

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Energy management system application program interface (EMS-API) –
Part 301: Common information model (CIM) base**

**Interface de programmation d'application pour système de gestion d'énergie
(EMS-API) –
Partie 301: Base de modèle d'information commun (CIM)**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2013 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester.

If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de la CEI ou du Comité national de la CEI du pays du demandeur.

Si vous avez des questions sur le copyright de la CEI ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de la CEI de votre pays de résidence.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

Useful links:

IEC publications search - www.iec.ch/searchpub

The advanced search enables you to find IEC publications by a variety of criteria (reference number, text, technical committee,...).

It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available on-line and also once a month by email.

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary (IEV) on-line.

Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

A propos de la CEI

La Commission Electrotechnique Internationale (CEI) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications CEI

Le contenu technique des publications de la CEI est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Liens utiles:

Recherche de publications CEI - www.iec.ch/searchpub

La recherche avancée vous permet de trouver des publications CEI en utilisant différents critères (numéro de référence, texte, comité d'études,...). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

Just Published CEI - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications de la CEI. Just Published détaille les nouvelles publications parues. Disponible en ligne et aussi une fois par mois par email.

Electropedia - www.electropedia.org

Le premier dictionnaire en ligne au monde de termes électroniques et électriques. Il contient plus de 30 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans les langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (VEI) en ligne.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: csc@iec.ch.

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Energy management system application program interface (EMS-API) –
Part 301: Common information model (CIM) base**

**Interface de programmation d'application pour système de gestion d'énergie
(EMS-API) –
Partie 301: Base de modèle d'information commun (CIM)**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE
CODE PRIX

XH

ICS 33.200

ISBN 978-2-8322-1291-2

**Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.**

CONTENTS

FOREWORD.....	27
INTRODUCTION.....	29
1 Scope.....	31
2 Normative references	31
3 Terms and definitions	32
4 CIM specification	33
4.1 CIM modeling notation	33
4.2 CIM packages.....	33
4.3 CIM classes and relationships.....	35
4.3.1 Classes	35
4.3.2 Generalization	36
4.3.3 Simple association.....	37
4.3.4 Aggregation	37
4.4 CIM model concepts and examples	38
4.4.1 Concepts	38
4.4.2 Containment, equipment hierarchies and naming.....	38
4.4.3 Names model	40
4.4.4 Connectivity model	41
4.4.5 Inheritance hierarchy.....	44
4.4.6 Transformer model	45
4.4.7 Transformer tap modeling.....	48
4.4.8 Phase wire modeling	52
4.4.9 Cuts, clamps and jumpers model	53
4.4.10 Measurements and controls	56
4.4.11 Regulating control models	61
4.5 Modeling guidelines	62
4.5.1 Modeling for change	62
4.5.2 Process for amendments to the CIM	62
4.5.3 Changes to the CIM UML model	63
4.5.4 Changes to the CIM standards documents.....	63
4.5.5 Deprecations	63
4.5.6 CIM profiles	63
4.6 Modeling tools	64
4.7 User implementation conventions.....	64
4.7.1 Conventions beyond UML	64
4.7.2 Number of Terminals for ConductingEquipment objects	64
4.8 CIM modeling examples	64
5 Detailed model	65
5.1 Overview.....	65
5.2 Context	65
6 Top package IEC61970	67
6.1 General.....	67
6.2 IEC61970CIMVersion root class.....	67
6.3 Package Domain.....	68
6.3.1 General	68
6.3.2 ActivePower datatype	73

6.3.3	ActivePowerChangeRate datatype	73
6.3.4	Admittance datatype	73
6.3.5	AngleDegrees datatype	73
6.3.6	AngleRadians datatype	74
6.3.7	ApparentPower datatype	74
6.3.8	Boolean primitive	74
6.3.9	Capacitance datatype	74
6.3.10	Conductance datatype	75
6.3.11	ConductancePerLength datatype	75
6.3.12	CostPerEnergyUnit datatype	75
6.3.13	CostPerVolume datatype	75
6.3.14	CostRate datatype	76
6.3.15	Currency enumeration	76
6.3.16	CurrentFlow datatype	77
6.3.17	Damping datatype	77
6.3.18	Date primitive	78
6.3.19	DateInterval compound	78
6.3.20	DateTime primitive	78
6.3.21	DateTimeInterval compound	78
6.3.22	Decimal primitive	78
6.3.23	Displacement datatype	78
6.3.24	Duration primitive	79
6.3.25	Float primitive	79
6.3.26	FloatQuantity datatype	79
6.3.27	Frequency datatype	79
6.3.28	Hours datatype	79
6.3.29	Impedance datatype	80
6.3.30	Inductance datatype	80
6.3.31	Integer primitive	80
6.3.32	IntegerQuantity datatype	80
6.3.33	KiloActivePower datatype	80
6.3.34	Length datatype	81
6.3.35	Minutes datatype	81
6.3.36	Money datatype	81
6.3.37	PU datatype	81
6.3.38	PerCent datatype	82
6.3.39	Pressure datatype	82
6.3.40	Reactance datatype	82
6.3.41	ReactancePerLength datatype	83
6.3.42	ReactivePower datatype	83
6.3.43	RealEnergy datatype	83
6.3.44	Resistance datatype	83
6.3.45	ResistancePerLength datatype	84
6.3.46	RotationSpeed datatype	84
6.3.47	Seconds datatype	84
6.3.48	Speed datatype	85
6.3.49	String primitive	85
6.3.50	StringQuantity datatype	85
6.3.51	Susceptance datatype	85

6.3.52	SusceptancePerLength datatype	86
6.3.53	Temperature datatype.....	86
6.3.54	Time primitive	86
6.3.55	TimeInterval compound	86
6.3.56	UnitMultiplier enumeration	87
6.3.57	UnitSymbol enumeration	87
6.3.58	Voltage datatype.....	88
6.3.59	VoltagePerReactivePower datatype	88
6.3.60	Volume datatype.....	89
6.3.61	VolumeFlowRate datatype	89
6.3.62	WaterLevel datatype	89
6.3.63	Weight datatype.....	89
6.4	Package Core	90
6.4.1	General	90
6.4.2	BasePower	94
6.4.3	BaseVoltage	94
6.4.4	BasicIntervalSchedule	95
6.4.5	Bay.....	96
6.4.6	BreakerConfiguration enumeration.....	97
6.4.7	BusbarConfiguration enumeration.....	97
6.4.8	ConductingEquipment.....	97
6.4.9	ConnectivityNode.....	98
6.4.10	ConnectivityNodeContainer	99
6.4.11	Curve	99
6.4.12	CurveData root class	100
6.4.13	CurveStyle enumeration	101
6.4.14	Equipment	101
6.4.15	EquipmentContainer	102
6.4.16	GeographicalRegion	103
6.4.17	IdentifiedObject root class	103
6.4.18	IrregularIntervalSchedule.....	104
6.4.19	IrregularTimePoint root class	105
6.4.20	Name root class.....	105
6.4.21	NameType root class	106
6.4.22	NameTypeAuthority root class	106
6.4.23	OperatingParticipant.....	107
6.4.24	OperatingShare root class	107
6.4.25	PhaseCode enumeration.....	108
6.4.26	PowerSystemResource.....	109
6.4.27	PsrList.....	109
6.4.28	PSRType	110
6.4.29	RegularIntervalSchedule.....	110
6.4.30	RegularTimePoint root class	111
6.4.31	ReportingGroup	112
6.4.32	ReportingSuperGroup	112
6.4.33	SubGeographicalRegion	113
6.4.34	Substation	113
6.4.35	Terminal	114
6.4.36	VoltageLevel.....	115

6.5	Package DiagramLayout	116
6.5.1	General	116
6.5.2	Diagram	117
6.5.3	DiagramObject	118
6.5.4	DiagramObjectGluePoint root class	119
6.5.5	DiagramObjectPoint root class	119
6.5.6	DiagramObjectStyle	120
6.5.7	OrientationKind enumeration	120
6.5.8	TextDiagramObject	120
6.5.9	VisibilityLayer	121
6.6	Package OperationalLimits	122
6.6.1	General	122
6.6.2	ActivePowerLimit	123
6.6.3	ApparentPowerLimit	124
6.6.4	BranchGroup	125
6.6.5	BranchGroupTerminal root class	125
6.6.6	CurrentLimit	126
6.6.7	OperationalLimit	126
6.6.8	OperationalLimitDirectionKind enumeration	127
6.6.9	OperationalLimitSet	127
6.6.10	OperationalLimitType	128
6.6.11	VoltageLimit	128
6.7	Package Topology	129
6.7.1	General	129
6.7.2	BusNameMarker	133
6.7.3	TopologicalIsland	134
6.7.4	TopologicalNode	135
6.8	Package Wires	136
6.8.1	General	136
6.8.2	ACLineSegment	149
6.8.3	ACLineSegmentPhase	151
6.8.4	AsynchronousMachine	152
6.8.5	Breaker	154
6.8.6	BusbarSection	155
6.8.7	Clamp	156
6.8.8	CompositeSwitch	157
6.8.9	CompositeSwitchType datatype	158
6.8.10	Conductor	158
6.8.11	Connector	159
6.8.12	CoolantType enumeration	160
6.8.13	Cut	160
6.8.14	DCLineSegment	162
6.8.15	Disconnecter	163
6.8.16	EnergyConsumer	164
6.8.17	EnergyConsumerPhase	166
6.8.18	EnergySource	166
6.8.19	FrequencyConverter	167
6.8.20	Fuse	169
6.8.21	Ground	170

6.8.22	GroundDisconnector	171
6.8.23	Jumper	172
6.8.24	Junction	173
6.8.25	Line	174
6.8.26	LoadBreakSwitch	175
6.8.27	MutualCoupling	176
6.8.28	OperatingMode datatype	177
6.8.29	PerLengthImpedance	177
6.8.30	PerLengthPhaseImpedance	178
6.8.31	PerLengthSequenceImpedance	179
6.8.32	PhaseImpedanceData root class	179
6.8.33	PhaseShuntConnectionKind enumeration	180
6.8.34	PhaseTapChanger	180
6.8.35	PhaseTapChangerAsymetrical	181
6.8.36	PhaseTapChangerLinear	183
6.8.37	PhaseTapChangerNonLinear	184
6.8.38	PhaseTapChangerSymetrical	185
6.8.39	PhaseTapChangerTabular	186
6.8.40	PhaseTapChangerTabularPoint root class	186
6.8.41	Plant	187
6.8.42	PowerTransformer	188
6.8.43	PowerTransformerEnd	190
6.8.44	ProtectedSwitch	192
6.8.45	RatioTapChanger	193
6.8.46	RatioTapChangerTabular	194
6.8.47	RatioTapChangerTabularPoint root class	195
6.8.48	ReactiveCapabilityCurve	196
6.8.49	Recloser	197
6.8.50	RectifierInverter	198
6.8.51	RegulatingCondEq	199
6.8.52	RegulatingControl	200
6.8.53	RegulatingControlModeKind enumeration	201
6.8.54	RegulationSchedule	202
6.8.55	Resistor	203
6.8.56	RotatingMachine	203
6.8.57	Sectionaliser	205
6.8.58	SeriesCompensator	206
6.8.59	ShuntCompensator	207
6.8.60	ShuntCompensatorPhase	209
6.8.61	SinglePhaseKind enumeration	210
6.8.62	StaticVarCompensator	210
6.8.63	SVCControlMode enumeration	211
6.8.64	Switch	212
6.8.65	SwitchPhase	213
6.8.66	SwitchSchedule	214
6.8.67	SynchronousGeneratorType enumeration	215
6.8.68	SynchronousMachine	215
6.8.69	SynchronousMachineOperatingMode enumeration	218
6.8.70	SynchronousMachineType enumeration	219

6.8.71	TapChanger	219
6.8.72	TapChangerControl	220
6.8.73	TapChangerKind enumeration	221
6.8.74	TapSchedule	221
6.8.75	TransformerControlMode enumeration	222
6.8.76	TransformerCoreAdmittance	222
6.8.77	TransformerEnd	223
6.8.78	TransformerMeshImpedance	225
6.8.79	TransformerStarImpedance	225
6.8.80	TransformerTank	226
6.8.81	TransformerTankEnd	227
6.8.82	VoltageControlZone	228
6.8.83	WindingConnection enumeration	229
6.9	Package Generation	229
6.9.1	General	229
6.9.2	Package GenerationDynamics	230
6.9.3	Package Production	246
6.10	Package LoadModel	294
6.10.1	General	294
6.10.2	ConformLoad	295
6.10.3	ConformLoadGroup	296
6.10.4	ConformLoadSchedule	297
6.10.5	DayType	298
6.10.6	EnergyArea	298
6.10.7	LoadArea	299
6.10.8	LoadGroup	299
6.10.9	LoadResponseCharacteristic	300
6.10.10	NonConformLoad	301
6.10.11	NonConformLoadGroup	302
6.10.12	NonConformLoadSchedule	303
6.10.13	PowerCutZone	304
6.10.14	Season root class	304
6.10.15	SeasonDayTypeSchedule	305
6.10.16	SeasonName enumeration	305
6.10.17	StationSupply	306
6.10.18	SubLoadArea	307
6.11	Package Outage	308
6.11.1	General	308
6.11.2	ClearanceTag	309
6.11.3	ClearanceTagType	310
6.11.4	OutageSchedule	311
6.11.5	SwitchingOperation	311
6.11.6	SwitchState enumeration	312
6.12	Package AuxiliaryEquipment	312
6.12.1	General	312
6.12.2	AuxiliaryEquipment	313
6.12.3	CurrentTransformer	314
6.12.4	FaultIndicator	315
6.12.5	PostLineSensor	316

6.12.6	PotentialTransformer	317
6.12.7	Sensor	318
6.12.8	SurgeProtector	319
6.13	Package Protection	320
6.13.1	General	320
6.13.2	CurrentRelay	321
6.13.3	ProtectionEquipment	322
6.13.4	RecloseSequence	323
6.13.5	SynchrocheckRelay	324
6.14	Package Equivalents	325
6.14.1	General	325
6.14.2	EquivalentBranch	325
6.14.3	EquivalentEquipment	326
6.14.4	EquivalentInjection	327
6.14.5	EquivalentNetwork	328
6.14.6	EquivalentShunt	329
6.15	Package Meas	330
6.15.1	General	330
6.15.2	Accumulator	333
6.15.3	AccumulatorLimit	334
6.15.4	AccumulatorLimitSet	335
6.15.5	AccumulatorValue	335
6.15.6	Analog	336
6.15.7	AnalogLimit	337
6.15.8	AnalogLimitSet	337
6.15.9	AnalogValue	338
6.15.10	Command	339
6.15.11	Control	340
6.15.12	ControlType	341
6.15.13	Discrete	341
6.15.14	DiscreteValue	342
6.15.15	Limit	343
6.15.16	LimitSet	343
6.15.17	Measurement	344
6.15.18	MeasurementValue	345
6.15.19	MeasurementValueQuality	346
6.15.20	MeasurementValueSource	347
6.15.21	Quality61850 root class	348
6.15.22	SetPoint	348
6.15.23	StringMeasurement	349
6.15.24	StringMeasurementValue	350
6.15.25	Validity enumeration	350
6.15.26	ValueAliasSet	351
6.15.27	ValueToAlias	352
6.16	Package SCADA	352
6.16.1	General	352
6.16.2	CommunicationLink	353
6.16.3	RemoteControl	354
6.16.4	RemotePoint	354

6.16.5	RemoteSource	355
6.16.6	RemoteUnit	356
6.16.7	RemoteUnitType enumeration	357
6.16.8	Source enumeration	357
6.17	Package ControlArea	357
6.17.1	General	357
6.17.2	AltGeneratingUnitMeas root class	359
6.17.3	AltTieMeas root class	360
6.17.4	ControlArea	360
6.17.5	ControlAreaGeneratingUnit root class	361
6.17.6	ControlAreaTypeKind enumeration	362
6.17.7	TieFlow root class	362
6.18	Package Contingency	363
6.18.1	General	363
6.18.2	Contingency	363
6.18.3	ContingencyElement	364
6.18.4	ContingencyEquipment	364
6.18.5	ContingencyEquipmentStatusKind enumeration	365
6.19	Package StateVariables	365
6.19.1	General	365
6.19.2	StateVariable root class	366
6.19.3	SvInjection	366
6.19.4	SvPowerFlow	367
6.19.5	SvShortCircuit	367
6.19.6	SvShuntCompensatorSections	368
6.19.7	SvStatus	368
6.19.8	SvTapStep	369
6.19.9	SvVoltage	369
	Bibliography	371
	Figure 1 – Example of working group package dependencies	34
	Figure 2 – CIM IEC 61970-301 package diagram	35
	Figure 3 – Example of generalization	37
	Figure 4 – Example of simple association	37
	Figure 5 – Example of aggregation	38
	Figure 6 – Equipment containers	39
	Figure 7 – Names	40
	Figure 8 – Connectivity model	41
	Figure 9 – Simple network example	43
	Figure 10 – Simple network connectivity modeled with CIM Topology	44
	Figure 11 – Equipment inheritance hierarchy	45
	Figure 12 – Transformer and Tank model	46
	Figure 13 – Transformer model	47
	Figure 14 – Transformer tap model	48
	Figure 15 – Symmetrical Phase Shifter	50
	Figure 16 – Asymmetrical Phase Shifter	51

Figure 17 – Phase connectivity	52
Figure 18 – Cuts, clamps, and jumpers UML model	53
Figure 19 – Example before cuts and jumpers applied	54
Figure 20 – Example after cuts and jumpers applied	55
Figure 21 – Example of jumper without cut or clamp	56
Figure 22 – Navigating from PSR to MeasurementValue	58
Figure 23 – Measurement placement	61
Figure 24 – Regulating control models	62
Figure 25 – CIM top level packages	65
Figure 26 – Class diagram IEC61970::IEC61970Dependencies	67
Figure 27 – Class diagram Domain::BasicDatatypes	68
Figure 28 – Class diagram Domain::ElectricityDatatypes	69
Figure 29 – Class diagram Domain::EnumeratedUnitDatatypes	70
Figure 30 – Class diagram Domain::GeneralDatatypes	71
Figure 31 – Class diagram Domain::MonetaryDatatypes	72
Figure 32 – Class diagram Domain::TimeDatatypes	72
Figure 33 – Class diagram Core::Main	91
Figure 34 – Class diagram Core::Names	92
Figure 35 – Class diagram Core::CurveSchedule	92
Figure 36 – Class diagram Core::Datatypes	93
Figure 37 – Class diagram Core::Reporting	93
Figure 38 – Class diagram Core::OperatingShare	94
Figure 39 – Class diagram DiagramLayout::DiagramLayout	117
Figure 40 – Class diagram OperationalLimits::OperationalLimits	123
Figure 41 – Class diagram OperationalLimits::BranchGroup	123
Figure 42 – Class diagram Topology::TopologicalNodeTerminal	130
Figure 43 – Class diagram Topology::TopologyMeasRelations	131
Figure 44 – Class diagram Topology::TopologyReporting	132
Figure 45 – Class diagram Topology::Main	133
Figure 46 – Class diagram Wires::CutsAndJumpers	136
Figure 47 – Class diagram Wires::MutualCoupling	137
Figure 48 – Class diagram Wires::Schedules	138
Figure 49 – Class diagram Wires::SwitchingEquipment	139
Figure 50 – Class diagram Wires::WiresPhaseModel	140
Figure 51 – Class diagram Wires::Datatypes	141
Figure 52 – Class diagram Wires::InheritanceHierarchy	142
Figure 53 – Class diagram Wires::LineModel	143
Figure 54 – Class diagram Wires::NamingHierarchyPart1	144
Figure 55 – Class diagram Wires::NamingHierarchyPart2	145
Figure 56 – Class diagram Wires::RegulatingEquipment	146
Figure 57 – Class diagram Wires::TapChanger	147
Figure 58 – Class diagram Wires::VoltageControl	148
Figure 59 – Class diagram Wires::Transformer	149

Figure 60 – Class diagram Generation::Main	230
Figure 61 – Class diagram GenerationDynamics::Main	230
Figure 62 – Class diagram GenerationDynamics::Datatypes	231
Figure 63 – Class diagram Production::Nuclear	247
Figure 64 – Class diagram Production::Main	248
Figure 65 – Class diagram Production::Datatypes	249
Figure 66 – Class diagram Production::Hydro	250
Figure 67 – Class diagram Production::Thermal	251
Figure 68 – Class diagram LoadModel::Main	294
Figure 69 – Class diagram LoadModel::Datatypes	295
Figure 70 – Class diagram Outage::Datatypes	308
Figure 71 – Class diagram Outage::Main	309
Figure 72 – Class diagram AuxiliaryEquipment::AuxiliaryEquipment	313
Figure 73 – Class diagram Protection::Main	320
Figure 74 – Class diagram Equivalent::Main	325
Figure 75 – Class diagram Meas::Datatypes	330
Figure 76 – Class diagram Meas::Control	331
Figure 77 – Class diagram Meas::Measurement	332
Figure 78 – Class diagram Meas::MeasurementInheritance	333
Figure 79 – Class diagram Meas::Quality	333
Figure 80 – Class diagram SCADA::Datatypes	352
Figure 81 – Class diagram SCADA::Main	353
Figure 82 – Class diagram ControlArea::ControlArea	358
Figure 83 – Class diagram ControlArea::ControlAreaInheritance	359
Figure 84 – Class diagram ControlArea::Datatypes	359
Figure 85 – Class diagram Contingency::Contingency	363
Figure 86 – Class diagram StateVariables::StateVariables	366
Table 1 – NameType class naming conventions	40
Table 2 – measurementType naming conventions	59
Table 3 – MeasurementValueSource naming conventions	60
Table 4 – Attributes of Package1::Class1	66
Table 5 – Association ends of Package1::Class1 with other classes	66
Table 6 – Literals of Package1::Enumeration1	66
Table 7 – Attributes of IEC61970::IEC61970CIMVersion	68
Table 8 – Attributes of Domain::ActivePower	73
Table 9 – Attributes of Domain::ActivePowerChangeRate	73
Table 10 – Attributes of Domain::Admittance	73
Table 11 – Attributes of Domain::AngleDegrees	74
Table 12 – Attributes of Domain::AngleRadians	74
Table 13 – Attributes of Domain::ApparentPower	74
Table 14 – Attributes of Domain::Capacitance	74
Table 15 – Attributes of Domain::Conductance	75

