



European Network of
Transmission System Operators
for Electricity

REMEDIAL ACTION SCHEDULE PROFILE SPECIFICATION

2026-09-10

ICTC APPROVED
VERSION 2.4.1

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
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	2024-01-04		For CIM WG review.
2.4.0	2025-09-11		ICTC approval.
2.4.1	2026-08-05		For CIM WG review.
2.4.1	2026-09-10		ICTC approval.

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	14
46	3 Detailed Profile Specification	14
47	3.1 General.....	14
48	3.2 (abstract) Contingency root class	14
49	3.3 (abstract) IdentifiedObject root class	14
50	3.4 (NC) CostSettledKind enumeration	15
51	3.5 Currency enumeration.....	15
52	3.6 (NC) RejectionReasonKind enumeration	19
53	3.7 (NC) RemedialActionScheduleDependencyKind enumeration.....	19
54	3.8 (NC) RemedialActionScheduleResponseKind enumeration	20
55	3.9 (NC) RemedialActionScheduleStatusKind enumeration	20
56	3.10 (NC) TimeSeriesInterpolationKind enumeration.....	21
57	3.11 UnitMultiplier enumeration	21
58	3.12 UnitSymbol enumeration	21
59	3.13 (NC) ValueOffsetKind enumeration	22
60	3.14 ActivePower datatype	22
61	3.15 ReactivePower datatype	23
62	3.16 RealEnergy datatype.....	23
63	3.17 Seconds datatype	23
64	3.18 Boolean primitive	23
65	3.19 DateTime primitive	23
66	3.20 Decimal primitive	24
67	3.21 Duration primitive.....	24
68	3.22 Float primitive	24
69	3.23 Integer primitive	24
70	3.24 String primitive.....	24
71	3.25 (abstract,NC) AvailabilityRemedialAction	24
72	3.26 (NC,Description) AvailabilitySchedule	24
73	3.27 (abstract,NC) BaseIrregularTimeSeries	25
74	3.28 (abstract,NC) BaseTimeSeries	25
75	3.29 (NC) CountertradeScheduleAction	25
76	3.30 (NC) EventSchedule	26
77	3.31 (NC) EventTimePoint root class	26

78	3.32	(abstract,NC) GenericValueSchedule	27
79	3.33	(NC) GenericValueTimePoint root class	27
80	3.34	(abstract,NC) GridStateAlteration	28
81	3.35	(NC) GridStateIntensitySchedule	28
82	3.36	(abstract,NC) PowerBidSchedule	29
83	3.37	(NC) PowerSchedule	29
84	3.38	(abstract,NC) PowerScheduleAction	29
85	3.39	(NC) PowerTimePoint root class	30
86	3.40	(NC) ProposingRemedialActionScheduleShare root class	30
87	3.41	(NC) RedispatchScheduleAction	31
88	3.42	(abstract,NC) Region root class	32
89	3.43	(abstract,NC) RemedialAction	32
90	3.44	(NC) RemedialActionCost root class	32
91	3.45	(NC) RemedialActionSchedule	33
92	3.46	(NC) RemedialActionScheduleDependency root class	34
93	3.47	(NC) RemedialActionScheduleGroup	34
94	3.48	(NC) RemedialActionScheduleResponse root class	35
95	3.49	(NC,Description) RemedialActionSchemeSchedule	35
96	3.50	(NC,Description) SchemeRemedialAction	36
97	3.51	(abstract,NC) SecurityCoordinator root class	36
98	3.52	(abstract,NC) SystemOperator root class	36
99		Annex A (informative): Sample data	37
100	A.1	General.....	37
101	A.2	Sample instance data.....	37

