
PKR	586	Pakistani rupee.
PLN	985	Polish zloty.
PYG	600	Paraguayan guaraní.
QAR	634	Qatari rial.
RON	946	Romanian new leu.
RSD	941	Serbian dinar.
RUB	643	Russian rouble.
RWF	646	Rwandan franc.
SAR	682	Saudi riyal.
SBD	090	Solomon Islands dollar.
SCR	690	Seychelles rupee.
SDG	938	Sudanese pound.
SEK	752	Swedish krona/kronor.
SGD	702	Singapore dollar.
SHP	654	Saint Helena pound.
SLL	694	Sierra Leonean leone.
SOS	706	Somali shilling.
SRD	968	Surinamese dollar.
STD	678	São Tomé and Príncipe dobra.
SYP	760	Syrian pound.
SZL	748	Lilangeni.
THB	764	Thai baht.
TJS	972	Tajikistani somoni.
TMT	934	Turkmenistani manat.
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.

literal	value	description
UYU	858	Uruguayan peso.
UZS	860	Uzbekistan som.
VEF	937	Venezuelan bolívar fuerte.
VND	704	Vietnamese Dong.
VUV	548	Vanuatu vatu.
WST	882	Samoan tala.
XAF	950	CFA franc BEAC.
XCD	951	East Caribbean dollar.
XOF	952	CFA Franc BCEAO.
XPF	953	CFP franc.
YER	886	Yemeni rial.
ZAR	710	South African rand.
ZMK	894	Zambian kwacha.
ZWL	932	Zimbabwe dollar.

725

726 **3.44 (NC) ProposingRemedialActionScheduleShare root class**727 Proposing entity (System Operator) with a proper cost share for a given remedial action
728 schedule.

729 Table 42 shows all attributes of ProposingRemedialActionScheduleShare.

730

731

**Table 42 – Attributes of
RemedialActionScheduleProfile::ProposingRemedialActionScheduleShare**

name	mult	type	description
costSharingFactor	0..1	Decimal	(NC) Sharing factor of the cost of the remedial action as a fraction of the total cost, i.e. system operator's cost = cost x (costSharingFactor / sum of all costSharingFactor).
mRID	0..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.

732

733 Table 43 shows all association ends of ProposingRemedialActionScheduleShare with other
734 classes.

735

736

737

**Table 43 – Association ends of
RemedialActionScheduleProfile::ProposingRemedialActionScheduleShare with other
classes**

mult from	name	mult to	type	description
0..*	RemedialActionSchedule	1..1	RemedialActionSchedule	(NC) Remedial action schedule proposed by the proposing entity.

mult from	name	mult to	type	description
0..*	ProposingEntity	1..1	SystemOperator	(NC) Proposing entity making the proposing remedial action schedule share.

738

739 **3.45 (NC) RemedialActionScheduleGroup**740 Inheritance path = [IdentifiedObject](#)741 Remedial action schedule group collects two or more remedial action schedules together. The
742 remedial action schedule group needs to be set up for the same remedial action or proposing
743 alternative remedial action by including a reference to another remedial action.

744 Table 44 shows all attributes of RemedialActionScheduleGroup.

745

Table 44 – Attributes of

746

RemedialActionScheduleProfile::RemedialActionScheduleGroup

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

747

748 **3.46 (abstract,NC) PowerScheduleAction**749 Inheritance path = [IdentifiedObject](#)

750 Power schedule action is an action to rearrange power schedules.

751 Table 45 shows all attributes of PowerScheduleAction.

752

Table 45 – Attributes of RemedialActionScheduleProfile::PowerScheduleAction

name	mult	type	description
currency	0..1	Currency	(NC) Currency the energy price is given in.
energyPrice	0..1	Decimal	(NC) Energy price for the power schedule action.
description	0..1	String	inherited from: IdentifiedObject
mRID	1..1	String	inherited from: IdentifiedObject
name	0..1	String	inherited from: IdentifiedObject

753

754 Table 46 shows all association ends of PowerScheduleAction with other classes.

Table 46 – Association ends of RemedialActionScheduleProfile::PowerScheduleAction with other classes

755

756

mult from	name	mult to	type	description
0..*	RemedialActionSchedule	1..1	RemedialActionSchedule	(NC) Remedial action schedule which power schedule actions.
0..1	PowerBidSchedule	0..1	PowerBidSchedule	(NC) The power bid schedule which contains the power schedule action.
0..1	PowerSchedule	1..1	PowerSchedule	(NC) Power schedule which contains the power schedule action.

757

758 **3.47 (NC) CountertradeScheduleAction**759 Inheritance path = [PowerScheduleAction](#) : [IdentifiedObject](#)

760 Countertrade schedule action is an action to rearrange power schedules based on a Generation
761 and Load Shift Key (GLSK) strategy.
762 Table 47 shows all attributes of CountertradeScheduleAction.