Table 16 – Attributes of Domain::ConductancePerLength	75
Table 17 – Attributes of Domain::CostPerEnergyUnit	75
Table 18 – Attributes of Domain::CostPerVolume	76
Table 19 – Attributes of Domain::CostRate	76
Table 20 – Literals of Domain::Currency	77
Table 21 – Attributes of Domain::CurrentFlow	77
Table 22 – Attributes of Domain::Damping	77
Table 23 – Attributes of Domain::DateInterval	78
Table 24 – Attributes of Domain::DateTimeInterval	78
Table 25 – Attributes of Domain::Displacement	78
Table 26 – Attributes of Domain::FloatQuantity	79
Table 27 – Attributes of Domain::Frequency	79
Table 28 – Attributes of Domain::Hours	79
Table 29 – Attributes of Domain::Impedance	80
Table 30 – Attributes of Domain::Inductance	80
Table 31 – Attributes of Domain::IntegerQuantity	80
Table 32 – Attributes of Domain::KiloActivePower	81
Table 33 – Attributes of Domain::Length	81
Table 34 – Attributes of Domain::Minutes	81
Table 35 – Attributes of Domain::Money	81
Table 36 – Attributes of Domain::PU	82
Table 37 – Attributes of Domain::PerCent	82
Table 38 – Attributes of Domain::Pressure	82
Table 39 – Attributes of Domain::Reactance	82
Table 40 – Attributes of Domain::ReactancePerLength	83
Table 41 – Attributes of Domain::ReactivePower	83
Table 42 – Attributes of Domain::RealEnergy	83
Table 43 – Attributes of Domain::Resistance	84
Table 44 – Attributes of Domain::ResistancePerLength	84
Table 45 – Attributes of Domain::RotationSpeed	84
Table 46 – Attributes of Domain::Seconds	85
Table 47 – Attributes of Domain::Speed	85
Table 48 – Attributes of Domain::StringQuantity	85
Table 49 – Attributes of Domain::Susceptance	86
Table 50 – Attributes of Domain::SusceptancePerLength	86
Table 51 – Attributes of Domain::Temperature	86
Table 52 – Attributes of Domain::TimeInterval	87
Table 53 – Literals of Domain::UnitMultiplier	87
Table 54 – Literals of Domain::UnitSymbol	87
Table 55 – Attributes of Domain::Voltage	88
Table 56 – Attributes of Domain::VoltagePerReactivePower	88
Table 57 – Attributes of Domain::Volume	89
Table 58 – Attributes of Domain::VolumeFlowRate	89

Table 59 – Attributes of Domain::WaterLevel	89
Table 60 – Attributes of Domain::Weight	90
Table 61 – Attributes of Core::BasePower	94
Table 62 – Association ends of Core::BasePower with other classes	94
Table 63 – Attributes of Core::BaseVoltage	95
Table 64 – Association ends of Core::BaseVoltage with other classes	95
Table 65 – Attributes of Core::BasicIntervalSchedule	95
Table 66 – Association ends of Core::BasicIntervalSchedule with other classes	96
Table 67 – Attributes of Core::Bay	96
Table 68 – Association ends of Core::Bay with other classes	96
Table 69 – Literals of Core::BreakerConfiguration	97
Table 70 – Literals of Core::BusbarConfiguration	97
Table 71 – Attributes of Core::ConductingEquipment	97
Table 72 – Association ends of Core::ConductingEquipment with other classes	98
Table 73 – Attributes of Core::ConnectivityNode	98
Table 74 – Association ends of Core::ConnectivityNode with other classes	99
Table 75 – Attributes of Core::ConnectivityNodeContainer	99
Table 76 – Association ends of Core::ConnectivityNodeContainer with other classes	99
Table 77 – Attributes of Core::Curve	100
Table 78 – Association ends of Core::Curve with other classes	100
Table 79 – Attributes of Core::CurveData	100
Table 80 – Association ends of Core::CurveData with other classes	101
Table 81 – Literals of Core::CurveStyle	101
Table 82 – Attributes of Core::Equipment	101
Table 83 – Association ends of Core::Equipment with other classes	102
Table 84 – Attributes of Core::EquipmentContainer	102
Table 85 – Association ends of Core::EquipmentContainer with other classes	103
Table 86 – Attributes of Core::GeographicalRegion	103
Table 87 – Association ends of Core::GeographicalRegion with other classes	103
Table 88 – Attributes of Core::IdentifiedObject	104
Table 89 – Association ends of Core::IdentifiedObject with other classes	104
Table 90 – Attributes of Core::IrregularIntervalSchedule	104
Table 91 – Association ends of Core::IrregularIntervalSchedule with other classes	105
Table 92 – Attributes of Core::IrregularTimePoint	105
Table 93 – Association ends of Core::IrregularTimePoint with other classes	105
Table 94 – Attributes of Core::Name	105
Table 95 – Association ends of Core::Name with other classes	106
Table 96 – Attributes of Core::NameType	106
Table 97 – Association ends of Core::NameType with other classes	106
Table 98 – Attributes of Core::NameTypeAuthority	106
Table 99 – Association ends of Core::NameTypeAuthority with other classes	107
Table 100 – Attributes of Core::OperatingParticipant	107
Table 101 – Association ends of Core::OperatingParticipant with other classes	107

Table 102 – Attributes of Core::OperatingShare	107
Table 103 – Association ends of Core::OperatingShare with other classes	108
Table 104 – Literals of Core::PhaseCode	108
Table 105 – Attributes of Core::PowerSystemResource	109
Table 106 – Association ends of Core::PowerSystemResource with other classes	109
Table 107 – Attributes of Core::PsrList	109
Table 108 – Association ends of Core::PsrList with other classes	110
Table 109 – Attributes of Core::PSRType	110
Table 110 – Association ends of Core::PSRType with other classes	110
Table 111 – Attributes of Core::RegularIntervalSchedule	111
Table 112 – Association ends of Core::RegularIntervalSchedule with other classes	111
Table 113 – Attributes of Core::RegularTimePoint	111
Table 114 – Association ends of Core::RegularTimePoint with other classes	112
Table 115 – Attributes of Core::ReportingGroup	112
Table 116 – Association ends of Core::ReportingGroup with other classes	112
Table 117 – Attributes of Core::ReportingSuperGroup	112
Table 118 – Association ends of Core::ReportingSuperGroup with other classes	113
Table 119 – Attributes of Core::SubGeographicalRegion	113
Table 120 – Association ends of Core::SubGeographicalRegion with other classes	113
Table 121 – Attributes of Core::Substation	113
Table 122 – Association ends of Core::Substation with other classes	114
Table 123 – Attributes of Core::Terminal	114
Table 124 – Association ends of Core::Terminal with other classes	115
Table 125 – Attributes of Core::VoltageLevel	116
Table 126 – Association ends of Core::VoltageLevel with other classes	116
Table 127 – Attributes of DiagramLayout::Diagram	117
Table 128 – Association ends of DiagramLayout::Diagram with other classes	118
Table 129 – Attributes of DiagramLayout::DiagramObject	118
Table 130 – Association ends of DiagramLayout::DiagramObject with other classes	119
Table 131 – Association ends of DiagramLayout::DiagramObjectGluePoint with other classes	119
Table 132 – Attributes of DiagramLayout::DiagramObjectPoint	119
Table 133 – Association ends of DiagramLayout::DiagramObjectPoint with other classes	120
Table 134 – Attributes of DiagramLayout::DiagramObjectStyle	120
Table 135 – Association ends of DiagramLayout::DiagramObjectStyle with other classes	120
Table 136 – Literals of DiagramLayout::OrientationKind	120
Table 137 – Attributes of DiagramLayout::TextDiagramObject	121
Table 138 – Association ends of DiagramLayout::TextDiagramObject with other classes	121
Table 139 – Attributes of DiagramLayout::VisibilityLayer	122
Table 140 – Association ends of DiagramLayout::VisibilityLayer with other classes	122
Table 141 – Attributes of OperationalLimits::ActivePowerLimit	124

Table 142 – Association ends of OperationalLimits::ActivePowerLimit with other classes	124
Table 143 – Attributes of OperationalLimits::ApparentPowerLimit	124
Table 144 – Association ends of OperationalLimits::ApparentPowerLimit with other classes	124
Table 145 – Attributes of OperationalLimits::BranchGroup	125
Table 146 – Association ends of OperationalLimits::BranchGroup with other classes	125
Table 147 – Attributes of OperationalLimits::BranchGroupTerminal	125
Table 148 – Association ends of OperationalLimits::BranchGroupTerminal with other classes	126
Table 149 – Attributes of OperationalLimits::CurrentLimit	126
Table 150 – Association ends of OperationalLimits::CurrentLimit with other classes	126
Table 151 – Attributes of OperationalLimits::OperationalLimit	126
Table 152 – Association ends of OperationalLimits::OperationalLimit with other classes	127
Table 153 – Literals of OperationalLimits::OperationalLimitDirectionKind	127
Table 154 – Attributes of OperationalLimits::OperationalLimitSet	127
Table 155 – Association ends of OperationalLimits::OperationalLimitSet with other classes	128
Table 156 – Attributes of OperationalLimits::OperationalLimitType	128
Table 157 – Association ends of OperationalLimits::OperationalLimitType with other classes	128
Table 158 – Attributes of OperationalLimits::VoltageLimit	129
Table 159 – Association ends of OperationalLimits::VoltageLimit with other classes	129
Table 160 – Attributes of Topology::BusNameMarker	134
Table 161 – Association ends of Topology::BusNameMarker with other classes	134
Table 162 – Attributes of Topology::TopologicalIsland	134
Table 163 – Association ends of Topology::TopologicalIsland with other classes	134
Table 164 – Attributes of Topology::TopologicalNode	135
Table 165 – Association ends of Topology::TopologicalNode with other classes	135
Table 166 – Attributes of Wires::ACLineSegment	150
Table 167 – Association ends of Wires::ACLineSegment with other classes	151
Table 168 – Attributes of Wires::ACLineSegmentPhase	151
Table 169 – Association ends of Wires::ACLineSegmentPhase with other classes	152
Table 170 – Attributes of Wires::AsynchronousMachine	153
Table 171 – Association ends of Wires::AsynchronousMachine with other classes	154
Table 172 – Attributes of Wires::Breaker	154
Table 173 – Association ends of Wires::Breaker with other classes	155
Table 174 – Attributes of Wires::BusbarSection	155
Table 175 – Association ends of Wires::BusbarSection with other classes	156
Table 176 – Attributes of Wires::Clamp	156
Table 177 – Association ends of Wires::Clamp with other classes	157
Table 178 – Attributes of Wires::CompositeSwitch	157
Table 179 – Association ends of Wires::CompositeSwitch with other classes	158
Table 180 – Attributes of Wires::CompositeSwitchType	158
Table 181 – Attributes of Wires::Conductor	158

Table 182 – Association ends of Wires::Conductor with other classes.....	159
Table 183 – Attributes of Wires::Connector.....	159
Table 184 – Association ends of Wires::Connector with other classes.....	160
Table 185 – Literals of Wires::CoolantType.....	160
Table 186 – Attributes of Wires::Cut	161
Table 187 – Association ends of Wires::Cut with other classes	161
Table 188 – Attributes of Wires::DCLineSegment.....	162
Table 189 – Association ends of Wires::DCLineSegment with other classes	162
Table 190 – Attributes of Wires::Disconnecter.....	163
Table 191 – Association ends of Wires::Disconnecter with other classes	164
Table 192 – Attributes of Wires::EnergyConsumer	165
Table 193 – Association ends of Wires::EnergyConsumer with other classes.....	165
Table 194 – Attributes of Wires::EnergyConsumerPhase	166
Table 195 – Association ends of Wires::EnergyConsumerPhase with other classes	166
Table 196 – Attributes of Wires::EnergySource	167
Table 197 – Association ends of Wires::EnergySource with other classes.....	167
Table 198 – Attributes of Wires::FrequencyConverter.....	168
Table 199 – Association ends of Wires::FrequencyConverter with other classes	168
Table 200 – Attributes of Wires::Fuse	169
Table 201 – Association ends of Wires::Fuse with other classes	170
Table 202 – Attributes of Wires::Ground	170
Table 203 – Association ends of Wires::Ground with other classes	171
Table 204 – Attributes of Wires::GroundDisconnecter	171
Table 205 – Association ends of Wires::GroundDisconnecter with other classes.....	172
Table 206 – Attributes of Wires::Jumper	172
Table 207 – Association ends of Wires::Jumper with other classes	173
Table 208 – Attributes of Wires::Junction.....	173
Table 209 – Association ends of Wires::Junction with other classes.....	174
Table 210 – Attributes of Wires::Line	174
Table 211 – Association ends of Wires::Line with other classes	175
Table 212 – Attributes of Wires::LoadBreakSwitch	175
Table 213 – Association ends of Wires::LoadBreakSwitch with other classes.....	176
Table 214 – Attributes of Wires::MutualCoupling.....	177
Table 215 – Association ends of Wires::MutualCoupling with other classes	177
Table 216 – Attributes of Wires::OperatingMode	177
Table 217 – Attributes of Wires::PerLengthImpedance	178
Table 218 – Association ends of Wires::PerLengthImpedance with other classes.....	178
Table 219 – Attributes of Wires::PerLengthPhaseImpedance	178
Table 220 – Association ends of Wires::PerLengthPhaseImpedance with other classes	178
Table 221 – Attributes of Wires::PerLengthSequenceImpedance	179
Table 222 – Association ends of Wires::PerLengthSequenceImpedance with other classes	179
Table 223 – Attributes of Wires::PhaseImpedanceData.....	180

Table 224 – Association ends of Wires::PhaseImpedanceData with other classes	180
Table 225 – Literals of Wires::PhaseShuntConnectionKind	180
Table 226 – Attributes of Wires::PhaseTapChanger	181
Table 227 – Association ends of Wires::PhaseTapChanger with other classes	181
Table 228 – Attributes of Wires::PhaseTapChangerAsymetrical	182
Table 229 – Association ends of Wires::PhaseTapChangerAsymetrical with other classes	182
Table 230 – Attributes of Wires::PhaseTapChangerLinear	183
Table 231 – Association ends of Wires::PhaseTapChangerLinear with other classes	183
Table 232 – Attributes of Wires::PhaseTapChangerNonLinear	184
Table 233 – Association ends of Wires::PhaseTapChangerNonLinear with other classes	184
Table 234 – Attributes of Wires::PhaseTapChangerSymetrical	185
Table 235 – Association ends of Wires::PhaseTapChangerSymetrical with other classes	186
Table 236 – Attributes of Wires::PhaseTapChangerTabular	186
Table 237 – Association ends of Wires::PhaseTapChangerTabular with other classes	186
Table 238 – Attributes of Wires::PhaseTapChangerTabularPoint	187
Table 239 – Association ends of Wires::PhaseTapChangerTabularPoint with other classes	187
Table 240 – Attributes of Wires::Plant	187
Table 241 – Association ends of Wires::Plant with other classes	188
Table 242 – Attributes of Wires::PowerTransformer	189
Table 243 – Association ends of Wires::PowerTransformer with other classes	190
Table 244 – Attributes of Wires::PowerTransformerEnd	191
Table 245 – Association ends of Wires::PowerTransformerEnd with other classes	192
Table 246 – Attributes of Wires::ProtectedSwitch	192
Table 247 – Association ends of Wires::ProtectedSwitch with other classes	193
Table 248 – Attributes of Wires::RatioTapChanger	194
Table 249 – Association ends of Wires::RatioTapChanger with other classes	194
Table 250 – Attributes of Wires::RatioTapChangerTabular	195
Table 251 – Association ends of Wires::RatioTapChangerTabular with other classes	195
Table 252 – Attributes of Wires::RatioTapChangerTabularPoint	195
Table 253 – Association ends of Wires::RatioTapChangerTabularPoint with other classes	196
Table 254 – Attributes of Wires::ReactiveCapabilityCurve	196
Table 255 – Association ends of Wires::ReactiveCapabilityCurve with other classes	196
Table 256 – Attributes of Wires::Recloser	197
Table 257 – Association ends of Wires::Recloser with other classes	197
Table 258 – Attributes of Wires::RectifierInverter	198
Table 259 – Association ends of Wires::RectifierInverter with other classes	199
Table 260 – Attributes of Wires::RegulatingCondEq	199
Table 261 – Association ends of Wires::RegulatingCondEq with other classes	200
Table 262 – Attributes of Wires::RegulatingControl	201
Table 263 – Association ends of Wires::RegulatingControl with other classes	201

Table 264 – Literals of Wires::RegulatingControlModeKind	202
Table 265 – Attributes of Wires::RegulationSchedule	202
Table 266 – Association ends of Wires::RegulationSchedule with other classes.....	202
Table 267 – Attributes of Wires::Resistor	203
Table 268 – Association ends of Wires::Resistor with other classes.....	203
Table 269 – Attributes of Wires::RotatingMachine	203
Table 270 – Association ends of Wires::RotatingMachine with other classes.....	205
Table 271 – Attributes of Wires::Sectionaliser.....	205
Table 272 – Association ends of Wires::Sectionaliser with other classes.....	206
Table 273 – Attributes of Wires::SeriesCompensator	206
Table 274 – Association ends of Wires::SeriesCompensator with other classes.....	207
Table 275 – Attributes of Wires::ShuntCompensator	208
Table 276 – Association ends of Wires::ShuntCompensator with other classes.....	209
Table 277 – Attributes of Wires::ShuntCompensatorPhase	209
Table 278 – Association ends of Wires::ShuntCompensatorPhase with other classes	210
Table 279 – Literals of Wires::SinglePhaseKind	210
Table 280 – Attributes of Wires::StaticVarCompensator	211
Table 281 – Association ends of Wires::StaticVarCompensator with other classes.....	211
Table 282 – Literals of Wires::SVCControlMode.....	212
Table 283 – Attributes of Wires::Switch.....	212
Table 284 – Association ends of Wires::Switch with other classes	213
Table 285 – Attributes of Wires::SwitchPhase.....	214
Table 286 – Association ends of Wires::SwitchPhase with other classes.....	214
Table 287 – Attributes of Wires::SwitchSchedule	215
Table 288 – Association ends of Wires::SwitchSchedule with other classes	215
Table 289 – Literals of Wires::SynchronousGeneratorType	215
Table 290 – Attributes of Wires::SynchronousMachine.....	216
Table 291 – Association ends of Wires::SynchronousMachine with other classes	218
Table 292 – Literals of Wires::SynchronousMachineOperatingMode	218
Table 293 – Literals of Wires::SynchronousMachineType.....	219
Table 294 – Attributes of Wires::TapChanger.....	219
Table 295 – Association ends of Wires::TapChanger with other classes	220
Table 296 – Attributes of Wires::TapChangerControl	220
Table 297 – Association ends of Wires::TapChangerControl with other classes	221
Table 298 – Literals of Wires::TapChangerKind	221
Table 299 – Attributes of Wires::TapSchedule.....	222
Table 300 – Association ends of Wires::TapSchedule with other classes	222
Table 301 – Literals of Wires::TransformerControlMode.....	222
Table 302 – Attributes of Wires::TransformerCoreAdmittance	223
Table 303 – Association ends of Wires::TransformerCoreAdmittance with other classes	223
Table 304 – Attributes of Wires::TransformerEnd.....	224
Table 305 – Association ends of Wires::TransformerEnd with other classes.....	224
Table 306 – Attributes of Wires::TransformerMeshImpedance.....	225

Table 307 – Association ends of Wires::TransformerMeshImpedance with other classes	225
Table 308 – Attributes of Wires::TransformerStarImpedance.....	226
Table 309 – Association ends of Wires::TransformerStarImpedance with other classes	226
Table 310 – Attributes of Wires::TransformerTank	226
Table 311 – Association ends of Wires::TransformerTank with other classes	227
Table 312 – Attributes of Wires::TransformerTankEnd	227
Table 313 – Association ends of Wires::TransformerTankEnd with other classes	228
Table 314 – Attributes of Wires::VoltageControlZone	228
Table 315 – Association ends of Wires::VoltageControlZone with other classes.....	229
Table 316 – Literals of Wires::WindingConnection	229
Table 317 – Literals of GenerationDynamics::BoilerControlMode	231
Table 318 – Attributes of GenerationDynamics::BWRSteamSupply	231
Table 319 – Association ends of GenerationDynamics::BWRSteamSupply with other classes	232
Table 320 – Attributes of GenerationDynamics::CombustionTurbine.....	232
Table 321 – Association ends of GenerationDynamics::CombustionTurbine with other classes	233
Table 322 – Attributes of GenerationDynamics::CTTempActivePowerCurve.....	234
Table 323 – Association ends of GenerationDynamics::CTTempActivePowerCurve with other classes	234
Table 324 – Attributes of GenerationDynamics::DrumBoiler	234
Table 325 – Association ends of GenerationDynamics::DrumBoiler with other classes.....	235
Table 326 – Attributes of GenerationDynamics::FossilSteamSupply.....	236
Table 327 – Association ends of GenerationDynamics::FossilSteamSupply with other classes	237
Table 328 – Attributes of GenerationDynamics::HeatRecoveryBoiler.....	237
Table 329 – Association ends of GenerationDynamics::HeatRecoveryBoiler with other classes	238
Table 330 – Attributes of GenerationDynamics::HydroTurbine	239
Table 331 – Association ends of GenerationDynamics::HydroTurbine with other classes	239
Table 332 – Attributes of GenerationDynamics::PrimeMover	240
Table 333 – Association ends of GenerationDynamics::PrimeMover with other classes.....	240
Table 334 – Attributes of GenerationDynamics::PWRSteamSupply	240
Table 335 – Association ends of GenerationDynamics::PWRSteamSupply with other classes	241
Table 336 – Attributes of GenerationDynamics::SteamSupply	241
Table 337 – Association ends of GenerationDynamics::SteamSupply with other classes.....	242
Table 338 – Attributes of GenerationDynamics::SteamTurbine.....	242
Table 339 – Association ends of GenerationDynamics::SteamTurbine with other classes	243
Table 340 – Attributes of GenerationDynamics::Subcritical	244
Table 341 – Association ends of GenerationDynamics::Subcritical with other classes	245
Table 342 – Attributes of GenerationDynamics::Supercritical	245
Table 343 – Association ends of GenerationDynamics::Supercritical with other classes.....	246
Table 344 – Literals of GenerationDynamics::TurbineType	246