102

103 List of figures

104	Figure 1 – Other diagram	
105	RemedialActionScheduleProfile::RemedialActionScheduleProfile	14

106

107 List of tables

108	Table 1 – Attributes of RemedialActionScheduleProfile::IdentifiedObject	14
109	Table 2 – Literals of RemedialActionScheduleProfile::CostSettledKind	15
110	Table 3 – Literals of RemedialActionScheduleProfile::Currency	15
111	Table 4 – Literals of RemedialActionScheduleProfile::RejectionReasonKind	19
112	Table 5 – Literals of	
113	RemedialActionScheduleProfile::RemedialActionScheduleDependencyKind	20
114	Table 6 – Literals of	
115	RemedialActionScheduleProfile::RemedialActionScheduleResponseKind	20
116	Table 7 – Literals of	
117	RemedialActionScheduleProfile::RemedialActionScheduleStatusKind	20
118	Table 8 – Literals of RemedialActionScheduleProfile::TimeSeriesInterpolationKind	21
119	Table 9 – Literals of RemedialActionScheduleProfile::UnitMultiplier	21
120	Table 10 – Literals of RemedialActionScheduleProfile::UnitSymbol	22

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

165	Table 43 – Association ends of	
166	RemedialActionScheduleProfile::RedispatchScheduleAction with other classes	31
167	Table 44 – Attributes of RemedialActionScheduleProfile::RemedialAction	32
168	Table 45 – Attributes of RemedialActionScheduleProfile::RemedialActionCost	32
169	Table 46 – Association ends of RemedialActionScheduleProfile::RemedialActionCost	
170	with other classes	33
171	Table 47 – Attributes of RemedialActionScheduleProfile::RemedialActionSchedule	33
172	Table 48 – Association ends of	
173	RemedialActionScheduleProfile::RemedialActionSchedule with other classes	33
174	Table 49 – Attributes of	
175	RemedialActionScheduleProfile::RemedialActionScheduleDependency	34
176	Table 50 – Association ends of	
177	RemedialActionScheduleProfile::RemedialActionScheduleDependency with other	
178	classes	34
179	Table 51 – Attributes of	
180	RemedialActionScheduleProfile::RemedialActionScheduleGroup	34
181	Table 52 – Attributes of	
182	RemedialActionScheduleProfile::RemedialActionScheduleResponse	35
183	Table 53 – Association ends of	
184	RemedialActionScheduleProfile::RemedialActionScheduleResponse with other classes	35
185	Table 54 – Attributes of	
186	RemedialActionScheduleProfile::RemedialActionSchemeSchedule	35
187	Table 55 – Association ends of	
188	RemedialActionScheduleProfile::RemedialActionSchemeSchedule with other classes	36
189	Table 56 – Attributes of RemedialActionScheduleProfile::SchemeRemedialAction	36
190		

1 Introduction

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

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

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

2 Application profile specification

2.1 Version information

The content is generated from UML model file CIM17-2_CGMES31v01_PROF-20v03_NC25v25_MS10v12_DES10v01.qea.

This edition is based on the IEC 61970 UML version “”, dated “”.

- Title:
- Keyword:
- Description:
- Version IRI:
- Version info:
- Prior version:
- Conforms to:
- Identifier:

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.

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

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

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

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

232 2.3 Profile constraints

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

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

- 238 • C:452:ALL:NA:datatypes

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

- 241 • R:452:ALL:NA:exchange

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

- 244 • R:452:ALL:NA:exchange1

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

- 250 • R:452:ALL:NA:exchange2

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

- 259 • R:452:ALL:NA:exchange3

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

- 263 • R:452:ALL:NA:exchange4

264 From the standpoint of the model import used by a data recipient, the document
265 describes a subset of the CIM that importing software shall be able to interpret in order
266 to import exported models. Data providers are free to exceed the minimum requirements

described herein as long as their resulting data files are compliant with the CIM Schema and the conventions established in Clause 5. The document, therefore, describes additional classes and class data that, although not required, exporters will, in all likelihood, choose to include in their data files. The additional classes and data are labelled as required (cardinality 1..1) or as optional (cardinality 0..1) to distinguish them from their required counterparts. Please note, however, that data importers could potentially receive data containing instances of any and all classes described by the CIM Schema.

• R:452:ALL:NA:cardinality

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

• R:452:ALL:NA:associations

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

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

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

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

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

• R:452:ALL:NA:uniqueIdentifier

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

• R:452:ALL:NA:unitMultiplier

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

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

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

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

The string IdentifiedObject.description is maximum 256 characters.