763 **Table 47 – Attributes of RemedialActionScheduleProfile::CountertradeScheduleAction**

name	mult	type	description
currency	0..1	Currency	(NC) inherited from: PowerScheduleAction
energyPrice	0..1	Decimal	(NC) inherited from: PowerScheduleAction
description	0..1	String	inherited from: IdentifiedObject
mRID	1..1	String	inherited from: IdentifiedObject
name	0..1	String	inherited from: IdentifiedObject

764
765 Table 48 shows all association ends of CountertradeScheduleAction with other classes.

766 **Table 48 – Association ends of**
767 **RemedialActionScheduleProfile::CountertradeScheduleAction with other classes**

mult from	name	mult to	type	description
0..*	RemedialActionSchedule	1..1	RemedialActionSchedule	(NC) inherited from: PowerScheduleAction
0..1	PowerBidSchedule	0..1	PowerBidSchedule	(NC) inherited from: PowerScheduleAction
0..1	PowerSchedule	1..1	PowerSchedule	(NC) inherited from: PowerScheduleAction

768

769 3.48 (NC) RedispatchScheduleAction

770 Inheritance path = [PowerScheduleAction](#) : [IdentifiedObject](#)

771 Redispatch schedule action is an action to rearrange power schedules for a scheduled resource
772 to obtain a feasible and secure operational state of the power electricity system.

773 Table 49 shows all attributes of RedispatchScheduleAction.

774 **Table 49 – Attributes of RemedialActionScheduleProfile::RedispatchScheduleAction**

name	mult	type	description
currency	0..1	Currency	(NC) inherited from: PowerScheduleAction
energyPrice	0..1	Decimal	(NC) inherited from: PowerScheduleAction
description	0..1	String	inherited from: IdentifiedObject
mRID	1..1	String	inherited from: IdentifiedObject
name	0..1	String	inherited from: IdentifiedObject

775

776 Table 50 shows all association ends of RedispatchScheduleAction with other classes.

777 **Table 50 – Association ends of**
778 **RemedialActionScheduleProfile::RedispatchScheduleAction with other classes**

mult from	name	mult to	type	description
0..*	RemedialActionSchedule	1..1	RemedialActionSchedule	(NC) inherited from: PowerScheduleAction
0..1	PowerBidSchedule	0..1	PowerBidSchedule	(NC) inherited from: PowerScheduleAction

mult from	name	mult to	type	description
0..1	PowerSchedule	1..1	PowerSchedule	(NC) inherited from: PowerScheduleAction

779

780 3.49 (NC,Description) RemedialActionSchemeSchedule

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

782 The schedule for a remedial action scheme.

783 Table 51 shows all attributes of RemedialActionSchemeSchedule.

784

785

**Table 51 – Attributes of
RemedialActionScheduleProfile::RemedialActionSchemeSchedule**

name	mult	type	description
interpolationKind	1..1	TimeSeriesInterpolationKind	(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

786

787 Table 52 shows all association ends of RemedialActionSchemeSchedule with other classes.

788

789

**Table 52 – Association ends of
RemedialActionScheduleProfile::RemedialActionSchemeSchedule with other classes**

mult from	name	mult to	type	description
0..1	SchemeRemedialAction	0..1	SchemeRemedialAction	(NC) Remedial action scheme which has remedial action scheme schedules.

790

791 3.50 (NC) RemedialActionScheduleDependency root class

792 Remedial action schedule dependency is making two remedial action schedules depending on each other.

794 Table 53 shows all attributes of RemedialActionScheduleDependency.

795

796

**Table 53 – Attributes of
RemedialActionScheduleProfile::RemedialActionScheduleDependency**

name	mult	type	description
kind	0..1	RemedialActionScheduleDependencyKind	(NC) Type of dependency between two remedial action schedules.
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.

797

798 Table 54 shows all association ends of RemedialActionScheduleDependency with other classes.

799

800
801
802

Table 54 – Association ends of RemedialActionScheduleProfile::RemedialActionScheduleDependency with other classes

mult from	name	mult to	type	description
0..*	ReplacedRemedialActionSchedule	0..1	RemedialActionSchedule	(NC) Remedial action schedule that is replaced by dependent remedial action schedule.
0..*	ProposedRemedialActionSchedule	0..1	RemedialActionSchedule	(NC) Proposed remedial action schedule that has dependency.
0..*	RemedialActionScheduleGroup	0..1	RemedialActionScheduleGroup	(NC) Remedial action schedule group which has remedial action schedule dependency.

803

804 3.51 (NC) RemedialActionScheduleDependencyKind enumeration

805 Kind of dependency between remedial action schedules.

806 Table 55 shows all literals of RemedialActionScheduleDependencyKind.

807
808

Table 55 – Literals of RemedialActionScheduleProfile::RemedialActionScheduleDependencyKind

literal	value	description
exclusive		Remedial action schedules are exclusive depending on each other. e.g. Only one of the remedial action schedules can be selected at the same time.
inclusive		Remedial action schedules are inclusive depending on each other. e.g. Both remedial action schedules need to be picked if one of them is needed.
none		Remedial action schedules are not depending on each other. However, the two remedial action schedules should be evaluated together.

809

810

811

812

Annex A (informative): Sample data

813 A.1 General

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

818 A.2 Sample instance data

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

820