Table 345 – Attributes of Production::AirCompressor	251
Table 346 – Association ends of Production::AirCompressor with other classes	252
Table 347 – Attributes of Production::CAESPlant	252
Table 348 – Association ends of Production::CAESPlant with other classes	253
Table 349 – Attributes of Production::Classification	253
Table 350 – Attributes of Production::CogenerationPlant	254
Table 351 – Association ends of Production::CogenerationPlant with other classes	254
Table 352 – Attributes of Production::CombinedCyclePlant	254
Table 353 – Association ends of Production::CombinedCyclePlant with other classes	255
Table 354 – Attributes of Production::CostPerHeatUnit	255
Table 355 – Attributes of Production::Emission	255
Table 356 – Attributes of Production::EmissionAccount	256
Table 357 – Association ends of Production::EmissionAccount with other classes	256
Table 358 – Attributes of Production::EmissionCurve	257
Table 359 – Association ends of Production::EmissionCurve with other classes	257
Table 360 – Literals of Production::EmissionType	257
Table 361 – Literals of Production::EmissionValueSource	258
Table 362 – Attributes of Production::FossilFuel	258
Table 363 – Association ends of Production::FossilFuel with other classes	259
Table 364 – Attributes of Production::FuelAllocationSchedule	259
Table 365 – Association ends of Production::FuelAllocationSchedule with other classes	260
Table 366 – Literals of Production::FuelType	260
Table 367 – Attributes of Production::GeneratingUnit	260
Table 368 – Association ends of Production::GeneratingUnit with other classes	263
Table 369 – Literals of Production::GeneratorControlMode	263
Table 370 – Literals of Production::GeneratorControlSource	264
Table 371 – Literals of Production::GeneratorOperatingMode	264
Table 372 – Attributes of Production::GenUnitOpCostCurve	265
Table 373 – Association ends of Production::GenUnitOpCostCurve with other classes	265
Table 374 – Attributes of Production::GenUnitOpSchedule	266
Table 375 – Association ends of Production::GenUnitOpSchedule with other classes	266
Table 376 – Attributes of Production::GrossToNetActivePowerCurve	267
Table 377 – Association ends of Production::GrossToNetActivePowerCurve with other classes	267
Table 378 – Attributes of Production::HeatInputCurve	268
Table 379 – Association ends of Production::HeatInputCurve with other classes	268
Table 380 – Attributes of Production::HeatRate	268
Table 381 – Attributes of Production::HeatRateCurve	269
Table 382 – Association ends of Production::HeatRateCurve with other classes	269
Table 383 – Literals of Production::HydroEnergyConversionKind	270
Table 384 – Attributes of Production::HydroGeneratingEfficiencyCurve	270
Table 385 – Association ends of Production::HydroGeneratingEfficiencyCurve with other classes	270
Table 386 – Attributes of Production::HydroGeneratingUnit	271

Table 387 – Association ends of Production::HydroGeneratingUnit with other classes	272
Table 388 – Literals of Production::HydroPlantType	273
Table 389 – Attributes of Production::HydroPowerPlant	273
Table 390 – Association ends of Production::HydroPowerPlant with other classes	274
Table 391 – Attributes of Production::HydroPump	274
Table 392 – Association ends of Production::HydroPump with other classes	275
Table 393 – Attributes of Production::HydroPumpOpSchedule	275
Table 394 – Association ends of Production::HydroPumpOpSchedule with other classes	276
Table 395 – Attributes of Production::IncrementalHeatRateCurve	276
Table 396 – Association ends of Production::IncrementalHeatRateCurve with other classes	276
Table 397 – Attributes of Production::InflowForecast	277
Table 398 – Association ends of Production::InflowForecast with other classes	277
Table 399 – Attributes of Production::LevelVsVolumeCurve	278
Table 400 – Association ends of Production::LevelVsVolumeCurve with other classes	278
Table 401 – Attributes of Production::NuclearGeneratingUnit	278
Table 402 – Association ends of Production::NuclearGeneratingUnit with other classes	280
Table 403 – Attributes of Production::PenstockLossCurve	280
Table 404 – Association ends of Production::PenstockLossCurve with other classes	281
Table 405 – Literals of Production::PenstockType	281
Table 406 – Attributes of Production::Reservoir	281
Table 407 – Association ends of Production::Reservoir with other classes	282
Table 408 – Attributes of Production::ShutdownCurve	283
Table 409 – Association ends of Production::ShutdownCurve with other classes	283
Table 410 – Literals of Production::SpillwayGateType	283
Table 411 – Attributes of Production::StartIgnFuelCurve	284
Table 412 – Association ends of Production::StartIgnFuelCurve with other classes	284
Table 413 – Attributes of Production::StartMainFuelCurve	285
Table 414 – Association ends of Production::StartMainFuelCurve with other classes	285
Table 415 – Attributes of Production::StartRampCurve	286
Table 416 – Association ends of Production::StartRampCurve with other classes	286
Table 417 – Attributes of Production::StartupModel	286
Table 418 – Association ends of Production::StartupModel with other classes	287
Table 419 – Attributes of Production::SteamSendoutSchedule	287
Table 420 – Association ends of Production::SteamSendoutSchedule with other classes	288
Table 421 – Literals of Production::SurgeTankCode	288
Table 422 – Attributes of Production::TailbayLossCurve	288
Table 423 – Association ends of Production::TailbayLossCurve with other classes	289
Table 424 – Attributes of Production::TargetLevelSchedule	289
Table 425 – Association ends of Production::TargetLevelSchedule with other classes	289
Table 426 – Attributes of Production::ThermalGeneratingUnit	290
Table 427 – Association ends of Production::ThermalGeneratingUnit with other classes	291

Table 428 – Attributes of Production::WindGeneratingUnit	292
Table 429 – Association ends of Production::WindGeneratingUnit with other classes	293
Table 430 – Attributes of LoadModel::ConformLoad	295
Table 431 – Association ends of LoadModel::ConformLoad with other classes	296
Table 432 – Attributes of LoadModel::ConformLoadGroup	296
Table 433 – Association ends of LoadModel::ConformLoadGroup with other classes	297
Table 434 – Attributes of LoadModel::ConformLoadSchedule	297
Table 435 – Association ends of LoadModel::ConformLoadSchedule with other classes	297
Table 436 – Attributes of LoadModel::DayType	298
Table 437 – Association ends of LoadModel::DayType with other classes	298
Table 438 – Attributes of LoadModel::EnergyArea	298
Table 439 – Association ends of LoadModel::EnergyArea with other classes	298
Table 440 – Attributes of LoadModel::LoadArea	299
Table 441 – Association ends of LoadModel::LoadArea with other classes	299
Table 442 – Attributes of LoadModel::LoadGroup	299
Table 443 – Association ends of LoadModel::LoadGroup with other classes	299
Table 444 – Attributes of LoadModel::LoadResponseCharacteristic	300
Table 445 – Association ends of LoadModel::LoadResponseCharacteristic with other classes	301
Table 446 – Attributes of LoadModel::NonConformLoad	301
Table 447 – Association ends of LoadModel::NonConformLoad with other classes	302
Table 448 – Attributes of LoadModel::NonConformLoadGroup	302
Table 449 – Association ends of LoadModel::NonConformLoadGroup with other classes	303
Table 450 – Attributes of LoadModel::NonConformLoadSchedule	303
Table 451 – Association ends of LoadModel::NonConformLoadSchedule with other classes	303
Table 452 – Attributes of LoadModel::PowerCutZone	304
Table 453 – Association ends of LoadModel::PowerCutZone with other classes	304
Table 454 – Attributes of LoadModel::Season	304
Table 455 – Association ends of LoadModel::Season with other classes	305
Table 456 – Attributes of LoadModel::SeasonDayTypeSchedule	305
Table 457 – Association ends of LoadModel::SeasonDayTypeSchedule with other classes	305
Table 458 – Literals of LoadModel::SeasonName	306
Table 459 – Attributes of LoadModel::StationSupply	306
Table 460 – Association ends of LoadModel::StationSupply with other classes	307
Table 461 – Attributes of LoadModel::SubLoadArea	307
Table 462 – Association ends of LoadModel::SubLoadArea with other classes	308
Table 463 – Attributes of Outage::ClearanceTag	310
Table 464 – Association ends of Outage::ClearanceTag with other classes	310
Table 465 – Attributes of Outage::ClearanceTagType	310
Table 466 – Association ends of Outage::ClearanceTagType with other classes	311
Table 467 – Attributes of Outage::OutageSchedule	311

Table 468 – Association ends of Outage::OutageSchedule with other classes	311
Table 469 – Attributes of Outage::SwitchingOperation	312
Table 470 – Association ends of Outage::SwitchingOperation with other classes	312
Table 471 – Literals of Outage::SwitchState	312
Table 472 – Attributes of AuxiliaryEquipment::AuxiliaryEquipment	313
Table 473 – Association ends of AuxiliaryEquipment::AuxiliaryEquipment with other classes	314
Table 474 – Attributes of AuxiliaryEquipment::CurrentTransformer	314
Table 475 – Association ends of AuxiliaryEquipment::CurrentTransformer with other classes	315
Table 476 – Attributes of AuxiliaryEquipment::FaultIndicator	315
Table 477 – Association ends of AuxiliaryEquipment::FaultIndicator with other classes	316
Table 478 – Attributes of AuxiliaryEquipment::PostLineSensor	316
Table 479 – Association ends of AuxiliaryEquipment::PostLineSensor with other classes	317
Table 480 – Attributes of AuxiliaryEquipment::PotentialTransformer	317
Table 481 – Association ends of AuxiliaryEquipment::PotentialTransformer with other classes	318
Table 482 – Attributes of AuxiliaryEquipment::Sensor	318
Table 483 – Association ends of AuxiliaryEquipment::Sensor with other classes	319
Table 484 – Attributes of AuxiliaryEquipment::SurgeProtector	319
Table 485 – Association ends of AuxiliaryEquipment::SurgeProtector with other classes	320
Table 486 – Attributes of Protection::CurrentRelay	321
Table 487 – Association ends of Protection::CurrentRelay with other classes	322
Table 488 – Attributes of Protection::ProtectionEquipment	322
Table 489 – Association ends of Protection::ProtectionEquipment with other classes	323
Table 490 – Attributes of Protection::RecloseSequence	323
Table 491 – Association ends of Protection::RecloseSequence with other classes	323
Table 492 – Attributes of Protection::SynchrocheckRelay	324
Table 493 – Association ends of Protection::SynchrocheckRelay with other classes	324
Table 494 – Attributes of Equivalents::EquivalentBranch	326
Table 495 – Association ends of Equivalents::EquivalentBranch with other classes	326
Table 496 – Attributes of Equivalents::EquivalentEquipment	327
Table 497 – Association ends of Equivalents::EquivalentEquipment with other classes	327
Table 498 – Attributes of Equivalents::EquivalentInjection	328
Table 499 – Association ends of Equivalents::EquivalentInjection with other classes	328
Table 500 – Attributes of Equivalents::EquivalentNetwork	329
Table 501 – Association ends of Equivalents::EquivalentNetwork with other classes	329
Table 502 – Attributes of Equivalents::EquivalentShunt	329
Table 503 – Association ends of Equivalents::EquivalentShunt with other classes	330
Table 504 – Attributes of Meas::Accumulator	334
Table 505 – Association ends of Meas::Accumulator with other classes	334
Table 506 – Attributes of Meas::AccumulatorLimit	334
Table 507 – Association ends of Meas::AccumulatorLimit with other classes	335

Table 508 – Attributes of Meas::AccumulatorLimitSet	335
Table 509 – Association ends of Meas::AccumulatorLimitSet with other classes	335
Table 510 – Attributes of Meas::AccumulatorValue	335
Table 511 – Association ends of Meas::AccumulatorValue with other classes	336
Table 512 – Attributes of Meas::Analog.....	336
Table 513 – Association ends of Meas::Analog with other classes	337
Table 514 – Attributes of Meas::AnalogLimit	337
Table 515 – Association ends of Meas::AnalogLimit with other classes	337
Table 516 – Attributes of Meas::AnalogLimitSet	338
Table 517 – Association ends of Meas::AnalogLimitSet with other classes	338
Table 518 – Attributes of Meas::AnalogValue.....	338
Table 519 – Association ends of Meas::AnalogValue with other classes	339
Table 520 – Attributes of Meas::Command.....	339
Table 521 – Association ends of Meas::Command with other classes	340
Table 522 – Attributes of Meas::Control	340
Table 523 – Association ends of Meas::Control with other classes	340
Table 524 – Attributes of Meas::ControlType	341
Table 525 – Association ends of Meas::ControlType with other classes	341
Table 526 – Attributes of Meas::Discrete.....	341
Table 527 – Association ends of Meas::Discrete with other classes	342
Table 528 – Attributes of Meas::DiscreteValue.....	342
Table 529 – Association ends of Meas::DiscreteValue with other classes	342
Table 530 – Attributes of Meas::Limit.....	343
Table 531 – Association ends of Meas::Limit with other classes	343
Table 532 – Attributes of Meas::LimitSet.....	343
Table 533 – Association ends of Meas::LimitSet with other classes	344
Table 534 – Attributes of Meas::Measurement	345
Table 535 – Association ends of Meas::Measurement with other classes	345
Table 536 – Attributes of Meas::MeasurementValue	346
Table 537 – Association ends of Meas::MeasurementValue with other classes	346
Table 538 – Attributes of Meas::MeasurementValueQuality.....	347
Table 539 – Association ends of Meas::MeasurementValueQuality with other classes	347
Table 540 – Attributes of Meas::MeasurementValueSource.....	347
Table 541 – Association ends of Meas::MeasurementValueSource with other classes	348
Table 542 – Attributes of Meas::Quality61850.....	348
Table 543 – Attributes of Meas::SetPoint	349
Table 544 – Association ends of Meas::SetPoint with other classes	349
Table 545 – Attributes of Meas::StringMeasurement	349
Table 546 – Association ends of Meas::StringMeasurement with other classes	350
Table 547 – Attributes of Meas::StringMeasurementValue	350
Table 548 – Association ends of Meas::StringMeasurementValue with other classes	350
Table 549 – Literals of Meas::Validity	351
Table 550 – Attributes of Meas::ValueAliasSet.....	351

Table 551 – Association ends of Meas::ValueAliasSet with other classes	351
Table 552 – Attributes of Meas::ValueToAlias	352
Table 553 – Association ends of Meas::ValueToAlias with other classes	352
Table 554 – Attributes of SCADA::CommunicationLink	353
Table 555 – Association ends of SCADA::CommunicationLink with other classes	354
Table 556 – Attributes of SCADA::RemoteControl	354
Table 557 – Association ends of SCADA::RemoteControl with other classes	354
Table 558 – Attributes of SCADA::RemotePoint	355
Table 559 – Association ends of SCADA::RemotePoint with other classes	355
Table 560 – Attributes of SCADA::RemoteSource	355
Table 561 – Association ends of SCADA::RemoteSource with other classes	356
Table 562 – Attributes of SCADA::RemoteUnit	356
Table 563 – Association ends of SCADA::RemoteUnit with other classes	356
Table 564 – Literals of SCADA::RemoteUnitType	357
Table 565 – Literals of SCADA::Source	357
Table 566 – Attributes of ControlArea::AltGeneratingUnitMeas	360
Table 567 – Association ends of ControlArea::AltGeneratingUnitMeas with other classes	360
Table 568 – Attributes of ControlArea::AltTieMeas	360
Table 569 – Association ends of ControlArea::AltTieMeas with other classes	360
Table 570 – Attributes of ControlArea::ControlArea	361
Table 571 – Association ends of ControlArea::ControlArea with other classes	361
Table 572 – Association ends of ControlArea::ControlAreaGeneratingUnit with other classes	362
Table 573 – Literals of ControlArea::ControlAreaTypeKind	362
Table 574 – Attributes of ControlArea::TieFlow	362
Table 575 – Association ends of ControlArea::TieFlow with other classes	362
Table 576 – Attributes of Contingency::Contingency	363
Table 577 – Association ends of Contingency::Contingency with other classes	364
Table 578 – Attributes of Contingency::ContingencyElement	364
Table 579 – Association ends of Contingency::ContingencyElement with other classes	364
Table 580 – Attributes of Contingency::ContingencyEquipment	364
Table 581 – Association ends of Contingency::ContingencyEquipment with other classes	365
Table 582 – Literals of Contingency::ContingencyEquipmentStatusKind	365
Table 583 – Attributes of StateVariables::SvInjection	367
Table 584 – Association ends of StateVariables::SvInjection with other classes	367
Table 585 – Attributes of StateVariables::SvPowerFlow	367
Table 586 – Association ends of StateVariables::SvPowerFlow with other classes	367
Table 587 – Attributes of StateVariables::SvShortCircuit	368
Table 588 – Association ends of StateVariables::SvShortCircuit with other classes	368
Table 589 – Attributes of StateVariables::SvShuntCompensatorSections	368
Table 590 – Association ends of StateVariables::SvShuntCompensatorSections with other classes	368

Table 591 – Attributes of StateVariables::SvStatus	369
Table 592 – Association ends of StateVariables::SvStatus with other classes	369
Table 593 – Attributes of StateVariables::SvTapStep	369
Table 594 – Association ends of StateVariables::SvTapStep with other classes	369
Table 595 – Attributes of StateVariables::SvVoltage	369
Table 596 – Association ends of StateVariables::SvVoltage with other classes	370

IECNORM.COM : Click to view the full PDF of IEC 61970-301:2013

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**ENERGY MANAGEMENT SYSTEM APPLICATION
PROGRAM INTERFACE (EMS-API) –****Part 301: Common information model (CIM) base**

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

International Standard IEC 61970-301 has been prepared by IEC technical committee 57: Power systems management and associated information exchange.

This fifth edition cancels and replaces the fourth edition, published in 2013-05. This fifth edition constitutes a technical revision.

Major changes from the fourth edition include the following.

- transformer models have been modified to be consistent for use by distribution and transmission purposes. Additionally the tap changer model was updated to more clearly reflect the intended usage without relying upon rules for which attributes are appropriate in which situations;
- a more general and clear naming approach was added and several ambiguous attributes related to naming were dropped. The approach allows for users to define new name domains and to give them their own unique description;
- phase component wires models have been enhanced to describe internal phase specific attributes and connections;

- addition of diagram layout models to facilitate the exchange of diagram layout information;
- addition of new data types for Decimal, and clean up of date and time types;
- addition of new Compound data types to the Domain package.

The text of this standard is based on the following documents:

FDIS	Report on voting
57/1395/FDIS	57/1417/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts of the IEC 61970 series, under the general title: *Energy management system application program interface (EMS-API)*, can be found on the IEC website.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

This standard is one of the IEC 61970 series which define an application program interface (API) for an energy management system (EMS). This standard was originally based upon the work of the EPRI Control Center API (CCAPI) research project (RP-3654-1). The principal objectives of the EPRI CCAPI project were to:

- reduce the cost and time needed to add new applications to an EMS;
- protect the investment of existing applications or systems that are working effectively with an EMS.

The principal objective of the IEC 61970 series of standards is to produce standards which facilitate the integration of EMS applications developed independently by different vendors, between entire EMS systems developed independently, or between an EMS system and other systems concerned with different aspects of power system operations, such as generation or distribution management systems (DMS). This is accomplished by defining application program interfaces to enable these applications or systems access to public data and exchange information independent of how such information is represented internally.

The common information model (CIM) specifies the semantics for this API. The component interface specifications (CIS), which are contained in other parts of the IEC 61970 standards, specify the content of the messages exchanged.

The CIM is an abstract model that represents all the major objects in an electric utility enterprise typically needed to model the operational aspects of a utility. This model includes public classes and attributes for these objects, as well as the relationships between them.

The objects represented in the CIM are abstract in nature and may be used in a wide variety of applications. The use of the CIM goes far beyond its application in an EMS. This standard should be understood as a tool to enable integration in any domain where a common power system model is needed to facilitate interoperability and plug compatibility between applications and systems independent of any particular implementation.

This standard, IEC 61970-301, defines the CIM base set of packages which provide a logical view of the functional aspects of an energy management system including SCADA. Other functional areas are standardized in separate IEC documents that augment and reference this base CIM standard. For example, IEC 61968-11 addresses distribution models and references this base CIM standard. While there are multiple IEC standards dealing with different parts of the CIM, there is a single, unified information model comprising the CIM behind all these individual standards documents.

The International Electrotechnical Commission (IEC) draws attention to the fact that it is claimed that compliance with this document may involve the use of a patent concerning a computer-based implementation of an object-oriented power system model in a relational database. As such, it does not conflict with the development of any logical power system model including the common information model (CIM), where implementation of the model is not defined.

The IEC takes no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured the IEC that he/she is willing to negotiate licences either free of charge or under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with IEC. Information may be obtained from:

ICL

Wenlock Way

West Gorton

Manchester

M12 5DR

United Kingdom (U.K.)

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. IEC shall not be held responsible for identifying any or all such patent rights.

ISO (www.iso.org/patents) and IEC (http://www.iec.ch/tctools/patent_decl.htm) maintain on-line data bases of patents relevant to their standards. Users are encouraged to consult the data bases for the most up to date information concerning patents.

Withdrawing
IECNORM.COM : Click to view the full PDF of IEC 61970-301:2013

ENERGY MANAGEMENT SYSTEM APPLICATION PROGRAM INTERFACE (EMS-API) –

Part 301: Common information model (CIM) base

1 Scope

The common information model (CIM) is an abstract model that represents all the major objects in an electric utility enterprise typically involved in utility operations. By providing a standard way of representing power system resources as object classes and attributes, along with their relationships, the CIM facilitates the integration of Energy Management System (EMS) applications developed independently by different vendors, between entire EMS systems developed independently, or between an EMS system and other systems concerned with different aspects of power system operations, such as generation or distribution management. SCADA is modeled to the extent necessary to support power system simulation and inter-control center communication. The CIM facilitates integration by defining a common language (i.e. semantics) based on the CIM to enable these applications or systems to access public data and exchange information independent of how such information is represented internally.