• C:452:ALL:NA:float

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

314 • R:NC:ALL:Region:reference

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

317 • R:NC:ALL:SystemOperator:reference

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

319 • C:NC:RAS:RemedialActionSchedule:proposingEntity

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

323 • R:NC:ALL:NA:serialization

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

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

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

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

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

347 It should be taken into account that this approach deviates from MVAL5 (IEC 61970-
348 600-1:2021), which creates multiple inheritance at serialization. For instance, with this
349 more explicit exchange the serialization of the association between abstract class
350 Equipment and abstract class Circuit for a PowerTransformer will be serialized as
351 follows:

- 352 ○ for association

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

356 ○ for attribute

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

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

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

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

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

367 A RemedialActionScheduleGroup shall have at least 3
 368 RemedialActionScheduleDependency referenced.

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

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

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

376 RemedialActionScheduleDependency.kind is required for all
 377 RemedialActionScheduleDependency objects part of a
 378 RemedialActionScheduleGroup.

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

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

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

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

389 When a RemedialActionScheduleDependency is not part of a
 390 RemedialActionScheduleGroup, it is required to provide both

- 391 RemedialActionScheduleDependency.ProposedRemedialActionSchedule and
392 RemedialActionScheduleDependency.ReplacedRemedialActionSchedule.
- 393 • C:NC:RAS:RemedialActionScheduleResponse.rejectionReasonKind:required
- 394 RemedialActionScheduleResponse.rejectionReasonKind is required if
395 RemedialActionScheduleResponse.kind equals
396 RemedialActionScheduleResponseKind.refused.
- 397 • C:NC:RAS:RemedialActionScheduleResponse.rejectionReason:applicability
- 398 RemedialActionScheduleResponse.rejectionReason is used when
399 RemedialActionScheduleResponse.rejectionReasonKind equals
400 RejectionReasonKind.other.
- 401 This constraint has warning severity.
- 402 • C:NC:RAS:PowerScheduleAction.currency
- 403 If PowerScheduleAction.energyPrice is provided, the PowerScheduleAction.currency is
404 required.
- 405 • C:NC:RAS:PowerSchedule.currency
- 406 If PowerTimePoint.price is provided, the PowerSchedule.currency is required.
- 407 • C:NC:RAS:PowerSchedule:currencyConsistency
- 408 If PowerTimePoint.price is provided and PowerSchedule is associated with
409 PowerScheduleAction either PowerScheduleAction.currency or
410 PowerSchedule.currency shall be provided.

411

412 2.4 Metadata

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

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

422 2.4.1 Constraints

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

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

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

428

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 remedial action schedule profile.

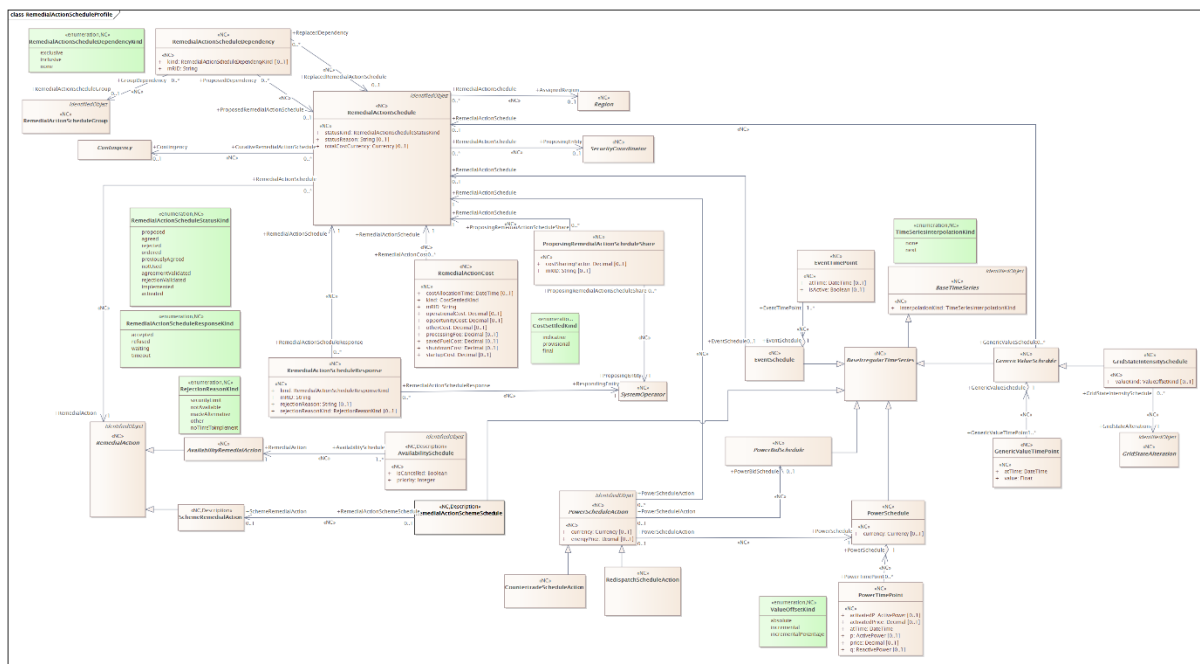


Figure 1 – Other diagram
RemedialActionScheduleProfile::RemedialActionScheduleProfile

Figure 1: The diagram contains the main classes used in the profile.

3.2 (abstract) Contingency root class

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

3.3 (abstract) IdentifiedObject root class

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

Table 1 shows all attributes of IdentifiedObject.

Table 1 – 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.

name	mult	type	description
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.

3.4 (NC) CostSettledKind enumeration

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

Table 2 shows all literals of CostSettledKind.

Table 2 – 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.

3.5 Currency enumeration

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

Table 3 shows all literals of Currency.

Table 3 – Literals of RemedialActionScheduleProfile::Currency

literal	value	description
AED		United Arab Emirates dirham.
AFN		Afghan afghani.
ALL		Albanian lek.
AMD		Armenian dram.
ANG		Netherlands Antillean guilder.
AOA		Angolan kwanza.
ARS		Argentine peso.
AUD		Australian dollar.
AWG		Aruban florin.
AZN		Azerbaijani manat.
BAM		Bosnia and Herzegovina convertible mark.
BBD		Barbados dollar.
BDT		Bangladeshi taka.
BGN		Bulgarian lev.
BHD		Bahraini dinar.
BIF		Burundian franc.

literal	value	description
BMD		Bermudian dollar (customarily known as Bermuda dollar).
BND		Brunei dollar.
BOB		Boliviano.
BOV		Bolivian Mvdol (funds code).
BRL		Brazilian real.
BSD		Bahamian dollar.
BTN		Bhutanese ngultrum.
BWP		Botswana pula.
BYR		Belarusian ruble.
BZD		Belize dollar.
CAD		Canadian dollar.
CDF		Congolese franc.
CHF		Swiss franc.
CLF		Unidad de Fomento (funds code), Chile.
CLP		Chilean peso.
CNY		Chinese yuan.
COP		Colombian peso.
COU		Unidad de Valor Real.
CRC		Costa Rican colon.
CUC		Cuban convertible peso.
CUP		Cuban peso.
CVE		Cape Verde escudo.
CZK		Czech koruna.
DJF		Djiboutian franc.
DKK		Danish krone.
DOP		Dominican peso.
DZD		Algerian dinar.
EEK		Estonian kroon.
EGP		Egyptian pound.
ERN		Eritrean nakfa.
ETB		Ethiopian birr.
EUR		Euro.
FJD		Fiji dollar.
FKP		Falkland Islands pound.
GBP		Pound sterling.
GEL		Georgian lari.
GHS		Ghanaian cedi.
GIP		Gibraltar pound.
GMD		Gambian dalasi.
GNF		Guinean franc.