The object classes represented in the CIM are abstract in nature and may be used in a wide variety of applications. The use of the CIM goes far beyond its application in an EMS. This standard should be understood as a tool to enable integration in any domain where a common power system model is needed to facilitate interoperability and plug compatibility between applications and systems independent of any particular implementation.

Due to the size of the complete CIM, the object classes contained in the CIM are grouped into a number of logical Packages, each of which represents a certain part of the overall power system being modeled. Collections of these Packages are progressed as separate International Standards. This particular International Standard specifies a Base set of packages which provide a logical view of the functional aspects of Energy Management System (EMS) information within the electric utility enterprise that is shared between all applications. Other standards specify more specific parts of the model that are needed by only certain applications. Subclause 4.2 below provides the current grouping of packages into standards documents.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 60050 (all parts), *International Electrotechnical Vocabulary (IEV)*
<http://www.electropedia.org>

IEC 60870-6 (all parts), *Telecontrol equipment and systems – Part 6: Telecontrol protocols compatible with ISO standards and ITU-T recommendations*

IEC 61850 (all parts), *Communication networks and systems for power utility automation*

IEC 61850-7-3:2010, *Communication networks and systems for power utility automation – Part 7-3: Basic communication structure – Common data classes*

IEC 61850-7-4:2010, *Communication networks and systems for power utility automation – Part 7-4: Basic communication structure – Compatible logical node classes and data object classes*

IEC 61968 (all parts), *Application integration at electric utilities – System interfaces for distribution management*

IEC/TS 61970-2, *Energy management system application program interface (EMS-API) – Glossary*

IEC 62325 (all parts), *Framework for energy market communications*

Object Management Group: UML 2.0 Specification – <http://www.omg.org>

3 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 60050 and IEC/TS 61970-2 apply, as well as the following.

3.1

energy management system

EMS

computer system comprising a software platform providing basic support services and a set of applications providing the functionality needed for the effective operation of electrical generation and transmission facilities so as to ensure adequate security of energy supply at minimum cost

3.2

application program interface

API

the set of public functions provided by an executable application component for use by other executable application components

3.3

unified modeling language

UML

formal and comprehensive descriptive language with diagramming techniques used to represent software systems, from requirements analysis, through design and implementation, to documentation

Note 1 to entry: UML has evolved from a collection of methods contributed by different practitioners, into an International Standard. The CIM relies on UML for defining the model, and automated tools generate the documentation, schemas, and other artifacts directly from the UML. A basic understanding of UML is necessary to understand the CIM.

3.4

profile

subset of DCIM classes, associations and attributes needed to accomplish a specific type of interface

Note 1 to entry: It may be expressed in XSD, RDF, and/or OWL files. A profile can be tested between applications. A profile is necessary in order to “use” the DCIM. Several profiles are defined in other parts of the IEC 61968 series of standards.

4 CIM specification

4.1 CIM modeling notation

The CIM is defined using object-oriented modeling techniques. Specifically, the CIM specification uses the Unified Modeling Language (UML) notation, which defines the CIM as a group of packages.

Each package in the CIM contains one or more class diagrams showing graphically all the classes in that package and their relationships. Each class is then defined in text in terms of its attributes and relationships to other classes.

The UML notation is described in Object Management Group (OMG) documents and several published textbooks.

4.2 CIM packages

The CIM is partitioned into a set of packages. A package is a general purpose means of grouping related model elements. The packages have been chosen to make the model easier to design, understand and review. The common information model consists of the complete set of packages. Entities may have associations that cross many package boundaries. Each application will use information represented in several packages.

The comprehensive CIM is partitioned into groups of packages for convenience in managing and maintaining them. Packages included in IEC 61970-301 (this document) are contained within the IEC 61970 package. The IEC 61970 packages do not depend upon other outside packages. As shown in Figure 1, the IEC 61970 package and its subpackages are used as a core or base model for other CIM packages. The dashed lines indicate dependency relationships, with the arrowhead pointing from the dependent package to the package on which it has a dependency. Packages for the IEC 61968 series of standards describe additional parts of the CIM that deal with other logical views of utility operations including assets, locations, activities, consumers, documentation, work management, and metering. Packages for the IEC 62325 series of standards describe electric energy markets.

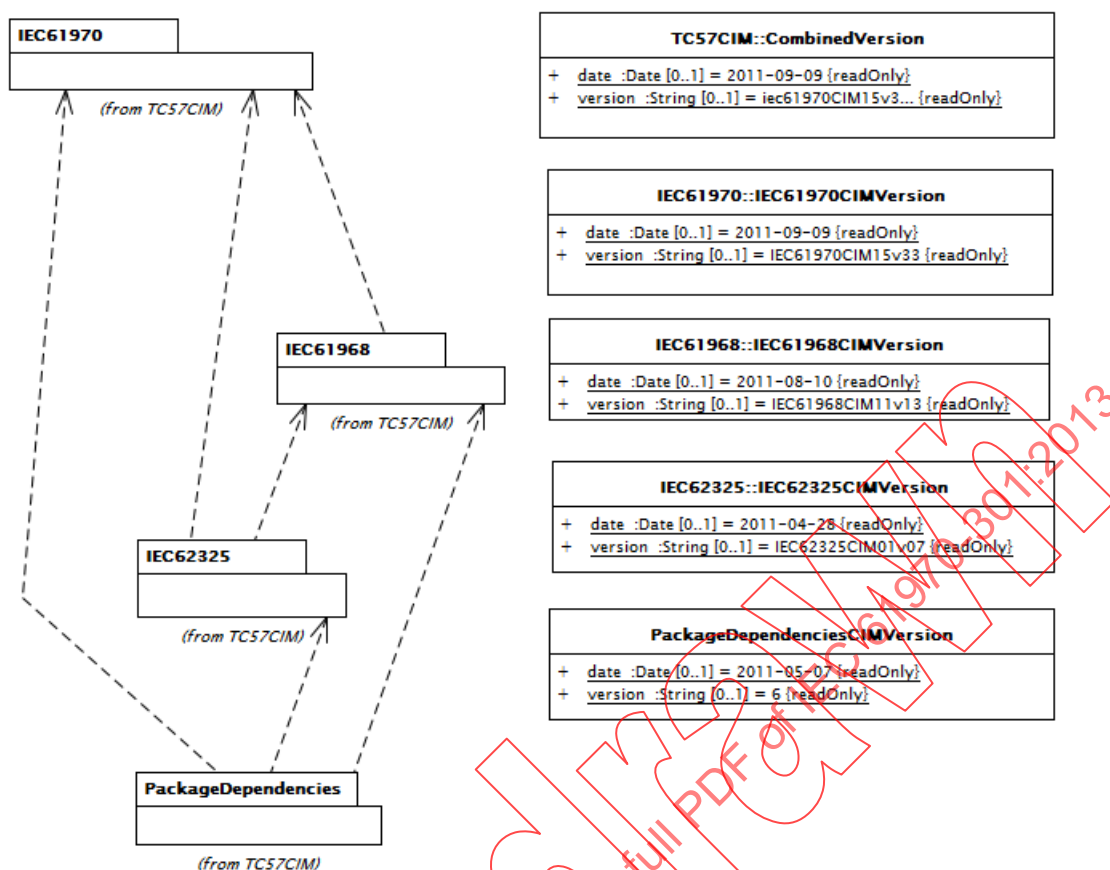


Figure 1 – Example of working group package dependencies

Note that the package boundaries do not imply application boundaries. An application may use CIM entities from several packages. It is also anticipated CIM packages outside of this standard will have dependencies upon some of the packages described in this standard, and particularly the Domain and Core packages, though other dependencies will also exist.

Figure 2 shows the packages defined for IEC 61970-301 CIM Base and their dependency relationships.

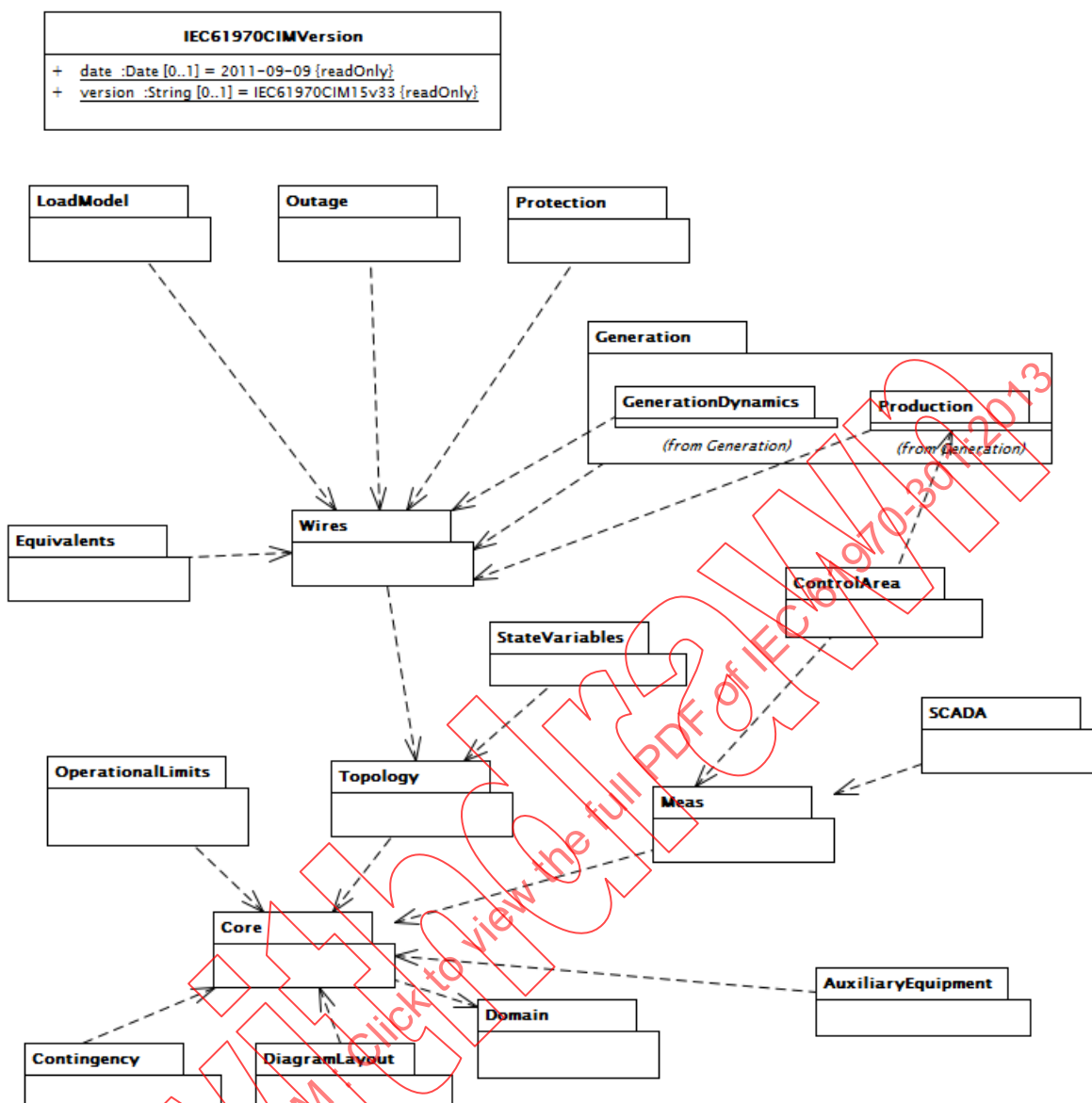


Figure 2 – CIM IEC 61970-301 package diagram

Clause 6 contains the specification for each of the CIM Base packages.

NOTE The contents of the CIM defined in this specification were auto-generated from the CIM UML electronic model release IEC61970CIM15v33, which is available through the CIM Users Group.

4.3 CIM classes and relationships

4.3.1 Classes

The class diagram(s) for each CIM package shows all the classes in the package and their relationships. Where relationships exist with classes in other packages, those classes may also be shown.

Classes and objects model what is in a power system that needs to be represented in a common way to power system applications. A class is a description of an object found in the real world, such as a power transformer, generator, or load that needs to be represented as part of the overall power system model in, for example, an EMS. Other types of objects include things such as schedules and measurements that EMS applications also need to process, analyze, and store. Such objects need a common representation to achieve the

purposes of the EMS-API standard for plug-compatibility and interoperability. A particular object in a power system with a unique identity is modeled as an instance of the class to which it belongs.

It should also be noted that the CIM is defined to facilitate data exchange. As defined in this document, CIM entities have no behaviour. For a specific interface, a profile is defined consisting of subsets of CIM classes, attributes, and associations. A profile defines the message payload for an interface.

Classes have attributes that describe the characteristics of the objects. Each class in the CIM contains the attributes that describe and identify a specific instance of the class. Only the attributes that are of public interest to EMS applications are included in the class descriptions.

Each attribute has a type, which identifies what kind of attribute it is. Typical attributes are of type Integer (CIM class Integer), Float (CIM class Float), Boolean (CIM class Boolean), String (CIM class String), Date (CIM class Date), and Decimal (CIM class Decimal), which are called primitive types. However, many additional types are defined as part of the CIM specification. For example, ShuntCompensator has a "maxU" attribute of type voltage (CIM class Voltage). The definition of many shared types is contained in the Domain package described in 6.3. The UML stereotypes of Primitive, enumeration, CIMDatatype, and Compound are added to classes used as types. The CIMDatatype stereotype is used with a specific CIM semantics for a triple of attributes {value, unit, multiplier}, which implies custom mapping to serialization artifacts such as RDFS, OWL, and XSD. Classes with these stereotypes do not participate in generalization or association relationships and are simply used as types for attributes. The enumeration stereotype is used to describe an attribute with an enumerated list of choices. The Compound stereotype is used to describe sets of related attributes that are commonly reused. Compound classes may consist of Primitive, enumeration, CIMDatatype or other Compound classes as long as the Compound classes do not recurse.

All CIM attributes are implicitly optional in the sense that profiles using the CIM may eliminate any attributes.

Relationships between classes reveal how they are structured in terms of each other. CIM classes are related in a variety of ways, as described in the subclauses below.

4.3.2 Generalization

A generalization is a relationship between a more general and a more specific class. The more specific class can contain only additional information. Generalization provides for the specific class to inherit attributes and relationships from all the more general classes above it.

Figure 3 is an example of generalization. In this example taken from the Wires package, a Breaker is a more specific type of ProtectedSwitch, which in turn is a more specific type of Switch, which is a more specific type of ConductingEquipment, etc. A PowerTransformer is another more specific type of ConductingEquipment. Note that PowerSystemResource inherits from class IdentifiedObject which is not on the diagram so IdentifiedObject is shown in italic type in the upper right corner of class PowerSystemResource.

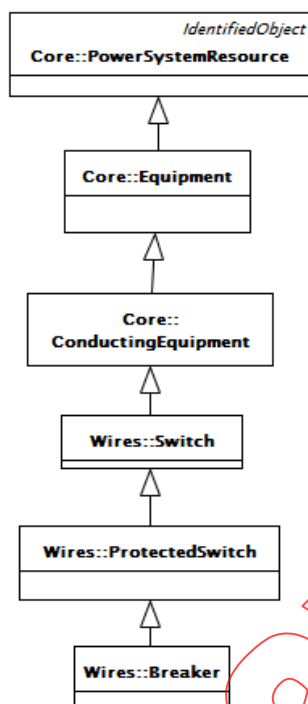


Figure 3 – Example of generalization

4.3.3 Simple association

An association is a conceptual connection between classes. Each association has two “association ends”. The “association ends” were called “roles” prior to the UML 2.0 specification. Each association end describes the role the target class (i.e., the class the association end goes *to*) has in relation to the source class (i.e., the class the association end goes *from*). Association ends are usually given the name of the target class with or without a verb phrase. Each association end also has multiplicity/cardinality, which is an indication of how many objects may participate in the given relationship. In the CIM, associations are not named, only association ends are named. For example, as shown by Figure 4, in the CIM there is an association between class BaseVoltage and class VoltageLevel. Multiplicity is shown at both ends of the association. In this example, a VoltageLevel object may reference 1 BaseVoltage, and a BaseVoltage may be referenced by 0 or more VoltageLevel objects.

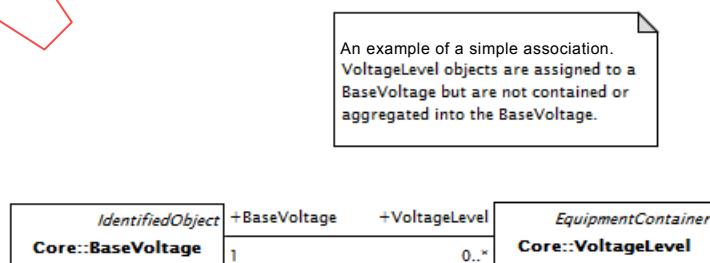


Figure 4 – Example of simple association

4.3.4 Aggregation

Aggregation is a special case of association. Aggregation indicates that the relationship between the classes is some sort of whole-part relationship, where the whole class “consists

of” or “contains” the part class, and the part class is “part of” the whole class. The part class does not inherit from the whole class as in generalization. Figure 5 illustrates an aggregation between the EquipmentContainer class and the Equipment class, which is taken from the Core package. As shown, an Equipment can be a member of zero or one EquipmentContainer objects, but an EquipmentContainer object can contain any number of Equipment objects. In the context of using CIM as an information model, aggregation does not have a precise or formal interpretation beyond a simple association and is intended to visually assist in representing normal usage.

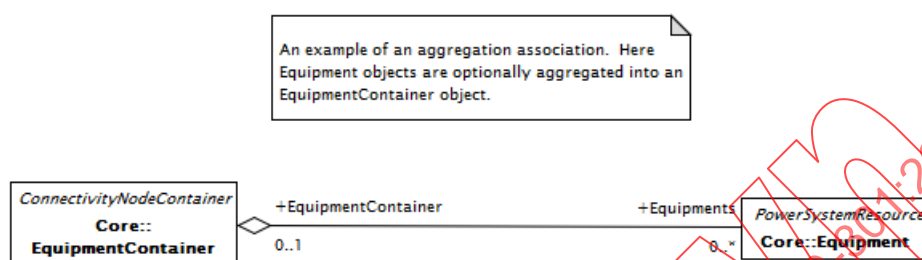


Figure 5 – Example of aggregation

4.4 CIM model concepts and examples

4.4.1 Concepts

The CIM classes, attributes, types, and relationships are specified in Clauses 5 and 6. Clause 6 comprises a complete description of the IEC 61970-301 CIM Base model. To help understand how to interpret the CIM, some key model concepts used in the CIM are introduced and described in the following subclauses.

4.4.2 Containment, equipment hierarchies and naming

4.4.2.1 Containment structure

Figure 6 shows the concept of equipment containers to form hierarchies in the CIM. Equipment containers represent ways of organizing and naming equipment typically found within a substation. As may be seen, there is some flexibility provided in which containers are used in a specific application of the CIM in order to accommodate different international practices as well as differences typically found between transmission and distribution substations. Bay, VoltageLevel, Substation, Line, and Plant are all types of EquipmentContainers. In general, a Bay is contained within a specific VoltageLevel, which in turn is contained within a Substation. Substations and Lines may be contained within SubGeographicRegions and GeographicRegions.

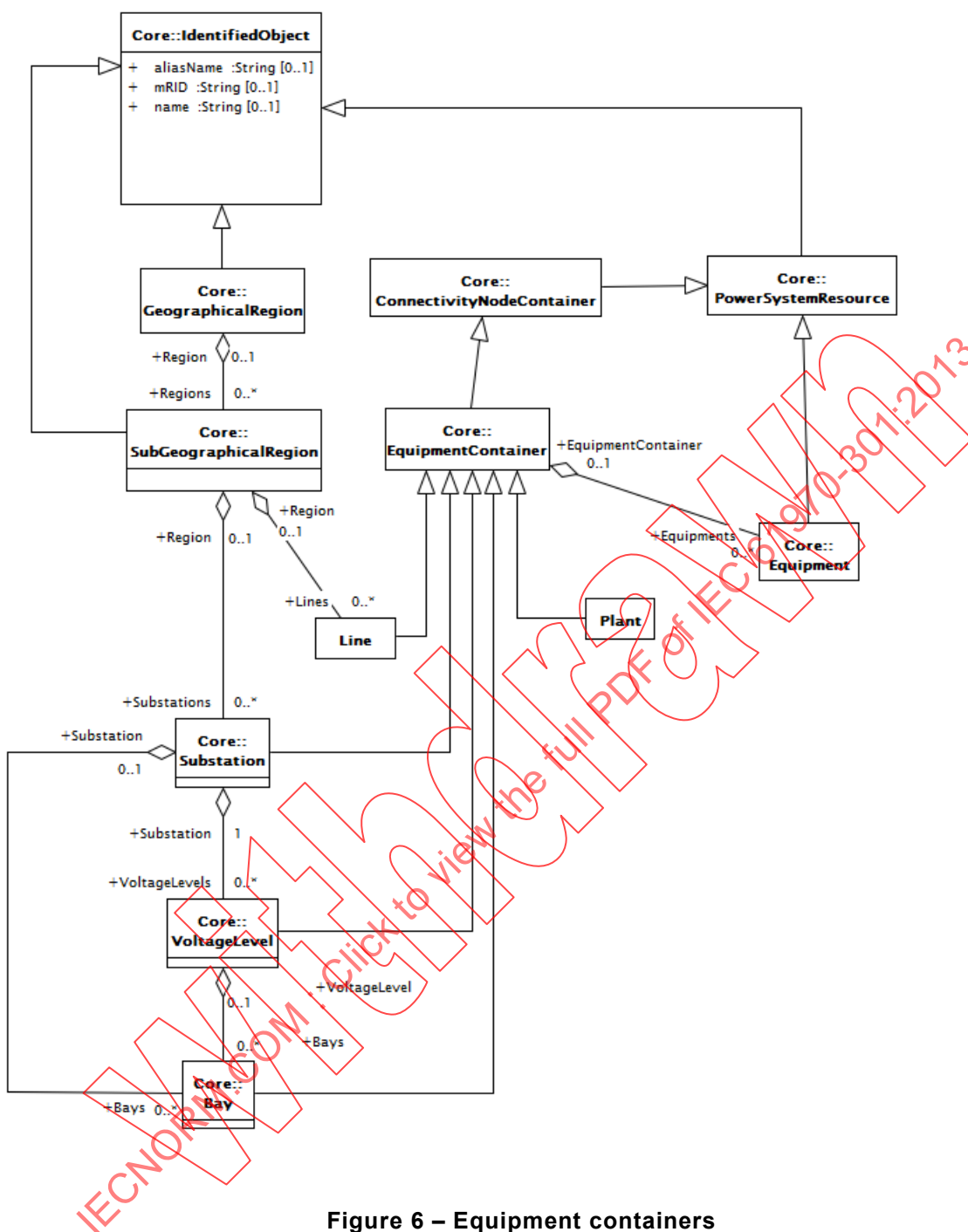


Figure 6 – Equipment containers

The diagrams `NamingHierarchyPart1` and `NamingHierarchyPart2` in the `Wires` package show the functional naming hierarchy (refer to the `Wires` package documentation in 6.8 for the details).