literal	value	description
GTQ		Guatemalan quetzal.
GYD		Guyanese dollar.
HKD		Hong Kong dollar.
HNL		Honduran lempira.
HRK		Croatian kuna.
HTG		Haitian gourde.
HUF		Hungarian forint.
IDR		Indonesian rupiah.
ILS		Israeli new sheqel.
INR		Indian rupee.
IQD		Iraqi dinar.
IRR		Iranian rial.
ISK		Icelandic króna.
JMD		Jamaican dollar.
JOD		Jordanian dinar.
JPY		Japanese yen.
KES		Kenyan shilling.
KGS		Kyrgyzstani som.
KHR		Cambodian riel.
KMF		Comoro franc.
KPW		North Korean won.
KRW		South Korean won.
KWD		Kuwaiti dinar.
KYD		Cayman Islands dollar.
KZT		Kazakhstani tenge.
LAK		Lao kip.
LBP		Lebanese pound.
LKR		Sri Lanka rupee.
LRD		Liberian dollar.
LSL		Lesotho loti.
LTL		Lithuanian litas.
LVL		Latvian lats.
LYD		Libyan dinar.
MAD		Moroccan dirham.
MDL		Moldovan leu.
MGA		Malagasy ariary.
MKD		Macedonian denar.
MMK		Myanma kyat.
MNT		Mongolian tugrik.
MOP		Macanese pataca.
MRO		Mauritanian ouguiya.

literal	value	description
MUR		Mauritian rupee.
MVR		Maldivian rufiyaa.
MWK		Malawian kwacha.
MXN		Mexican peso.
MYR		Malaysian ringgit.
MZN		Mozambican metical.
NAD		Namibian dollar.
NGN		Nigerian naira.
NIO		Cordoba oro.
NOK		Norwegian krone.
NPR		Nepalese rupee.
NZD		New Zealand dollar.
OMR		Omani rial.
PAB		Panamanian balboa.
PEN		Peruvian nuevo sol.
PGK		Papua New Guinean kina.
PHP		Philippine peso.
PKR		Pakistani rupee.
PLN		Polish zloty.
PYG		Paraguayan guaraní.
QAR		Qatari rial.
RON		Romanian new leu.
RSD		Serbian dinar.
RUB		Russian rouble.
RWF		Rwandan franc.
SAR		Saudi riyal.
SBD		Solomon Islands dollar.
SCR		Seychelles rupee.
SDG		Sudanese pound.
SEK		Swedish krona/kronor.
SGD		Singapore dollar.
SHP		Saint Helena pound.
SLL		Sierra Leonean leone.
SOS		Somali shilling.
SRD		Surinamese dollar.
STD		São Tomé and Príncipe dobra.
SYP		Syrian pound.
SZL		Lilangeni.
THB		Thai baht.
TJS		Tajikistani somoni.
TMT		Turkmenistani manat.

literal	value	description
TND		Tunisian dinar.
TOP		Tongan pa'anga.
TRY		Turkish lira.
TTD		Trinidad and Tobago dollar.
TWD		New Taiwan dollar.
TZS		Tanzanian shilling.
UAH		Ukrainian hryvnia.
UGX		Ugandan shilling.
USD		United States dollar.
UYU		Uruguayan peso.
UZS		Uzbekistan som.
VEF		Venezuelan bolívar fuerte.
VND		Vietnamese Dong.
VUV		Vanuatu vatu.
WST		Samoan tala.
XAF		CFA franc BEAC.
XCD		East Caribbean dollar.
XOF		CFA Franc BCEAO.
XPF		CFP franc.
YER		Yemeni rial.
ZAR		South African rand.
ZMK		Zambian kwacha.
ZWL		Zimbabwe dollar.

3.6 (NC) RejectionReasonKind enumeration

The kind of rejection reason.

Table 4 shows all literals of RejectionReasonKind.

Table 4 – Literals of RemedialActionScheduleProfile::RejectionReasonKind

literal	value	description
securityLimit		The rejection is due to security limit violation.
notAvailable		The rejection is due to unavailability of the object.
other		The rejection is due to other more complex reason or the details are unknown.
madeAlternative		The rejection is due to a proposal of an alternative remedial action.
noTimeToImplement		The rejection is due to lack of time to realise the proposed remedial action.

3.7 (NC) RemedialActionScheduleDependencyKind enumeration

Kind of dependency between remedial action schedules.

Table 5 shows all literals of RemedialActionScheduleDependencyKind.

Table 5 – 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.

3.8 (NC) RemedialActionScheduleResponseKind enumeration

The kind of response for a remedial action schedule.

Table 6 shows all literals of RemedialActionScheduleResponseKind.

Table 6 – Literals of RemedialActionScheduleProfile::RemedialActionScheduleResponseKind

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.

3.9 (NC) RemedialActionScheduleStatusKind enumeration

Remedial action schedule status kinds.