One containment hierarchy is used with the IdentifiedObject class to create hierarchical naming intended for human consumption. This hierarchy is specifically used to name equipment according to its function in the power system. This is the functional naming hierarchy. Other common identifications beside functional names are asset serial numbers. The functional name is different from a serial number in that it relates to the function of a particular equipment position or location in the power system. Regardless of what specific piece of physical equipment is placed at a location, the functional name is the same but the serial number varies depending on the physical equipment currently used.

4.4.2.2 IdentifiedObject class

The IdentifiedObject class contained in the Core package is inherited by all PowerSystemResource and many other classes. This class has attributes and associations to be used for naming all CIM objects. The mRID attribute of the IdentifiedObject class provides a straight forward and rigorous means of identity for CIM objects.

The following are definitions and conventions for how to use the IdentifiedObject attributes when naming PowerSystemResource objects (for more details, refer to documentation for IdentifiedObject and its attributes in Clause 6):

- mRID (Master Resource ID): A globally unique machine-readable identifier for an object instance.
- name: The name is any free human readable and possibly non-unique text naming the object.
- aliasName: This attribute is deprecated and the Name classes of 4.4.3 provide a better alternative. The aliasName is free text human readable name of the object alternative to IdentifiedObject.name. It may be non-unique and may not correlate to a naming hierarchy. The attribute aliasName is retained because of backwards compatibility between CIM releases. It is however recommended to replace aliasName with the Name class as aliasName is planned for retirement at a future time.

4.4.3 Names model

A flexible and extensible naming model is provided to specify alternate names for objects. Figure 7 shows the class diagram for alternate names and how they can be associated to a specific user defined name type. This model allows for specific definition of names within a particular domain and enforces no specific naming rules. The concepts of alias names, path names, and local names can all be accommodated by this model. The CIM models do not enforce or promote specific naming conventions, but allow for such naming conventions to be exchanged in a clear context.

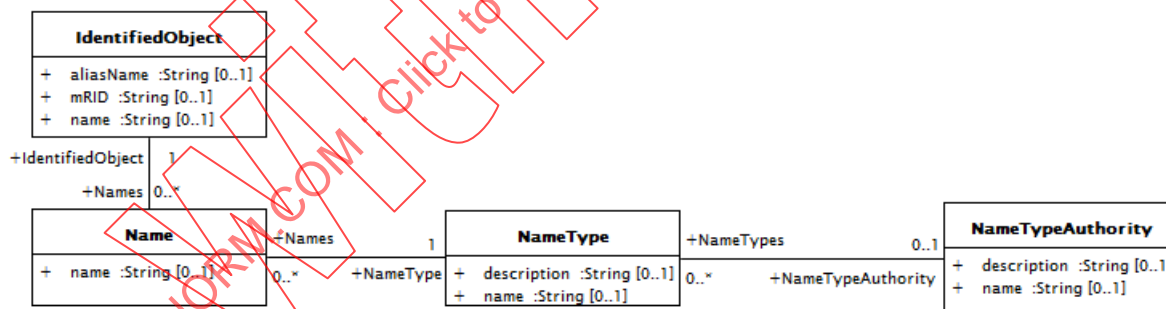


Figure 7 – Names

Table 1 – NameType class naming conventions

name	description
ICCP	Reserved for ICCP (TASE-2) names. Used to describe ICCP point names for the MeasurementValue class and ICCP source names for the MeasurementValueSource class.

Table 1 describes various name conventions for name types. The meanings of the columns in Table 1 are as follows:

name: The name attribute value.

description: The use of the name type.

Note that Table 1 is not an exhaustive list of values that can be used for the name attribute of the class Name.

4.4.4 Connectivity model

4.4.4.1 Connectivity description

Figure 8 shows the Topology class diagram which models connectivity between different types of ConductingEquipment. Also included is a portion of the Meas package class diagram dealing with measurements to illustrate how measurements are associated with conducting equipment.

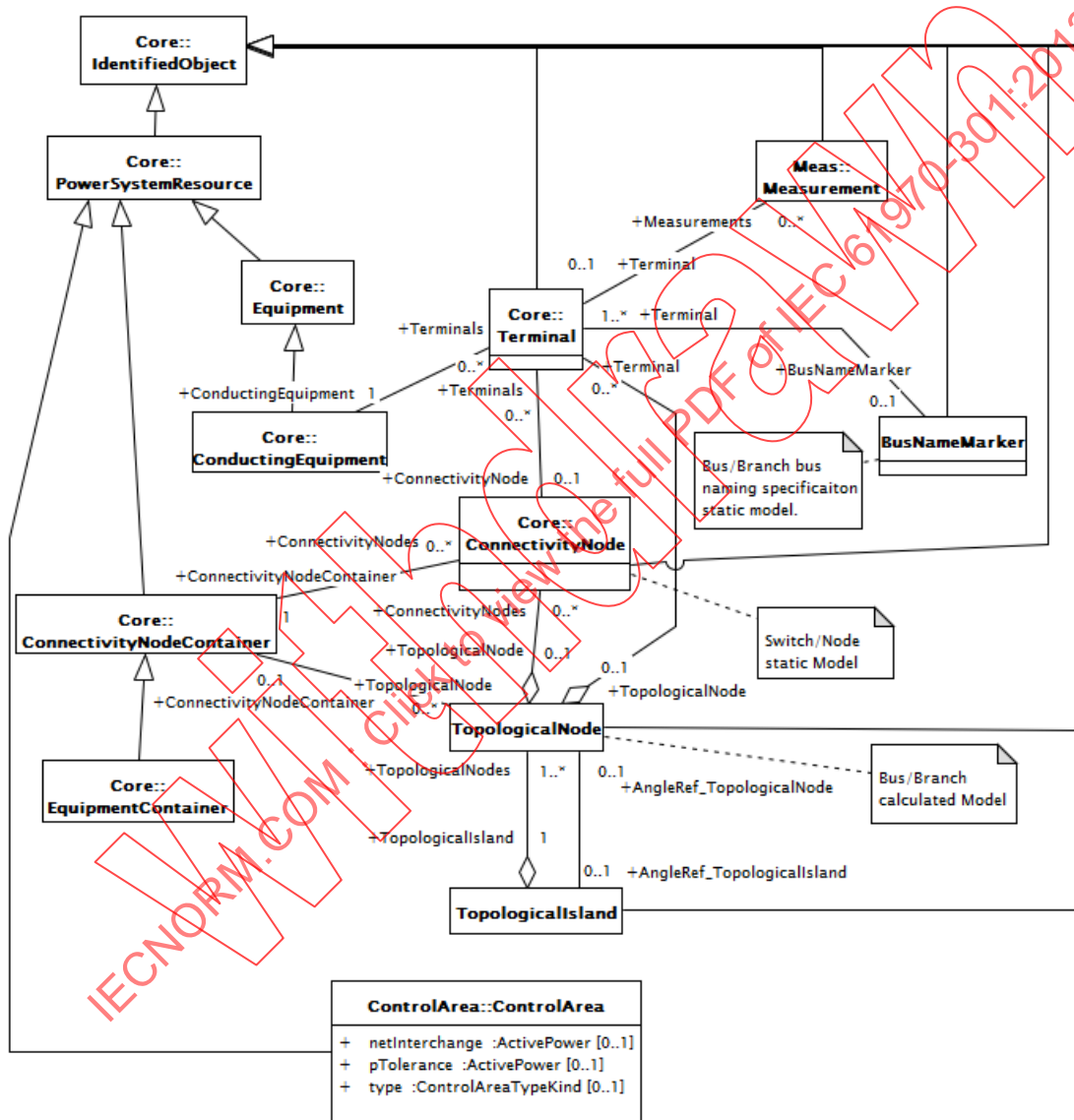


Figure 8 – Connectivity model

To model connectivity, Terminal and ConnectivityNode classes are defined. A Terminal belongs to one ConductingEquipment, although ConductingEquipment may have any number of Terminals. Each Terminal may be connected to a ConnectivityNode, which is a point where terminals of conducting equipment are connected together with zero impedance. A ConnectivityNode may have any number of terminals connected, and may be a member of a TopologicalNode (i.e., a bus), which is in turn a member of a TopologicalIsland.

TopologicalNodes and TopologicalIslands are created as a result of a topology processor evaluating the “as built” topology and the actual Switch positions.

It is possible to exchange messages directly involving TopologicalNode and bypassing the ConnectivityNode details by using the TopologicalNode to Terminal association. This is often useful for exchanges involving bus/branch models which do not typically contain switch detail. When the TopologicalNode model is exchanged it is useful to also exchange the “connected” attribute on the Terminal class. This allows for the topological structure of the bus/branch model to be retained, yet indicate a terminal is not completely connected. Normally this disconnection is the result of one or more switches opening in the detailed model which is not represented in a bus/branch model exchange.

EquipmentContainers, which are a specialization of a ConnectivityNodeContainer, may contain zero or more ConnectivityNodes. The associations, ConductingEquipment – Terminal and Terminal – ConnectivityNode, capture the as-built topology of an actual power system network. For each Terminal connected to a ConnectivityNode, the associations of the other Terminal(s) connected to the same ConnectivityNode identify the ConductingEquipment object(s) that are electrically connected. Similar connectivity at the “bus/branch” level of detail can be expressed using the TopologicalNode instead of the ConnectivityNode. The BusNameMarker class, which has an association to ConnectivityNode, is used to carry user bus names to be applied to TopologicalNodes. TopologicalNodes are created from ConnectivityNodes as a result of topology processing and various names might be applied depending upon which ConnectivityNode instances are present.

To model the analog values such as voltage and power, each Terminal has an association with a Measurement class from the Meas package. Although not shown in Figure 8, a Measurement object is associated with at least one MeasurementValue object. Each MeasurementValue object is an instance of a measurement from a specific source, for example, a telemetered measurement.

Clause 6 contains a complete description of each class in Figure 8 along with the definition of all the attributes and relationships supported in each class.

4.4.4.2 Connectivity and containment example

To illustrate how the connectivity model and containment model would appear as objects, a small example is presented in Figure 9. The example shows a transmission line with a T-junction spanning two substations and a substation having two voltage levels with a transformer between them. The transmission line consists of two different cables. One of the voltage levels is shown with a busbar section having a single busbar and two very simple switchgear bays connecting to the busbar.

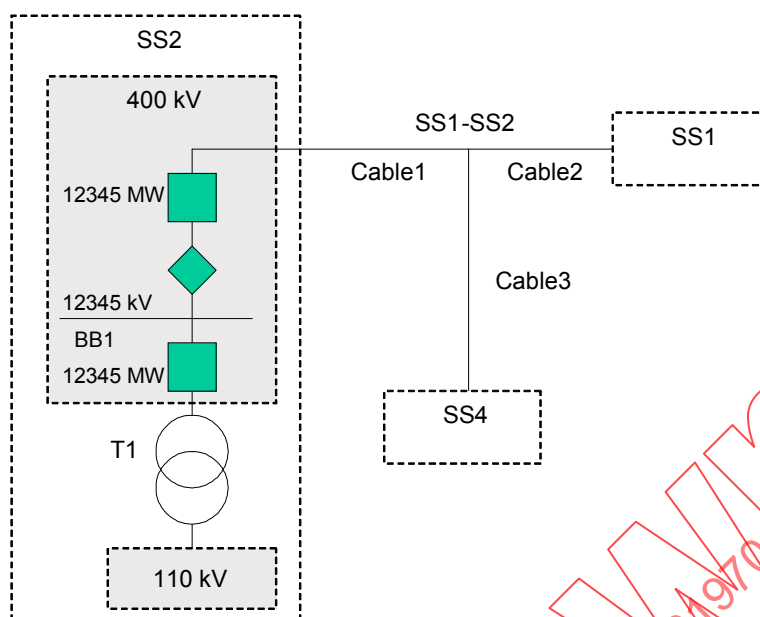


Figure 9 – Simple network example

Figure 10 shows how connectivity is modeled in the CIM as well as one way (but not necessarily the only way) containment is modeled for the diagram in Figure 10. The shaded square boxes represent EquipmentContainers, and the white square boxes represent ConductingEquipment. Darker shading indicates the EquipmentContainer is higher up in the containment hierarchy (i.e., Substation is highest, VoltageLevel next, etc.). White circles represent ConnectivityNodes, and black small circles represent Terminals. A Terminal belongs to a ConductingEquipment, and a ConnectivityNode belongs to an EquipmentContainer. This means that the borders (or contact points) between ConductingEquipment are their Terminals interconnected via ConnectivityNodes

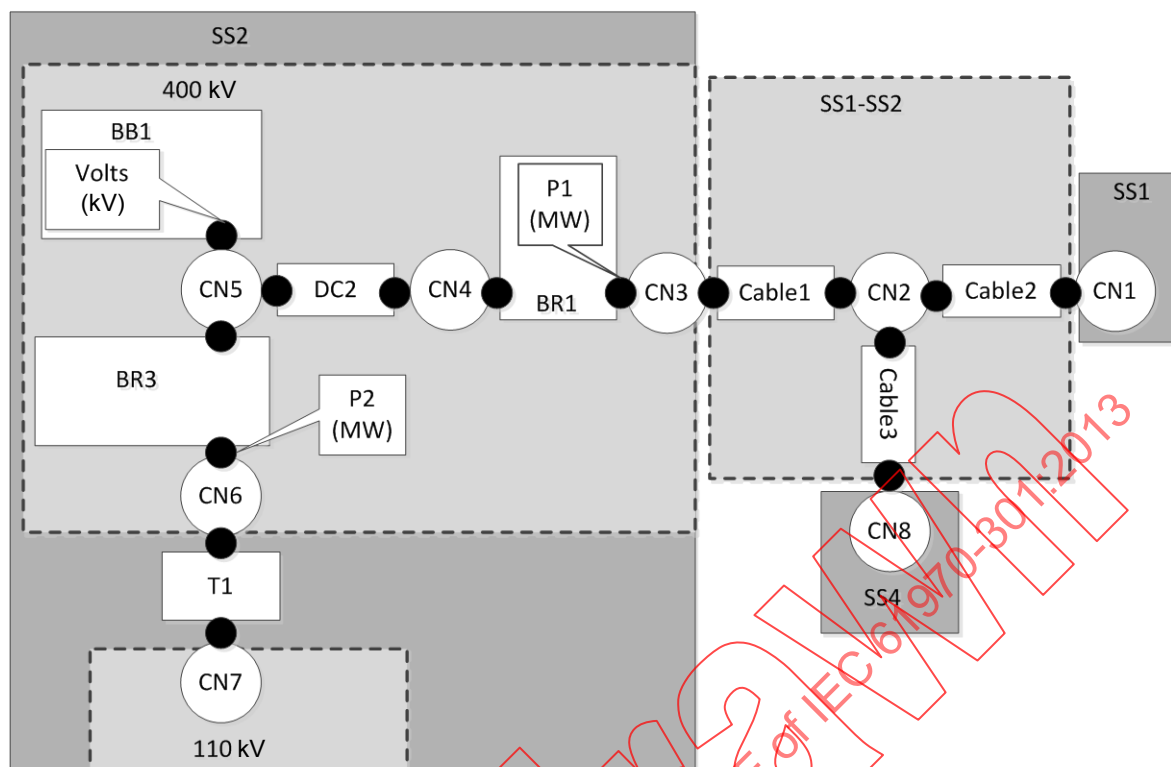


Figure 10 – Simple network connectivity modeled with CIM Topology

The Line SS1-SS2 contains three ACLineSegments (Cable1, Cable2, and Cable3) and associated ConnectivityNode (CN2) to model a T junction, which provides a connection to SS4. This represents just one way that this configuration could be modeled. Each ACLineSegment has two Terminals. Cable1 is connected to CN3 and CN2 via these Terminals. CN3 is contained by the VoltageLevel 400 kV. The breaker BR1 has two terminals of which one is connected to CN3.

Measurements are represented by square callouts where the arrow points to a Terminal. P1 is connected to the right Terminal belonging to Breaker BR1. Note that P1 is drawn inside the box representing BR1. This is because a Measurement may belong to a PowerSystemResource (PSR), as is the case with BR1. P2 is drawn inside the VoltageLevel 400 kV, which means it belongs to the 400 kV VoltageLevel instead of BR3.

4.4.5 Inheritance hierarchy

Figure 11 shows an overview of the inheritance hierarchy modeled in the CIM. This overview, which is included as one of the Wires package diagrams, spans several of the CIM packages.

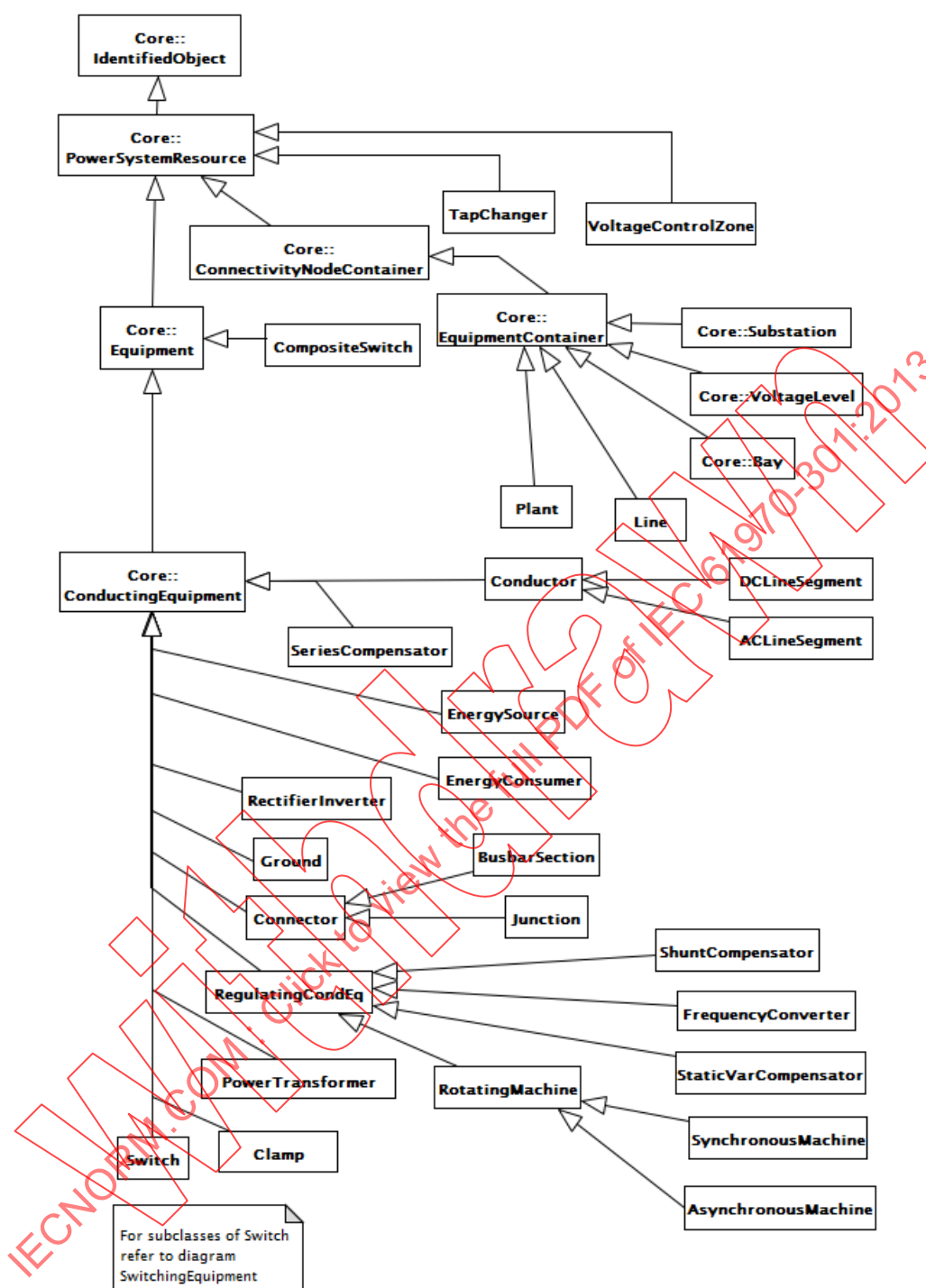


Figure 11 – Equipment inheritance hierarchy

4.4.6 Transformer model

Figure 13 shows a portion of the Wires package in a class diagram which models a PowerTransformer device.

The transformer model has been revised from previous versions to allow for use of the same instance model in both balanced and unbalanced models. Additionally the PowerTransformer itself is now a ConductingEquipment with potentially multiple terminals to more directly model transformers with one, two, three, or more terminals. As typical of ConductingEquipment, when in service, the PowerTransformer conducts power from one of its terminals to another.

As shown in Figure 12, the PowerTransformer is also able to optionally model tank details, which can be used to describe in detail the transformer internal winding phase connections and imbalances. In all cases a PowerTransformer models a group of physical devices acting together to transform power among terminals and in one physical location. Often for transmission systems three physical, single phase devices are represented by one PowerTransformer instance. If detail of the individual single phase devices were required, the TransformerTank objects should additionally be modeled.

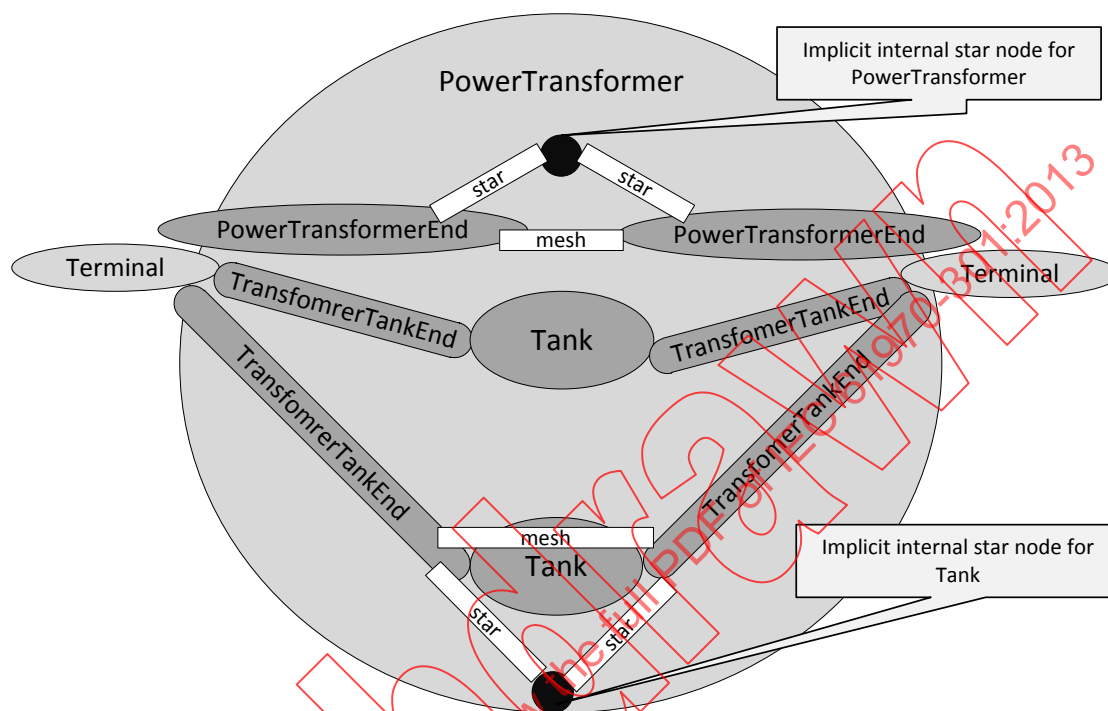


Figure 12 – Transformer and Tank model

Both a PowerTransformer and a TransformerTank can have impedance described in either an assumed star connection with implied center connection or with mesh impedance form by using the optional TransformerMeshImpedance class. The mesh form of impedance is required to accurately model transformers with more than three terminals. If using the mesh impedance form, you would specify a TransformerMeshImpedance class for each possible terminal to terminal connection, thus one instance for two terminals, three for three terminals, and six for four terminals and so on. If using the star connection, the impedance parameters can be specified directly on the PowerTransformerEnd class, or alternatively shared among transformers using the TransformerStarImpedance class.

As shown, a PowerTransformer is a specialized class of ConductingEquipment, which is a specialized class of Equipment. This is shown by the use of the generalization-type of relationship, which uses an arrow to point to the general class. The ConductingEquipment is shown to be a specialized class of Equipment by use of the italic "*Equipment*" class name in the upper right corner. The inheritance permits the PowerTransformer to inherit attributes from ConductingEquipment, Equipment, and all generalizations of Equipment that are not shown in this diagram.

A PowerTransformer also has relationships to PowerTransformerEnd and TransformerTank which are modeled with association type relationships. As shown, a PowerTransformer may associate zero or more PowerTransformerEnds, but a PowerTransformerEnd may associate to only zero or one PowerTransformer. Similarly, the PowerTransformer may associate to zero or more TransformerTank objects, but a TransformerTank may associate to only zero or one PowerTransformer object.

The `PowerTransformerEnd` specializes the `TransformerEnd` class which has other relationships as well:

- a generalization relationship with `IdentifiedObject`;
- two association relationships with the `TransformerMeshImpedance` class, such that a `TransformerEnd` object may be "From" 0, 1, or more `TransformerMeshImpedance` objects and To 0, 1, or more `TransformerMeshImpedance` objects;
- aggregation relationships with the `PhaseTapChanger` and `RatioTapChanger` classes, such that a `TransformerWinding` object may have 0 or 1 `PhaseTapChanger` objects and 0 or 1 `RatioTapChanger` objects associated with it.

Clause 6 contains a complete description of each class in Figure 13 along with the definition of all the attributes and relationships supported in each class.

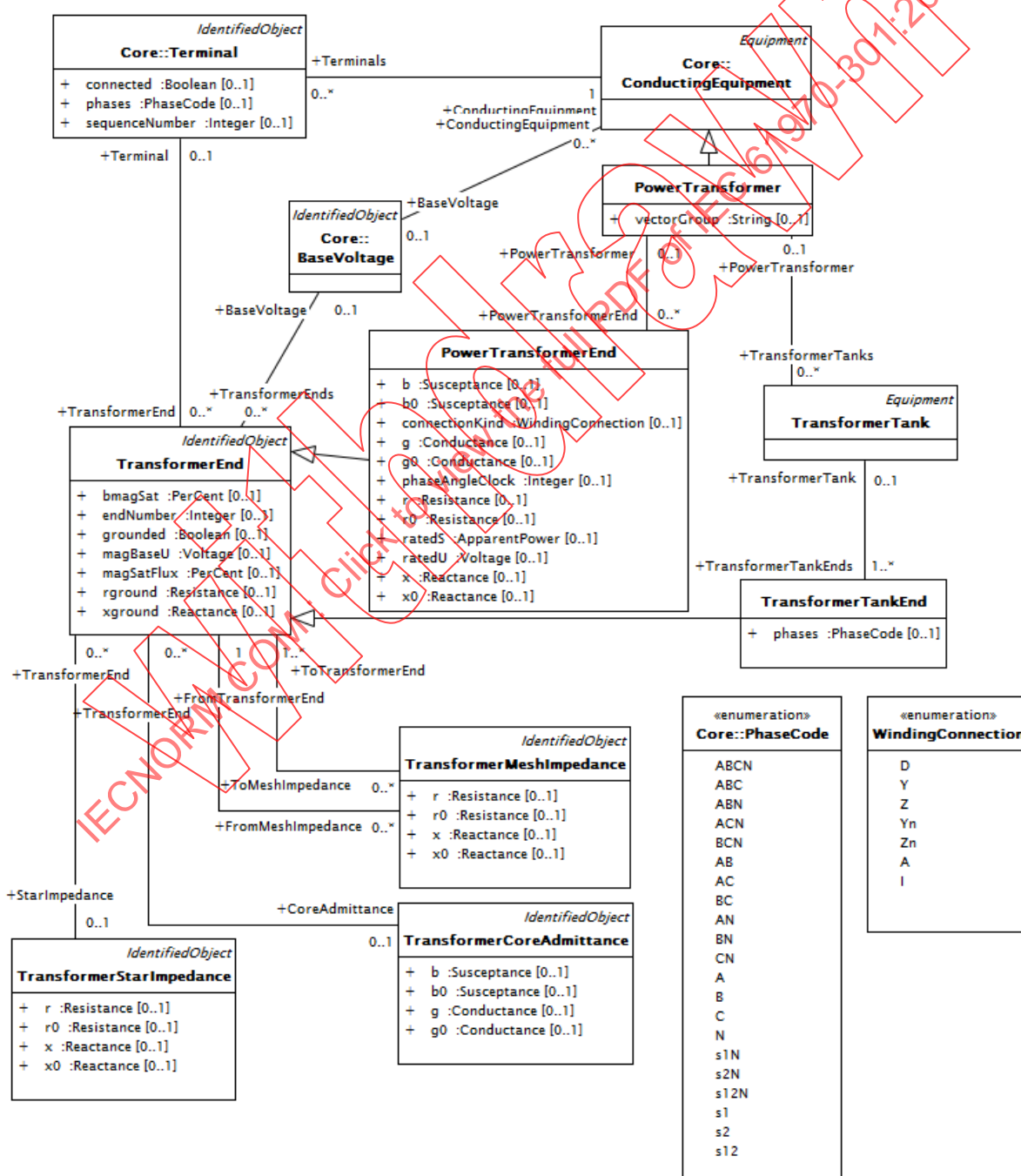


Figure 13 – Transformer model

4.4.7 Transformer tap modeling

4.4.7.1 Generalized tap modeling

Multiple types of transformer tap models are supported by the CIM model. Depending on the physical construction, the effects of tap movement will be defined using either the RatioTapChanger class or the PhaseTapChanger class and its various specializations. The PhaseTapChanger can be specialized into a number of different types representing the physical construction and hence behaviour. The attributes of TapChanger are applicable for all the RatioTapChanger specializations. Figure 14 shows the UML model for the transformer taps.

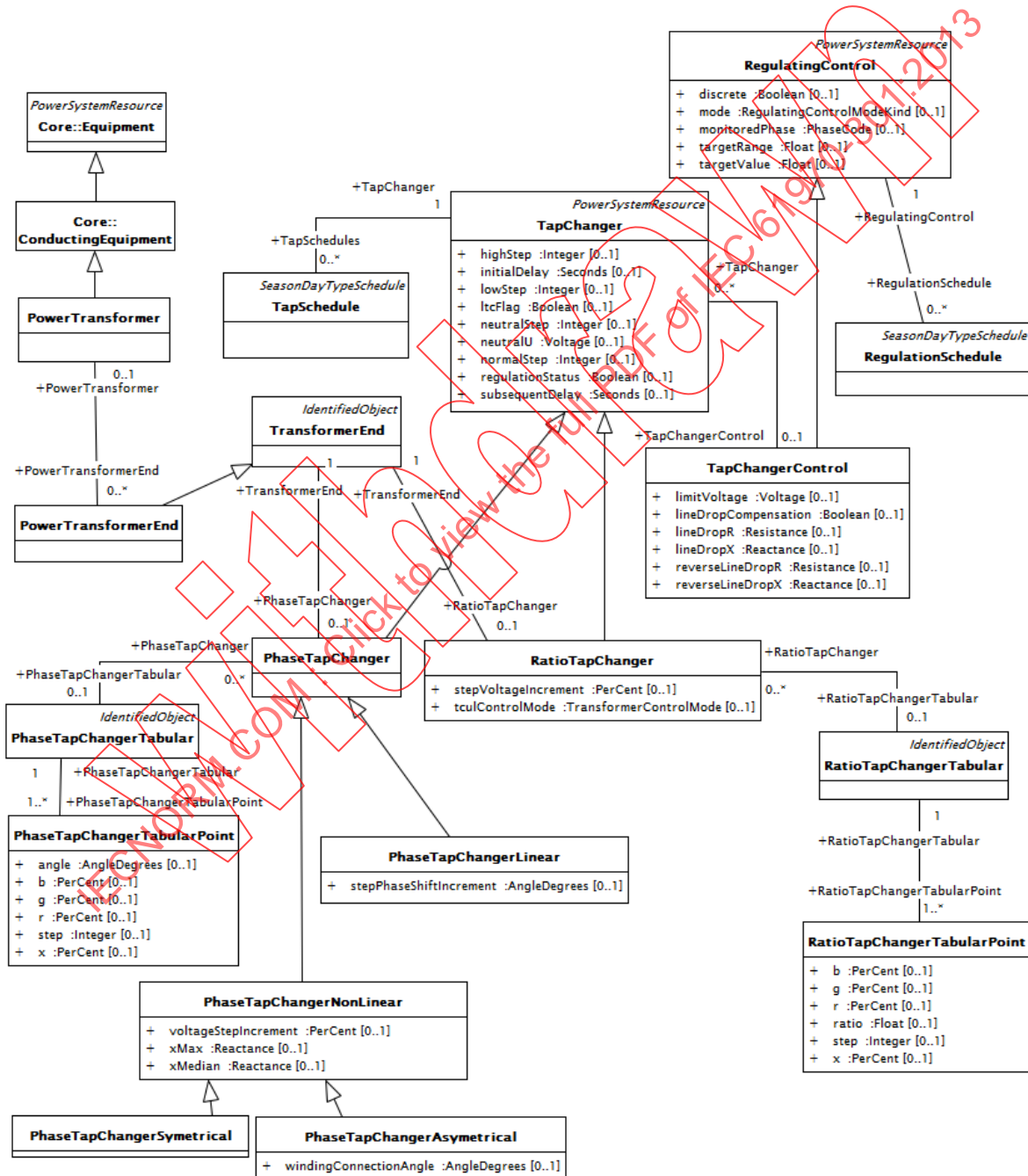


Figure 14 – Transformer tap model

Multiple tap changers can be combined within one power transformer, but it is expected that only one phase shifting and one ratio changing tap changer would be modeled on any given terminal. We document the ratio or phase shift change relative to the other terminal(s) assuming those terminals are at the nominal (1 PU) ratio and nominal (zero) phase angle shift. It is permitted to combine both a phase shifting and ratio tap changer on one terminal, though this practice is discouraged if the asymmetrical transformer model is an appropriate representation.

Tabular representations for both ratio and phase shifting transformers can be used where the available CIM models do not appropriately reflect the physical behaviour. The tabular models are optionally added to the other CIM tap models. It is recommended to use the linear models when adding tabular models. Tabular models may be used on non-linear phase shifter or ratio tap changer devices to describe the change in impedance through the tap range.

4.4.7.2 Voltage ratio transformer modeling

The RatioTapChanger class is used to define a transformer tap that changes the voltage magnitude ratio without changing the phase angle across the transformer. A RatioTapChanger may be used in cases where the transformer inherently introduces a fixed phase shift, such as from a wye-delta winding connection.

4.4.7.3 Voltage ratio transformer tabular modeling

A RatioTapChangerTabular may be optionally associated to any RatioTapChanger and represents a non-linear relationship between tap number, tap ratio, and impedances. Normally, if associated, the tabular ratio model takes precedence.

4.4.7.4 Linear phase shifter transformer modeling

A phase shifting transformer is linear if the phase shift increment per tap step is uniform over the full range of the tap changer. A linear phase shifter is modeled using the PhaseTapChangerLinear class.

4.4.7.5 Symmetrical phase shifter modeling

A symmetrical phase shifter construction is modeled with a PhaseTapChangerSymmetrical class and changes to voltage magnitude and phase over the full range of tap steps for a symmetrical phase shifter are shown in Figure 15. A symmetrical phase shifter will change the phase angle across the transformer but will not change the voltage ratio.

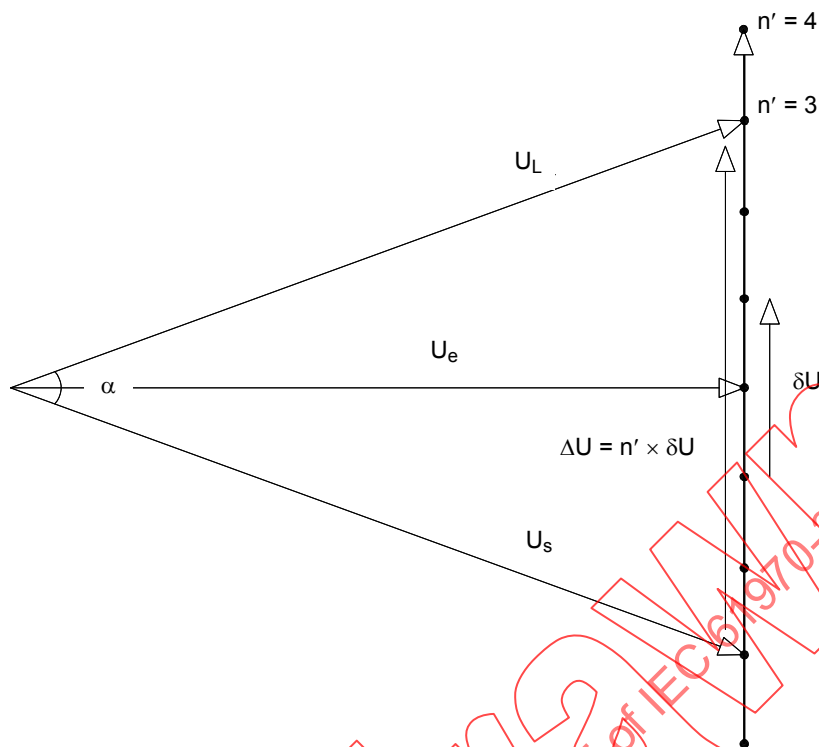


Figure 15 – Symmetrical Phase Shifter

The phase shift angle can be represented as:

$$\alpha = 2 \arctan \frac{\Delta U}{2U_e} = 2 \arctan \frac{n' \delta U}{2U_e}$$

$$\rho = 1$$

Where

- α is the phase shift angle, local terminal relative to other nominal terminal;
- n' is the current tap position;
- ΔU is the quadrature voltage change;
- U_e is the voltage at neutral tap;
- δU is the voltage change per tap position;
- ρ is the voltage magnitude transformation ratio, local terminal to other nominal terminal;
- U_L is the local terminal voltage;
- U_S is the system or other terminal voltage.

The sign of the $n'\delta U$ term is used to indicate the relative phase shift at the associated terminal. If $n'\delta U$ is positive, then the phase angle of the associated terminal is increased. If δU is negative then the phase angle of the associated terminal is decreased.

4.4.7.6 Asymmetrical phase shifter modeling

An asymmetrical phase shifter construction causes the voltage ratio to change as the phase shift changes. The changes to voltage magnitude and phase over the full range of tap steps for an asymmetrical phase shifter is shown in Figure 16 below.

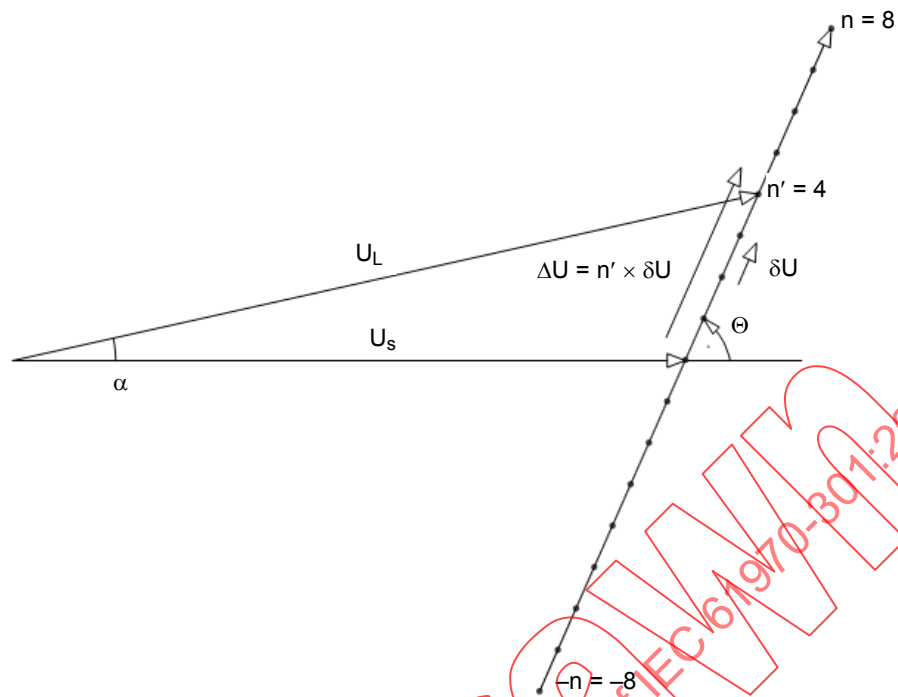


Figure 16 – Asymmetrical Phase Shifter

The phase shift angle can be represented as:

$$\alpha = \arctan\left(\frac{n' \delta U \sin \Theta}{U_s + n' \delta U \cos \Theta}\right)$$

$$\rho = \frac{U_s}{\sqrt{(n' \delta U \sin \Theta)^2 + (U_s + n' \delta U \cos \Theta)^2}}$$

Where

- α is the phase shift angle, local terminal relative to other nominal terminal;
- n' is the current tap position;
- δU is the voltage change per tap position;
- Θ is the winding connection angle;
- ρ is the voltage magnitude transformation ratio, local terminal to other nominal terminal;
- U_L is the local terminal voltage;
- U_s is the system or other terminal voltage.

An asymmetrical phase shifter is modeled with the PhaseTapChangerAsymmetrical class. The PhaseTapChanger.windingConnectionAngle shall be a multiple of 30 degrees: -90° , -60° , -30° , 30° , 60° , 90° , etc.). An asymmetrical phase shifter with a 90° winding connection angle is not the same as a symmetrical phase shifter.

4.4.7.7 Phase shifter transformer tabular modeling

A PhaseTapChangerTabular may be optionally associated to any PhaseTapChanger and represents a non-linear relationship between tap number, phase shift, and impedances. A PhaseTapChangerTabular may be used to describe only the impedance change over the range of the tapchanger. Under these circumstances, the phase shift attribute is null or omitted. Normally, if associated, the tabular phase shifter model takes precedence.

4.4.8 Phase wire modeling

The CIM models provide a means to specify equivalent single-line representation of balanced three phase networks or to additionally and optionally specify individual phase detail. In single phase networks there is a direct correspondence between Terminal objects and physical wire connections. In the single-line representation of balanced three phase networks we use one Terminal to represent the group of coordinating phase connections. Thus one Terminal can be used to specify one or more coordinated phase connections using the Terminal.phases attribute which is of enumeration type PhaseCode. PhaseCode has the possible phase connection combinations of ABC, ABCN, AB, BC, CA, A, B, C and so forth. The specification of phases at each terminal allows for modeling jumpers or switches that connect different phases. This phase connectivity specification is intended as a nominal or normal phase specification. The result of energization source is not reflected in this specification. This phase specification provides a means to locally connect like phases.

Figure 17 provides an example for specifying phase connections. This example shows a ConnectivityNode instance on the left with A, B, and C phases implied. Note that ConnectivityNode does not specify its phase content but rather implicitly connects like phases of terminals associated to the ConnectivityNode. The example shows a two phase jumper connecting on the left (Terminal.sequenceNumber=1) phases A and C and on the right (Terminal.sequenceNumber=2) phases A and B, with the internals of the jumper model SwitchPhase objects which specify in detail how the phase connections route. The Terminal.phases attribute only specifies the external connection of a device. Continuing to the right, two line segments are connected. The upper line segment is a two phase line on phases A and B. The lower line segment is single phase on phase B. The dark lines indicate CIM associations while the lighter arrows indicate the represented electrical connections at the phase level.