Table 7 shows all literals of RemedialActionScheduleStatusKind.

Table 7 – 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.

3.10 (NC) TimeSeriesInterpolationKind enumeration

Kinds of interpolation of values between two time point.

Table 8 shows all literals of TimeSeriesInterpolationKind.

Table 8 – Literals of RemedialActionScheduleProfile::TimeSeriesInterpolationKind

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

3.11 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 9 shows all literals of UnitMultiplier.

Table 9 – Literals of RemedialActionScheduleProfile::UnitMultiplier

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

3.12 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 "m³". 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 10 shows all literals of UnitSymbol.

Table 10 – Literals of RemedialActionScheduleProfile::UnitSymbol

literal	value	description
s		Time in seconds.
W		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.
VA _r		Reactive power in volt amperes reactive. The "reactive" or "imaginary" component of electrical power ($VI\sin(\phi)$). (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		Real energy in watt hours.

3.13 (NC) ValueOffsetKind enumeration

The kind of the value offset.

Table 11 shows all literals of ValueOffsetKind.

Table 11 – 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.

3.14 ActivePower datatype

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

Table 12 shows all attributes of ActivePower.

Table 12 – Attributes of RemedialActionScheduleProfile::ActivePower

name	mult	type	description
multiplier	0..1	UnitMultiplier	
unit	0..1	UnitSymbol	
value	0..1	Float	

3.15 ReactivePower datatype

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

Table 13 shows all attributes of ReactivePower.

Table 13 – Attributes of RemedialActionScheduleProfile::ReactivePower

name	mult	type	description
value	0..1	Float	
unit	0..1	UnitSymbol	
multiplier	0..1	UnitMultiplier	

3.16 RealEnergy datatype

Real electrical energy.

Table 14 shows all attributes of RealEnergy.

Table 14 – Attributes of RemedialActionScheduleProfile::RealEnergy

name	mult	type	description
multiplier	0..1	UnitMultiplier	
unit	0..1	UnitSymbol	
value	0..1	Float	

3.17 Seconds datatype

Time, in seconds.

Table 15 shows all attributes of Seconds.

Table 15 – Attributes of RemedialActionScheduleProfile::Seconds

name	mult	type	description
value	0..1	Float	Time, in seconds
unit	0..1	UnitSymbol	
multiplier	0..1	UnitMultiplier	

3.18 Boolean primitive

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

3.19 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.

3.20 Decimal primitive

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

3.21 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.

3.22 Float primitive

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

3.23 Integer primitive

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

3.24 String primitive

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

3.25 (abstract,NC) AvailabilityRemedialAction

Inheritance path = [RemedialAction](#) : [IdentifiedObject](#)

Availability remedial action is a remedial action that cancels or reschedules an availability schedule.

Table 16 shows all attributes of AvailabilityRemedialAction.

Table 16 – 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

3.26 (NC,Description) AvailabilitySchedule

Inheritance path = [IdentifiedObject](#)

A given (un)availability schedule with a given status and cause that include multiple equipment that need to follow the same scheduling periods.

Table 17 shows all attributes of AvailabilitySchedule.

Table 17 – 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.
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 AvailabilitySchedule with other classes.

Table 18 – Association ends of RemedialActionScheduleProfile::AvailabilitySchedule with other classes

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

3.27 (abstract,NC) BaselIrregularTimeSeriesInheritance path = [BaseTimeSeries](#) : [IdentifiedObject](#)

Time series that has irregular points in time.

Table 19 shows all attributes of BaselIrregularTimeSeries.

Table 19 – Attributes of RemedialActionScheduleProfile::BaselIrregularTimeSeries

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

3.28 (abstract,NC) BaseTimeSeriesInheritance path = [IdentifiedObject](#)

Time series of values at points in time.

Table 20 shows all attributes of BaseTimeSeries.

Table 20 – Attributes of RemedialActionScheduleProfile::BaseTimeSeries

name	mult	type	description
interpolationKind	1..1	TimeSeriesInterpolationKind	(NC) Kind of interpolation done between time point.
description	0..1	String	inherited from: IdentifiedObject
mRID	1..1	String	inherited from: IdentifiedObject
name	0..1	String	inherited from: IdentifiedObject

3.29 (NC) CountertradeScheduleActionInheritance path = [PowerScheduleAction](#) : [IdentifiedObject](#)

Countertrade schedule action is an action to rearrange power schedules based on a Generation and Load Shift Key (GLSK) strategy.