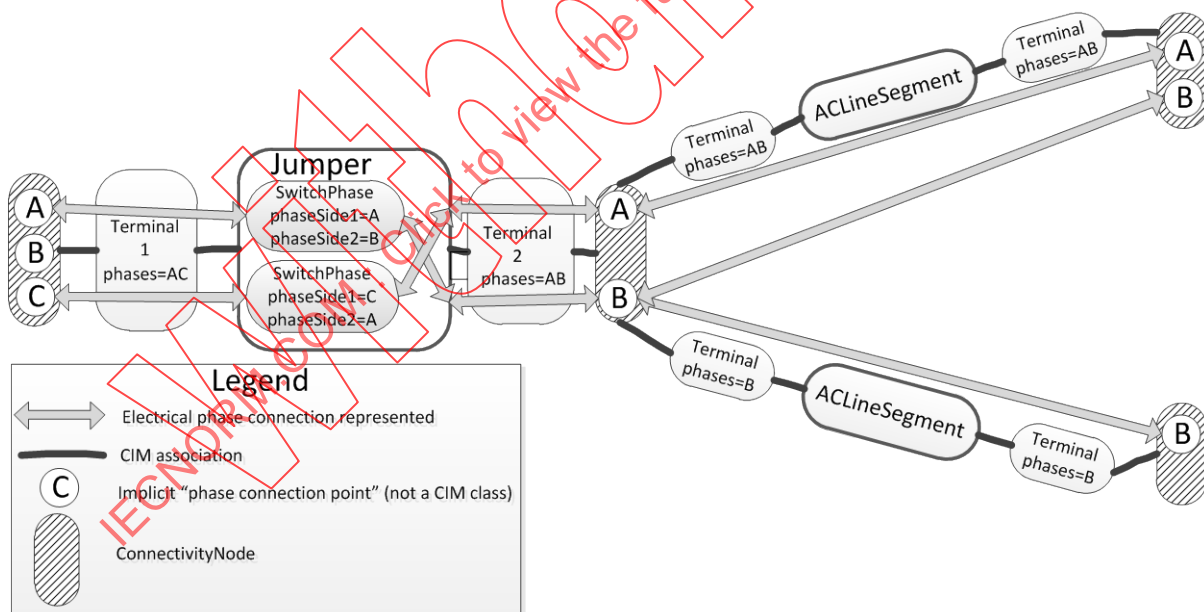


Figure 17 – Phase connectivity

As a clarification, the SwitchPhase attribute named phaseSide1 specifies any single phase (such as A, B, C, or N) and refers to the Terminal with sequenceNumber equal to 1. The SwitchPhase attribute named phaseSide2 refers to the Terminal with sequenceNumber equal to 2.

The switch model allows for cross phase connections to be modeled in general within the device. The transformer model also has explicit internal device modeling of phase connectivity. At this time all other ConductingEquipment branch specializations such as

ACLineSegment assume like phases connect at both terminals. The CIM model is open for future extensions to introduce internal device phase connectivity.

4.4.9 Cuts, clamps and jumpers model

The UML model for cuts and jumpers on a line segment and the additional connections are described in Figure 18. The Jumper class is a specialization of the more general Switch class and provides for phase crossings and phase connectivity as was illustrated in Figure 17. The jumpers therefore can connect terminals in the normal manner. The CIM classes Cut and Clamp provide a means to effectively add Terminals to an ACLineSegment at a specified distance down the line by associating new CIM objects to an ACLineSegment object, without other modification of ACLineSegment as-built model. Each Clamp instance introduces one new terminal connected at the specified distance down the line and without interruption of the line. Each Cut instance introduces two new terminals which are effectively the ends of the cut. By convention, each terminal of a cut is oriented toward the ACLineSegment terminal with the same sequenceNumber. If the cut is open, no flow passes through the ACLineSegment across the point of the cut, but flow can pass through either side of the ACLineSegment and through the appropriate cut terminal. Commonly, jumpers are connected to the ends of cuts, but they can also be connected to any type of conducting equipment in the normal manner using ConnectivityNode or TopologicalNode.

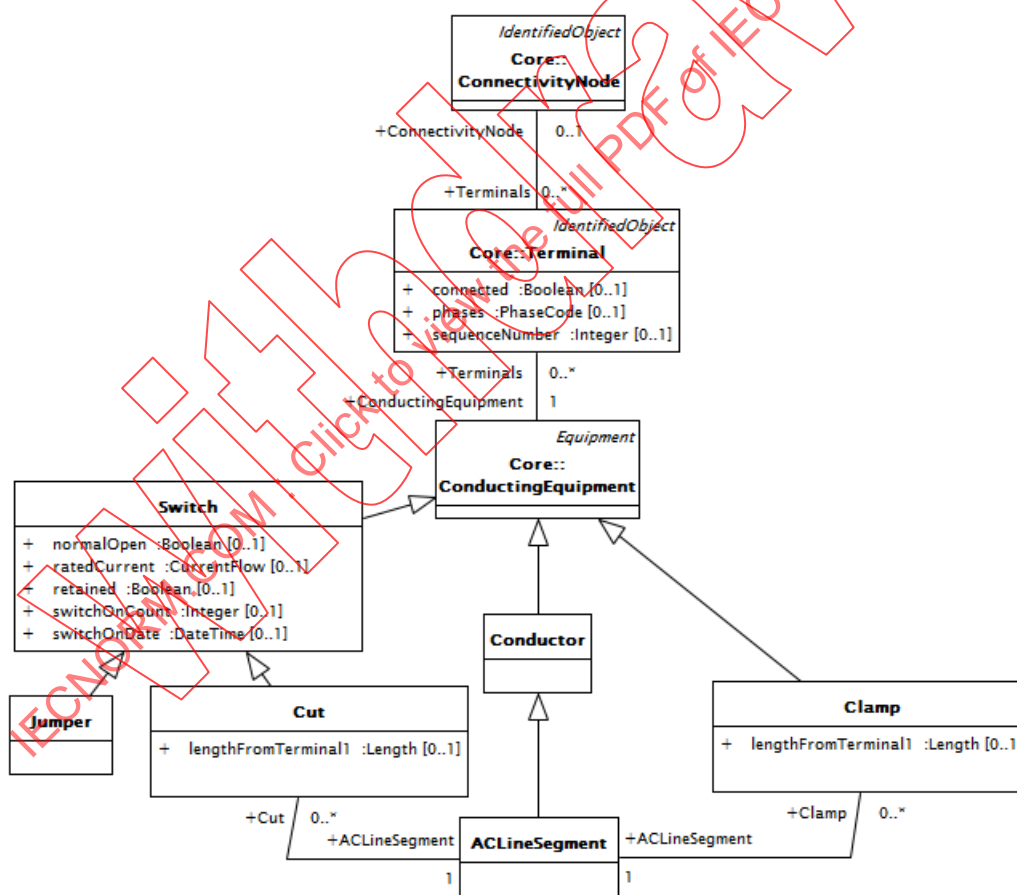


Figure 18 – Cuts, clamps, and jumpers UML model

An example of the addition of cuts and clamps to an existing CIM model is illustrated in Figure 19 and Figure 20. Figure 19 shows two physically parallel line segments named “FRED” and “LEROY”. In this example the system is energized from the right, the switch on the left is closed, and the loads name “ONE” and “TWO” are being supplied. Then the three faults are applied at the specified distances from Terminal 1 of the line respective line segments. In this example the phase detail is not specified for simplicity, but phase detail can follow the same approach outlined in 4.4.8.

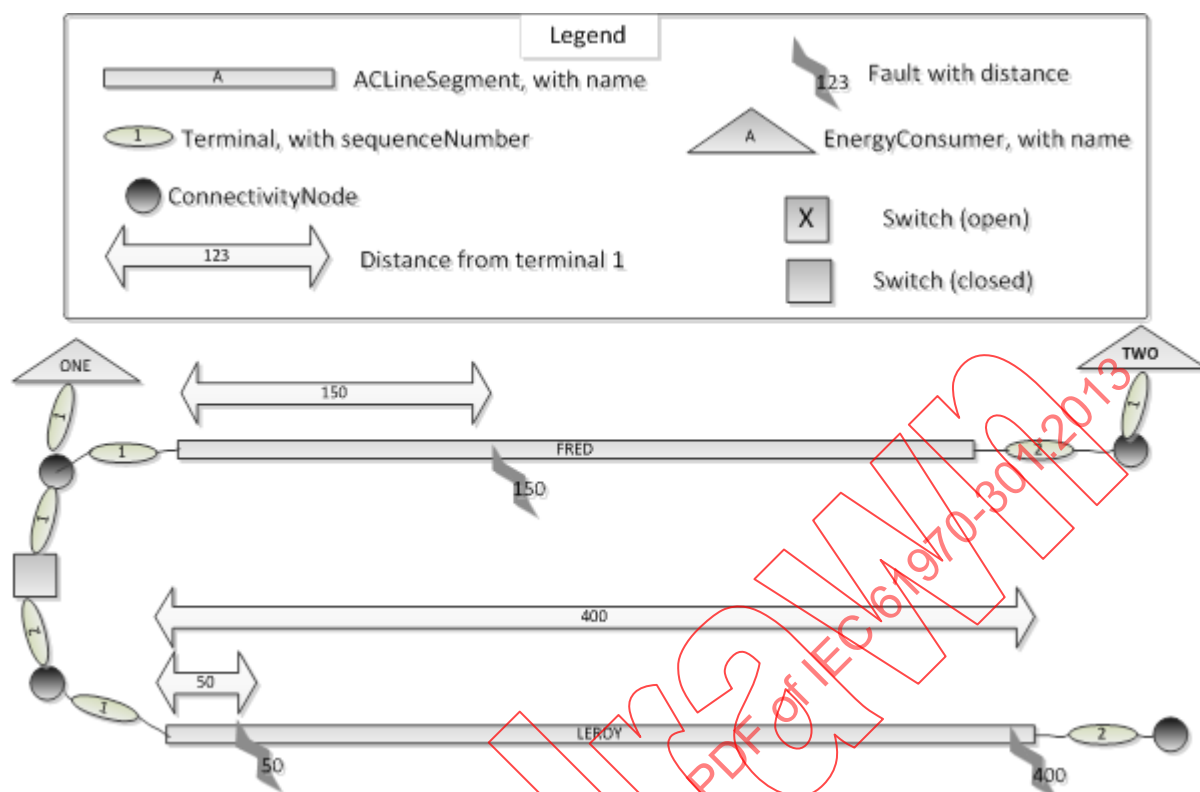


Figure 19 – Example before cuts and jumpers applied

Figure 20 shows how the pure addition of Cut, ConnectivityNode, Jumper, Clamp and potentially other Conducting Equipment can represent model changes reflecting typical temporary work. These same models can also be used for permanent model changes. The example shows how cuts applied at the specified distances can isolate the faults, leaving the faulted ends of cuts dangling, while the unfaulted network is connected to supply the loads. This example also demonstrates how multiple cuts can be applied to one line segment. This example shows open cuts, but if a cut were closed, flow could continue through the ACLineSegment with part of the flow potentially redirected to the terminals of the cut, much like the case for a clamp.

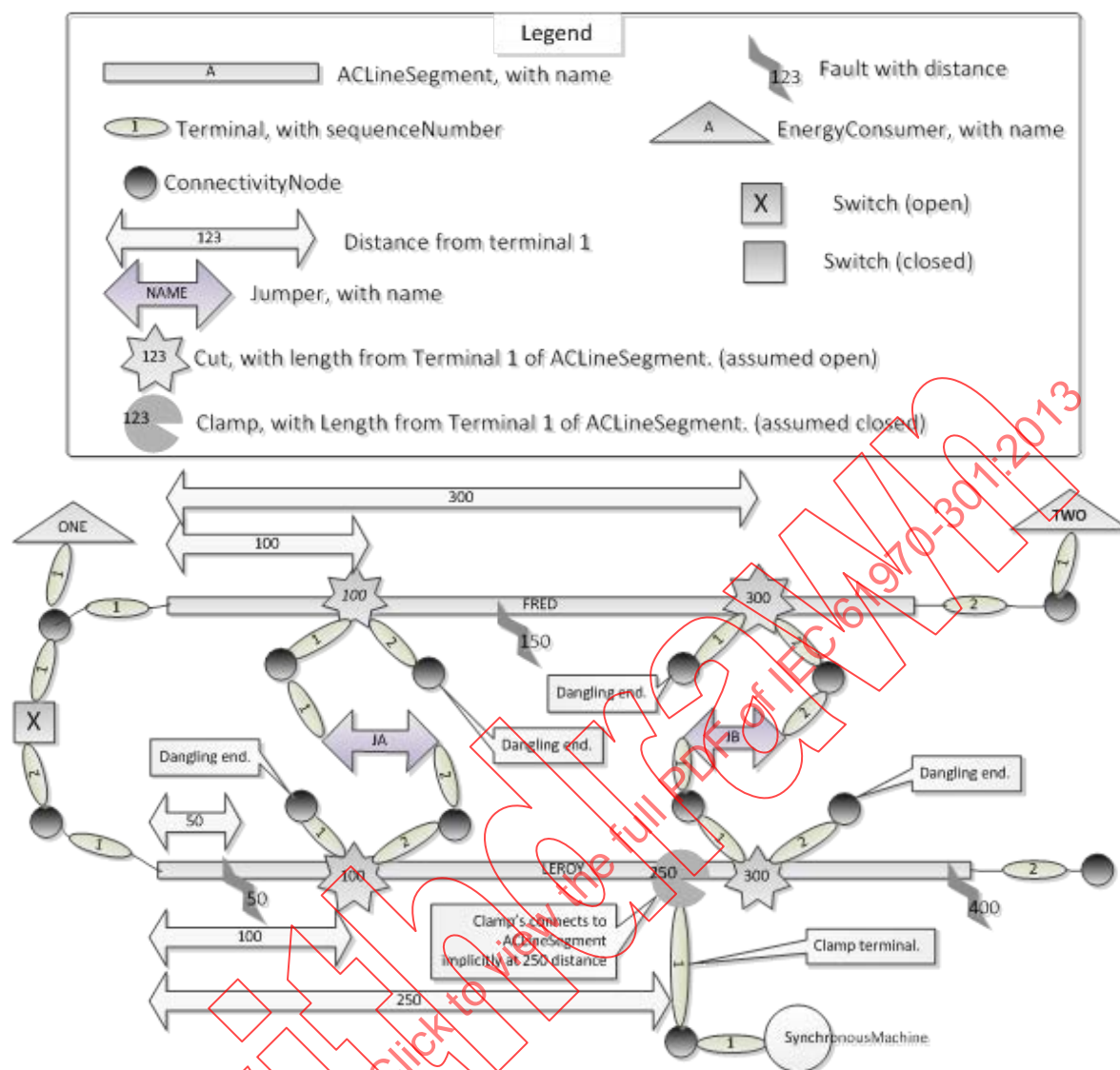


Figure 20 – Example after cuts and jumpers applied

A cut provides two terminals, and although not shown in the example above, connections could be made to either or both terminals of the cut. An example case might involve introducing a cut and connecting both cut terminals with a jumper that crosses phases.

A jumper may be used without clamps or cuts in a situation where existing terminals are to be connected. Figure 21 shows an example of using a jumper without a cut or clamp. In this case a temporary jumper might be used to bypass a switch under maintenance.

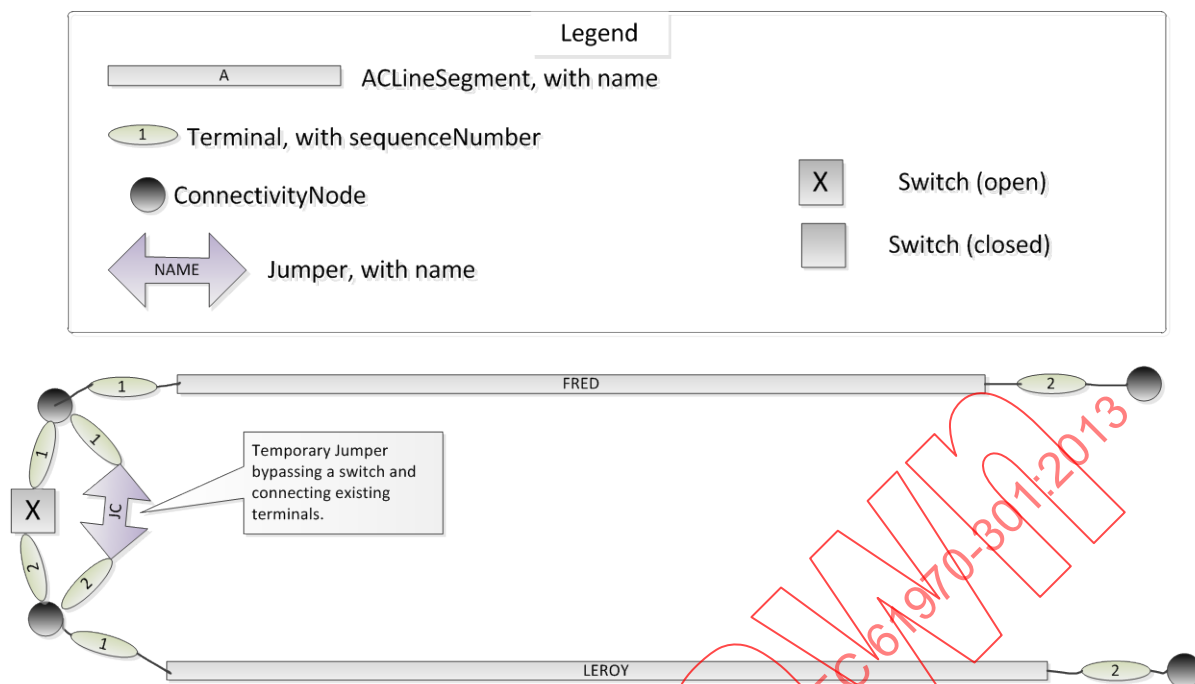


Figure 21 – Example of jumper without cut or clamp

4.4.10 Measurements and controls

4.4.10.1 Measurement overview

Measurements are used to represent the state variables that can be found in industrial processes. Each industrial process has its specific types of measurements. A power system typically has power flows, voltages, positions (e.g., breakers, isolators), fault indications (air pressure, oil pressure over temperature, etc.), counters (e.g., energy), etc.

The name "Measurement" would seem to indicate that all the state variables are measured. This is not always strictly the case as many measurements are calculated by SCADA or EMS/DMS functions, such as state estimator or power flow calculations. As a consequence, a measurement may have a number of alternate values (e.g., manually supplied, telemetered, state estimated, optimized etc.). This is supported by the Measurement and MeasurementValue models in the Meas package. The classes in the StateVariable package have been added to the CIM model specifically for the purpose of supporting exchange of values calculated by functions such as state estimator and power flow.

The measurement model now allows for optionally specifying the specific phases of a measurement.

4.4.10.2 Control overview

Controls are used to represent control variables. Power system control variables typically are set points, raise lower commands, select before execute commands and on/off commands. The Meas package supports control variables with the Control model.

4.4.10.3 Use of measurement-related classes

A PowerSystemResource (PSR) may have zero to many measurements associated with it by containing one or more measurements. Each measurement may have one or more measurement values. Observing the following guidelines will enable applications to navigate and find the required measurement values in a consistent way (see Figure 22):

- a) Measurements of a `PowerSystemResource` are classified by the attribute `measurementType`. The values to be used for `Measurement.measurementType` are specified in Table 2.
- b) `MeasurementValues` of a `Measurement` are classified by `MeasurementValueSource`.
- c) `MeasurementValueSource` inherits from `IdentifiedObject`. The values to be used for `MeasurementValueSource.name` are given in Table 3. This table provides a number of source names to be used where possible. However, the exact names to be used for specific applications are defined in related IEC 61970 Component Interface Specifications (CIS).
- d) The tables may be extended for proprietary needs. The names added shall start with a unique name (e.g. the company name) and an underscore. Example: `xyz_AverageTemperature`.
- e) The `ValueAliasSet` is used for discrete measurements and describes mappings from values to symbolic names, such as enumeration literals having specific integer values. Different communication protocols (e.g. for RTUs and for control centers as ICCP or ELCOM) use different data encodings. A system may have a system wide mapping for all Discrete values or group the Discrete values and make a mapping per group. Creation of a single system wide mapping that covers existing communication protocols is outside the scope of this specification.