Table 21 shows all attributes of CountertradeScheduleAction.

Table 21 – 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

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

Table 22 – Association ends of RemedialActionScheduleProfile::CountertradeScheduleAction with other classes

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

3.30 (NC) EventSchedule

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

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

Table 23 shows all attributes of EventSchedule.

Table 23 – 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

Table 24 shows all association ends of EventSchedule with other classes.

Table 24 – Association ends of RemedialActionScheduleProfile::EventSchedule with 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.

3.31 (NC) EventTimePoint root class

Event valid for a given point in time.

Table 25 shows all attributes of EventTimePoint.

Table 25 – 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.

Table 26 shows all association ends of EventTimePoint with other classes.

Table 26 – Association ends of RemedialActionScheduleProfile::EventTimePoint with other classes

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

3.32 (abstract,NC) GenericValueSchedule

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

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

Table 27 shows all attributes of GenericValueSchedule.

Table 27 – Attributes of 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

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

Table 28 – Association ends of RemedialActionScheduleProfile::GenericValueSchedule with other classes

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

3.33 (NC) GenericValueTimePoint root class

Generic value for a given point in time.

Table 29 shows all attributes of GenericValueTimePoint.

Table 29 – 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 1.0 equals true and 0.0 equals false. In case of integer the 0 should be included as a decimal, e.g. 2.0.

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

Table 30 – Association ends of RemedialActionScheduleProfile::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.

3.34 (abstract,NC) GridStateAlteration

Inheritance path = [IdentifiedObject](#)

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

Table 31 shows all attributes of GridStateAlteration.

Table 31 – Attributes of RemedialActionScheduleProfile::GridStateAlteration

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

3.35 (NC) GridStateIntensitySchedule

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

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

Table 32 shows all attributes of GridStateIntensitySchedule.

Table 32 – Attributes of 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

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

Table 33 – Association ends of 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

3.36 (abstract,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 34 shows all attributes of PowerBidSchedule.

Table 34 – 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

3.37 (NC) PowerSchedule

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

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

Table 35 shows all attributes of PowerSchedule.

Table 35 – Attributes of RemedialActionScheduleProfile::PowerSchedule

name	mult	type	description
currency	0..1	Currency	(NC) Currency the energy price is given in.
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

3.38 (abstract,NC) PowerScheduleAction

Inheritance path = [IdentifiedObject](#)

Power schedule action is an action to rearrange power schedules.

Table 36 shows all attributes of PowerScheduleAction.

Table 36 – 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

Table 37 shows all association ends of PowerScheduleAction with other classes.

Table 37 – Association ends of RemedialActionScheduleProfile::PowerScheduleAction with other classes

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

3.39 (NC) PowerTimePoint root class

Time series represent irregular values at given points in time.

Table 38 shows all attributes of PowerTimePoint.

Table 38 – 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 or area.
q	0..1	ReactivePower	(NC) Reactive power injection. Load sign convention is used, i.e. positive sign means flow out from a node or area.
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.

Table 39 shows all association ends of PowerTimePoint with other classes.

Table 39 – 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.

3.40 (NC) ProposingRemedialActionScheduleShare root class

Proposing entity (System Operator) with a proper cost share for a given remedial action schedule.

Table 40 shows all attributes of ProposingRemedialActionScheduleShare.

Table 40 – 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

name	mult	type	description
			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.

Table 41 shows all association ends of `ProposingRemedialActionScheduleShare` with other classes.

Table 41 – 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.
0..*	ProposingEntity	1..1	SystemOperator	(NC) Proposing entity making the proposing remedial action schedule share.

3.41 (NC) RedispatchScheduleAction

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

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

Table 42 shows all attributes of `RedispatchScheduleAction`.

Table 42 – 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

Table 43 shows all association ends of `RedispatchScheduleAction` with other classes.

Table 43 – Association ends of `RemedialActionScheduleProfile::RedispatchScheduleAction` with other classes

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

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

3.42 (abstract,NC) Region root class

A region where the system operator belongs to.

3.43 (abstract,NC) RemedialAction

Inheritance path = [IdentifiedObject](#)

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

Table 44 shows all attributes of RemedialAction.

Table 44 – 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

3.44 (NC) RemedialActionCost root class

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

Table 45 shows all attributes of RemedialActionCost.

Table 45 – 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

name	mult	type	description
			exchange context. Global uniqueness is easily achieved by using a UUID, as specified in RFC 4122, for the mRID. The use of UUID is strongly recommended. For CIMXML data files in RDF syntax conforming to IEC 61970-552, the mRID is mapped to rdf:ID or rdf:about attributes that identify CIM object elements.

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

Table 46 – Association ends of RemedialActionScheduleProfile::RemedialActionCost 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.

3.45 (NC) RemedialActionSchedule

Inheritance path = [IdentifiedObject](#)

A schedule for a determined remedial action.

Table 47 shows all attributes of RemedialActionSchedule.

Table 47 – 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

Table 48 shows all association ends of RemedialActionSchedule with other classes.

Table 48 – Association ends of 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.

3.46 (NC) RemedialActionScheduleDependency root class

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

Table 49 shows all attributes of RemedialActionScheduleDependency.

**Table 49 – 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.

Table 50 shows all association ends of RemedialActionScheduleDependency with other classes.

**Table 50 – 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.

3.47 (NC) RemedialActionScheduleGroup

Inheritance path = [IdentifiedObject](#)

Remedial action schedule group collects two or more remedial action schedules together. The remedial action schedule group needs to be set up for the same remedial action or proposing alternative remedial action by including a reference to another remedial action.

Table 51 shows all attributes of RemedialActionScheduleGroup.

**Table 51 – Attributes of
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

3.48 (NC) RemedialActionScheduleResponse root class

The response on the remedial action schedule by a given system operator.

Table 52 shows all attributes of RemedialActionScheduleResponse.

**Table 52 – Attributes of
RemedialActionScheduleProfile::RemedialActionScheduleResponse**

name	mult	type	description
kind	1..1	RemedialActionScheduleResponseKind	(NC) The kind of the remedial action response.
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.
rejectionReasonKind	0..1	RejectionReasonKind	(NC) The kind of the remedial action rejection reason.
rejectionReason	0..1	String	(NC) An additional type of rejection reason. It is used only when rejectionReasonKind equals RejectionReasonKind.other.

Table 53 shows all association ends of RemedialActionScheduleResponse with other classes.

**Table 53 – Association ends of
RemedialActionScheduleProfile::RemedialActionScheduleResponse 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 response is reported.
0..*	RespondingEntity	1..1	SystemOperator	(NC) An entity for which a remedial action schedule response is reported.

3.49 (NC,Description) RemedialActionSchemeSchedule

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

The schedule for a remedial action scheme.

Table 54 shows all attributes of RemedialActionSchemeSchedule.

**Table 54 – 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

Table 55 shows all association ends of RemedialActionSchemeSchedule with other classes.

Table 55 – 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.

3.50 (NC,Description) SchemeRemedialAction

Inheritance path = [RemedialAction](#) : [IdentifiedObject](#)

Scheme remedial action is remedial action that involves a scheme that can include conditional logic and stages of grid alteration. The primary remedial action is the arming of these schemes, that will then perform curative remedial action when the condition is met. System Integrity Protection Scheme (SIPS) and Special Protection Scheme (SPS) are example of this.

Table 56 shows all attributes of SchemeRemedialAction.

Table 56 – 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

3.51 (abstract,NC) SecurityCoordinator root class

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

3.52 (abstract,NC) SystemOperator root class

A power system operation post responsible for performing and coordinating the real-time and planning functions required to maintain the secure, reliable, and efficient operation of the power system within a defined operational scope.

Note: SystemOperator represents the execution of power system operational responsibilities in practice. While the operational post exists independently at the ecosystem level, SystemOperator identifies the specific agent fulfilling that responsibility at a given time. The operational scope may correspond to a control area, network, voltage domain, or other defined boundary.

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

840 **A.1 General**

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

845 **A.2 Sample instance data**

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

847