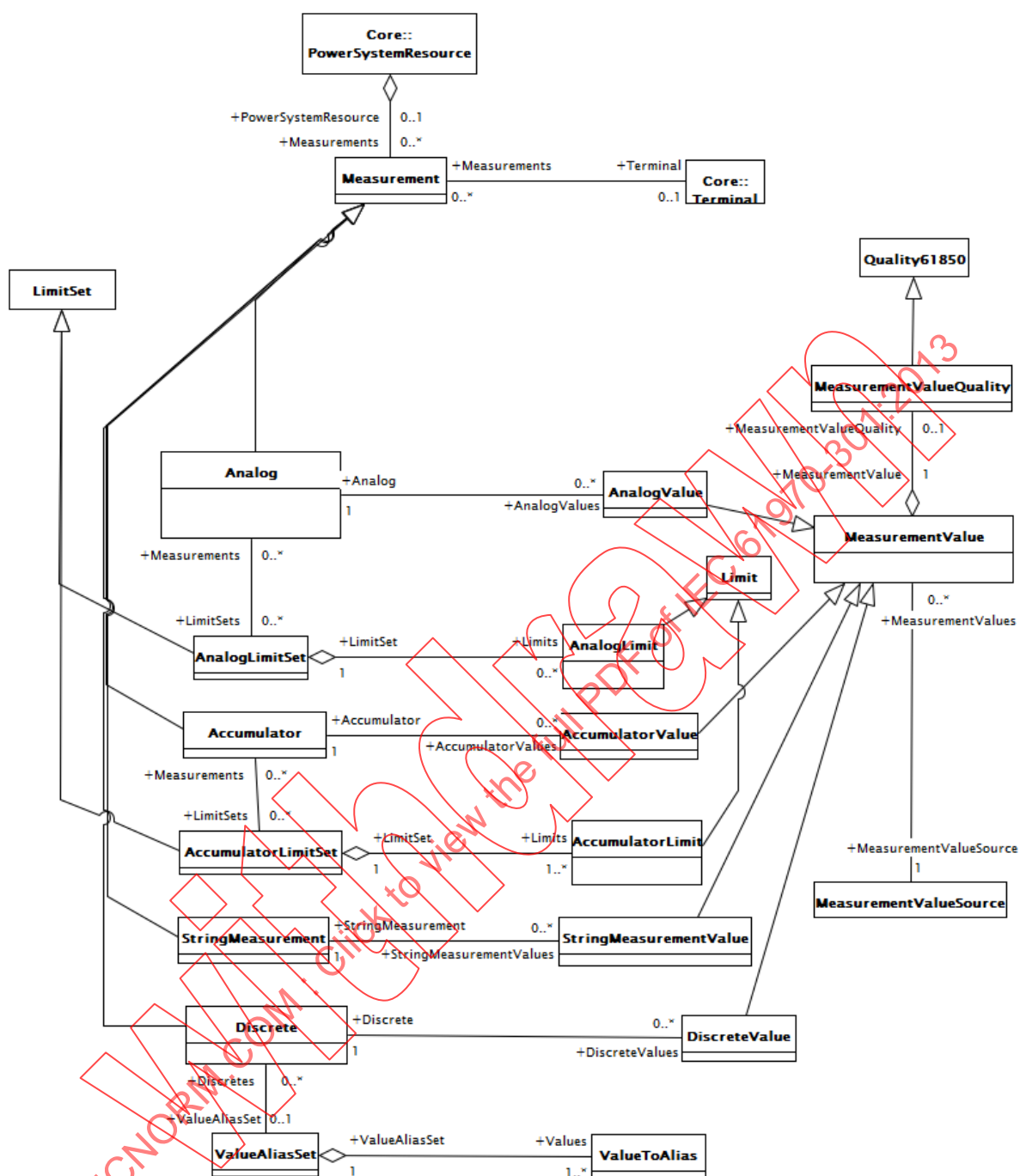


Figure 22 – Navigating from PSR to MeasurementValue

Table 2 – measurementType naming conventions

measurementType	61850 Name	description
Current	Amp	Current (r.m.s.) of a non-three phase circuit
ThreePhaseCurrent	AvAmps	Total current (r.m.s.) in a three phase circuit
PhaseCurrent	A	Measured phase current.
Frequency	Hz	Frequency
PowerFactor	PwrFact	Power Factor not allocated to a phase
ThreePhasePowerFactor	TotPF	Average power factor in a three phase circuit
ThreePhaseApparentPower	TotVA	Total apparent power in a three phase circuit
ThreePhaseReactivePower	TotVAr	Total reactive power in a three phase circuit
ThreePhaseActivePower	TotW	Total real power in a three phase circuit.
ApparentPower	VoltAmp	Apparent power in a non-three phase circuit
ReactivePower	VoltAmpr	Reactive power in a non-three phase circuit
Voltage	Vol	Voltage (r.m.s.) not allocated to a phase
ActivePower	Watt	Real power in a non-three phase circuit
Pressure	Pres	Pressure
Temperature	Tmp	Temperature
Angle	Ang	Angle between voltage and current
ApparentEnergy	TotVAh	Apparent energy
ReactiveEnergy	TotVArh	Reactive energy
ActiveEnergy	TotWh	Real energy
Automatic	Auto	Automatic operation (not manual)
LocalOperation	Loc	Local operation (not remote)
SwitchPosition	Pos	Switch position [2bits= intermediate,open,closed,ignore]
TapPosition	TapPos	Tap position of power transformer or phaseshifter
Operation Count	OperCnt	Operation count – typically for switches
LineToNeutralVoltage		Line to neutral voltage.
LineToGroundVoltage		Line to ground voltage.

Table 2 describes various types of measurements also defined in IEC 61850. The meanings of the columns in Table 2 are as follows:

- measurementType (Measurement.measurementType) is the IEC 61970 measurement type name.
- 61850 Name is the name assigned to the data object in IEC 61850. (Refer to Clause 6 of IEC 61850-7-4:2010, data object name semantics).
- description of the data.

It shall be noted that Table 2 is not an exhaustive list and that the mapping between measurements as defined in a control center and a substation is non-trivial.

Table 3 – MeasurementValueSource naming conventions

name	description
SCADA	Telemetered values received from a local SCADA system
CCLink	Value received from a remote control center via TASE.2 or other control centre protocol
Operator	Operator entered value (always manually maintained, PSR is not connected to an RTU)
Estimated	Value updated by a state estimator
PowerFlow	Value updated as result of a Powerflow
Forecasted	Value that is planned or forecasted.
Calculated	Calculated from other measurement values (e.g., a sum)
Allocated	Calculated by a load allocator

Following these conventions:

- each Measurement instance represents a technological quantity of a PowerSystemResource;
- each MeasurementValue of a Measurement represents a value for the technological quantity, as supplied from a single source;
- the source attribute in MeasurementValueQuality then indicates whether the source actually provided the current value, or whether it had been substituted or defaulted.

Note that a new MeasurementValue identity is not normally created for each exchange of a measured value. It is expected that the same MeasurementValue instance could exchange new values in subsequent messages. Therefore the identity of a MeasurementValue instance can be established apriori to exchange of measured values. The modeling of a time series of values is not explicitly expressed in the CIM model, though the model allows for systems to internally build such time series models through a series of measurement value exchanges.

4.4.10.4 Attachment of measurements

As mentioned in the previous subclause and as shown in Figure 22, Measurements are contained by a PowerSystemResource. This is sufficient for Measurements that are not related to connectivity, e.g. temperature, weight, size.

To specify the location of a Measurement in the network, an association to Terminal is used. Examples include power flows, voltages, and currents. Voltages have no direction and can be attached wherever appropriate in relation to the sensor placement. Flows have direction and shall be attached such the flow direction is evident from the placement.

Figure 23 shows two examples of the placement of Measurements.

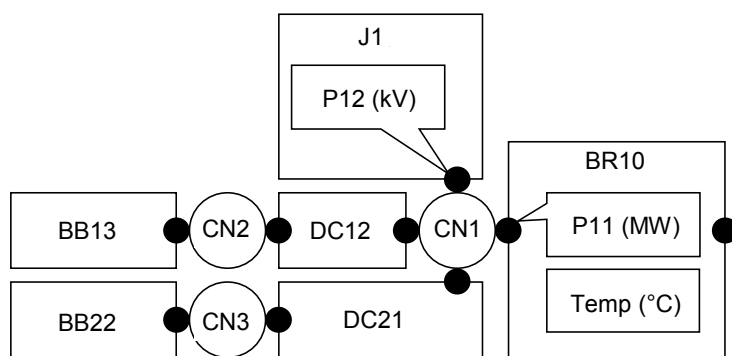


Figure 23 – Measurement placement

P12 is a voltage Measurement that measures the voltage at the Junction J1. P12 is topologically related to the ConnectivityNode CN1 via the Terminal in Junction J1. P11 is a Measurement that measures the flow through Breaker BR10 at the side connected to the ConnectivityNode CN1. P11 is topologically related to the ConnectivityNode CN1 via the left Terminal in Breaker BR10. Temp is a Measurement that measures the Breaker temperature. As a temperature is not related to connectivity, it has no relation to a Terminal – it just belongs to the Breaker BR10.

4.4.10.5 ICCP measurements

ICCP (known officially as IEC 60870-6 TASE.2) data sources are specified using the MeasurementValue classes (and its specializations) and the MeasurementValueSource class. The MeasurementValueSource class is used to define the control center supplying the ICCP data. The MeasurementValueSource shall be associated with an instance of Name where the attribute Name.name holds the name of the supplying control center. The instance of NameType associated with the control center Name shall have the NameType.name attribute set to "ICCP".

The MeasurementValue classes are used to specify the ICCP ID. The MeasurementValue shall be associated with an instance of Name where the attribute Name.name holds the ICCP ID. The instance of NameType associated with the ICCP ID Name shall have the NameType.name attribute set to "ICCP". The MeasurementValue.name attribute holds the SCADA point name. Each MeasurementValue will be associated with one Measurement. Each MeasurementValue being supplied via ICCP shall also have an association to a MeasurementValueSource.

4.4.11 Regulating control models

Regulation control, such as automatic voltage control at generators or voltage tap control at transformers is modeled using the RegulatingControl class which provides the capability to model multiple instances of equipment participating in a regulation scheme. These regulation schemes may be physical or manually implemented in actual power system operation, but are reflected in the models used for power system analysis purposes. Figure 24 shows how RegulatingCondEq and TapChanger classes can participate in regulation.

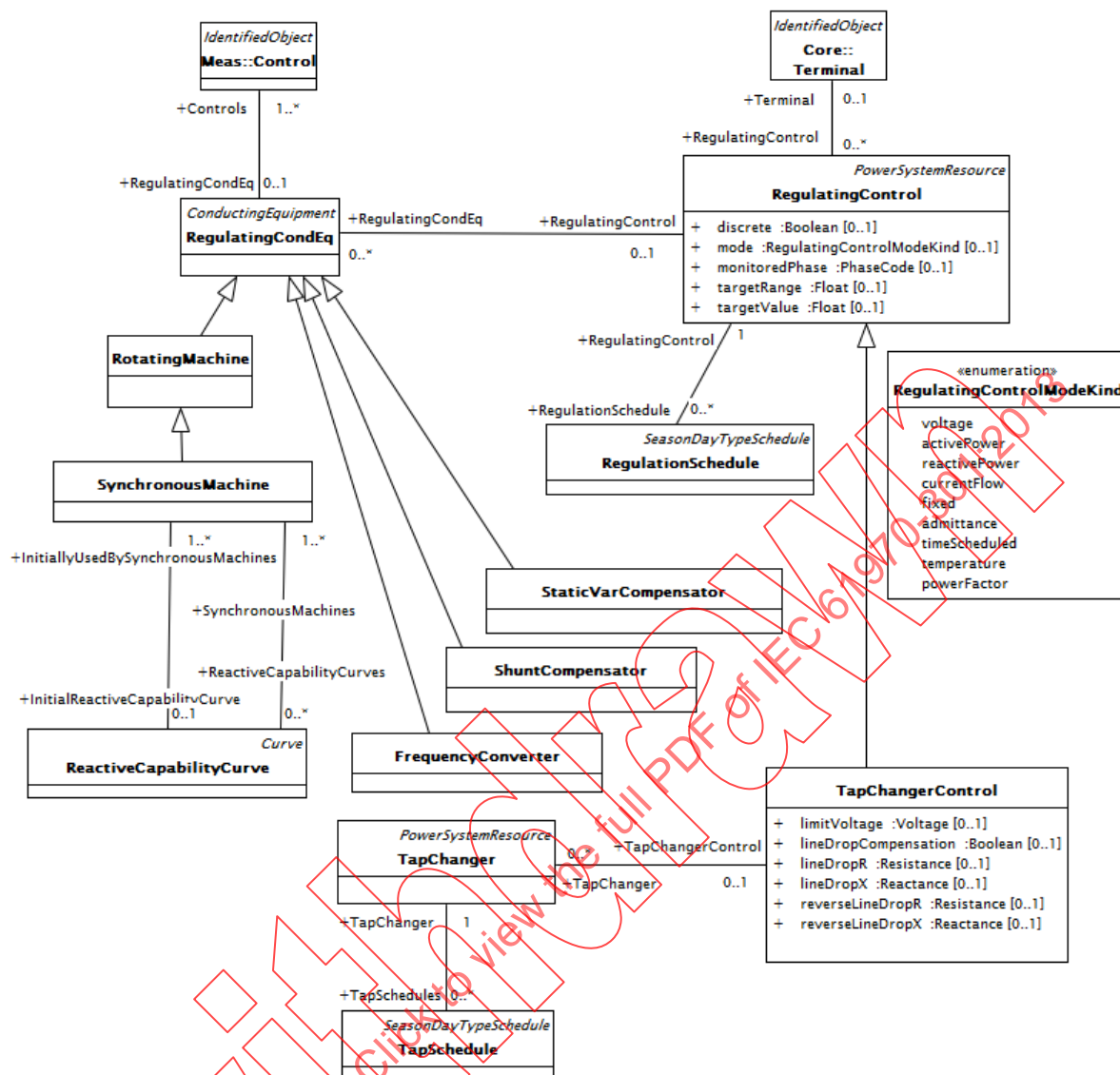


Figure 24 – Regulating control models

4.5 Modeling guidelines

4.5.1 Modeling for change

The subclauses 4.5.2 to 4.5.5 provide guidelines on how to maintain and extend the CIM.

The CIM is meant to contain classes and attributes that will be exchanged over public interfaces between major applications. The goal is to keep, as much as possible, only the generic features from which a detailed implementation may be derived. In general, it is easier to change the value or domain of an attribute than to change a class definition. This makes the model more robust because it is able to support a broader class of requirements, and more stable because new requirements may be able to be handled without requiring changes to the model.

4.5.2 Process for amendments to the CIM

It may be desirable to amend the CIM to either revise the existing model or to extend the CIM to model additional elements of an electric utility power system. The recommended process for such amendments is as follows.

- a) Prepare use case(s) and a request for change to describe the desired changes. This should include proposed changes to the appropriate class diagrams showing new/revised classes, attributes, and associations.
- b) The use case(s) and request for change are then reviewed by the appropriate IEC Working Group to decide if the requested changes should be treated as revisions to the current CIM standard, or if they should be treated as private amendments, not requiring a change to the standard itself.
- c) Proposed amendments accepted by the working group will be added to a list of outstanding issues, and at the appropriate time, a new version of the CIM model will be prepared and an update made to the appropriate IEC CIM specification.

4.5.3 Changes to the CIM UML model

From a modeling perspective, when the CIM is extended, the approach is to start with the existing CIM UML model. The extensions may be added in any of several ways that are available in UML, but in all cases the approach is to inspect the current model and determine the best way to build off of the existing class diagrams. The extensions may take the form of any of the following, starting from the simplest to the most complex:

- adding additional attributes to existing classes;
- adding new classes that are specializations of existing classes;
- adding new classes via associations with existing classes.

For the purpose of extending this standard, the main objective is to reuse the existing CIM to the maximum extent possible. From a packaging point of view, extensions should be made to existing packages where possible. If the extensions comprise a new domain of application, then consideration should be given to creating a new package for the additions, but still creating the necessary associations to the existing package, keeping in mind that even though a new package is being created, the CIM is still a single model.

For the purpose of creating non-standard changes to the information model, the standard CIM should be extended without modifying the standard CIM. Preferably, such extensions should be placed in new packages to both identify the changes as non-standard and to facilitate transferring changes to an updated version of the standard.

4.5.4 Changes to the CIM standards documents

From a documentation perspective, when the CIM is extended, a decision shall be made whether the changes constitute updates to existing CIM standards documents, or whether a new IEC 61970-3x specification is required. In either case, the extensions will then become part of the IEC standard CIM.

4.5.5 Deprecations

The CIM model may at time contain packages, classes, attributes, or associations that have been identified as deprecated. These items will be noted in the documentation or with a UML stereotype of “deprecated”. A deprecated item is retained in the present version of the model, but is expected to be removed from future versions. An item that has not been deprecated is not guaranteed to be retained in future versions, but using an item marked as deprecated should be avoided if possible.

4.5.6 CIM profiles

An implementation of the CIM need not include all classes, attributes, or associations in the standard CIM specification to be compliant with the CIM standard. Profiles may be defined to specify which elements shall be included (i.e., mandatory elements) in a particular use of the CIM, as well as which are optional. These profiles are defined in the profile documents of IEC 61970 series of standards, as well as in other TC57 series of standards that use and extend the base CIM specified in this document.

An example is the profile for exchanging power system models. This profile specifies how the CIM is to be used for exchanging power system models in XML, and also specifies the mandatory and optional classes, attributes, and associations to be supported for this use of the CIM. Profiles can be maintained using a variety of tools such as the open source CIMTool, available at <http://www.CIMTool.org>.

4.6 Modeling tools

This CIM release was constructed using Sparx Systems Enterprise Architect product. The entire CIM UML model exists as an Enterprise Architect¹ project file and is viewable with that tool, including the class diagrams and descriptions of classes, attributes, types, and relationships. Viewing the CIM in this fashion provides a graphical navigation interface that permits all CIM specification data to be viewed via point-and-click from the class diagram in each package. A free viewer is available through Sparx Systems.

Ideally, the CIM information model is independent of any specific UML tool, though experience has shown that exchanges between different tools are often less than perfect. Until tool interoperability is proven effective, future changes to the CIM specification, resulting in new versions of this standard, will be incorporated first into the Enterprise Architect project description to ensure a single source for the CIM model data.

Clause 6 of this standard was auto-generated using the open source tool jCleanCim, available at <http://cimug.ucaiug.org/MTEI/Shared%20Documents/jCleanCim>, which uses the automation interfaces of Microsoft Word^{TM2} and Enterprise Architect and the standard IEC format and styles.

4.7 User implementation conventions

4.7.1 Conventions beyond UML

The following subclauses provide recommended user conventions when using the CIM in actual system implementations.

4.7.2 Number of Terminals for ConductingEquipment objects

The following ConductingEquipment classes have two terminals: ACLineSegment, DCLineSegment, Switch and all its specializations (including Jumper, Fuse, Breaker, Disconnecter, LoadBreakSwitch, and Cut), SeriesCompensator, and EquivalentBranch. The PowerTransformer class typically has two terminals, but may also have one or more terminals. For example a zig-zag connected grounding transformer may have one terminal. Three terminal transformers are commonly used in transmission systems and in special cases transformers may have four, five, or more terminals. All other ConductingEquipment leaf classes (notably also including Clamp and BusbarSection) have a single terminal.

4.8 CIM modeling examples

Power system models have been created from the CIM UML model in various ways. The first example is an RDF (Resource Description Framework) Schema version of the CIM, which uses XML (eXtensible Markup Language) to describe a power system network model. IEC 61970-501 and other profile standards in development are used to specify the model exchange format. RDF Schema versions of the CIM have been used to create XML model files of actual networks for purposes of interoperability testing. An RDF Schema version of the

¹ Enterprise Architect is the [trade name or trademark] of a product supplied by Sparx Systems. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

² Word TM is the [trade name or trademark] of a product supplied by Microsoft. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

CIM is generated from the CIM UML model file using software tools based on the RDF Schema specification of the CIM.

It should be noted that an RDF schema version of the CIM is still metadata rather than an instantiation of an actual network. However, complete network model files with descriptions of all network elements and their electrical connectivity can be generated by system suppliers using proprietary export tools, and then imported by other systems via a similar import tool, which is used to populate a local network engineering tool database. Examples of such CIM XML models include the Siemens 100 bus model, the Alstom 60 bus model, and the ABB 40 bus model files³⁾ used for CIM XML interoperability testing.

5 Detailed model

5.1 Overview

The portion of the common information model (CIM) covered by la présente norme represents a comprehensive logical view of information exchanged among different systems in electrical utilities. This definition includes the public classes and attributes, as well as the relationships between them. The following Subclauses describe how Clause 6 is structured. Clause 6 is automatically generated from the CIM model maintained with the tools described in 4.6.

5.2 Context

The CIM is partitioned into subpackages. Classes within the packages are listed in the order they are defined in the UML model. Native class attributes are listed first, followed by inherited attributes in order of depth of inheritance. Native associations are listed first for each class, followed by inherited associations in order of depth of inheritance, then alphabetically by association end name. The associations are described according to the role of each class participating in the association. The association ends are listed under the class at each end of an association.

Figure 25 shows the top level packages included in this standard.

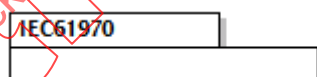


Figure 25 – CIM top level packages

For each package, the model information for each class is fully described. Attribute and association end information for native and inherited attributes is listed as in Table 4 and Table 5 for an example class named “Class1” in example package “Package1”. For any inherited attributes or association ends the “description” column will contain text indicating the attributes is inherited from a specific class. In this example we assume Class1 inherits from class IdentifiedObject. The description column for native attributes and association ends contains the actual description. For inherited attributes, the base class is noted in the description and the full attribute documentation is available in that base class as a native attribute.

3) Siemens100 bus model, Alstom 60 bus model, and ABB 40 model are the trade names of products supplied by Siemens, Alstom Grid, and ABB. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

Table 4 – Attributes of Package1::Class1

name	type	description
native1	Float	A floating point native attribute of the class is described here.
native2	ActivePower	Documentation for another native attribute of type ActivePower.
name	String	Inherited from: IdentifiedObject

For cases where an attribute is a constant, Table 4 will add the phrase “(const)” to the name column. In such cases, the attribute normally has an initial value also which is preceded by an equal sign and appended to the attribute name.

Table 5 – Association ends of Package1::Class1 with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] Class2	Class2	The description of the association end to another class named “Class1”. Note it is typical, but not required, that the association end is the same as the referenced type. This association end may reference zero or one objects. The other end of the association may reference any number of Class1 instances.
[0..1]	[0..*] SecondClass2	Class2	The description of references to class named “Class2”. In this case any number of references may be made. The reverse reference may reference zero or one Class1 instances.
[1..1]	[0..*] Class3	Class3	References any number of Class3 instances. The reverse reference shall reference only one instance of Class1.
[0..*]	[0..*] Class4	Class4	A many to many association.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

In Table 5, the first column describes the multiplicity at this end of the association (i.e., how this class participates in association). Second column describes the other association end. Its multiplicity is included in brackets. The association end name is listed in plain text. The class at the other end of the association is in the third column. A multiplicity of zero indicates an optional association. A multiplicity of “*” indicates any number is allowed. For example, a multiplicity of [1..*] indicates a range from 1 to any larger number is allowed.

In the case that a class is an enumeration named “Enumeration1”, the attributes table is replaced by the literals documentation as shown in Table 6, since enumeration literals have no type. There are no inherited enumeration literals for an enumeration class.

Table 6 – Literals of Package1::Enumeration1

literal	description
native1	This is the first native enumeration value.
native2	This is the second native enumeration value.
native3	There are typically no inherited attributes for enumerations.

6 Top package IEC61970

6.1 General

Top package for IEC 61970.

Figure 26 shows class diagram IEC61970Dependencies.

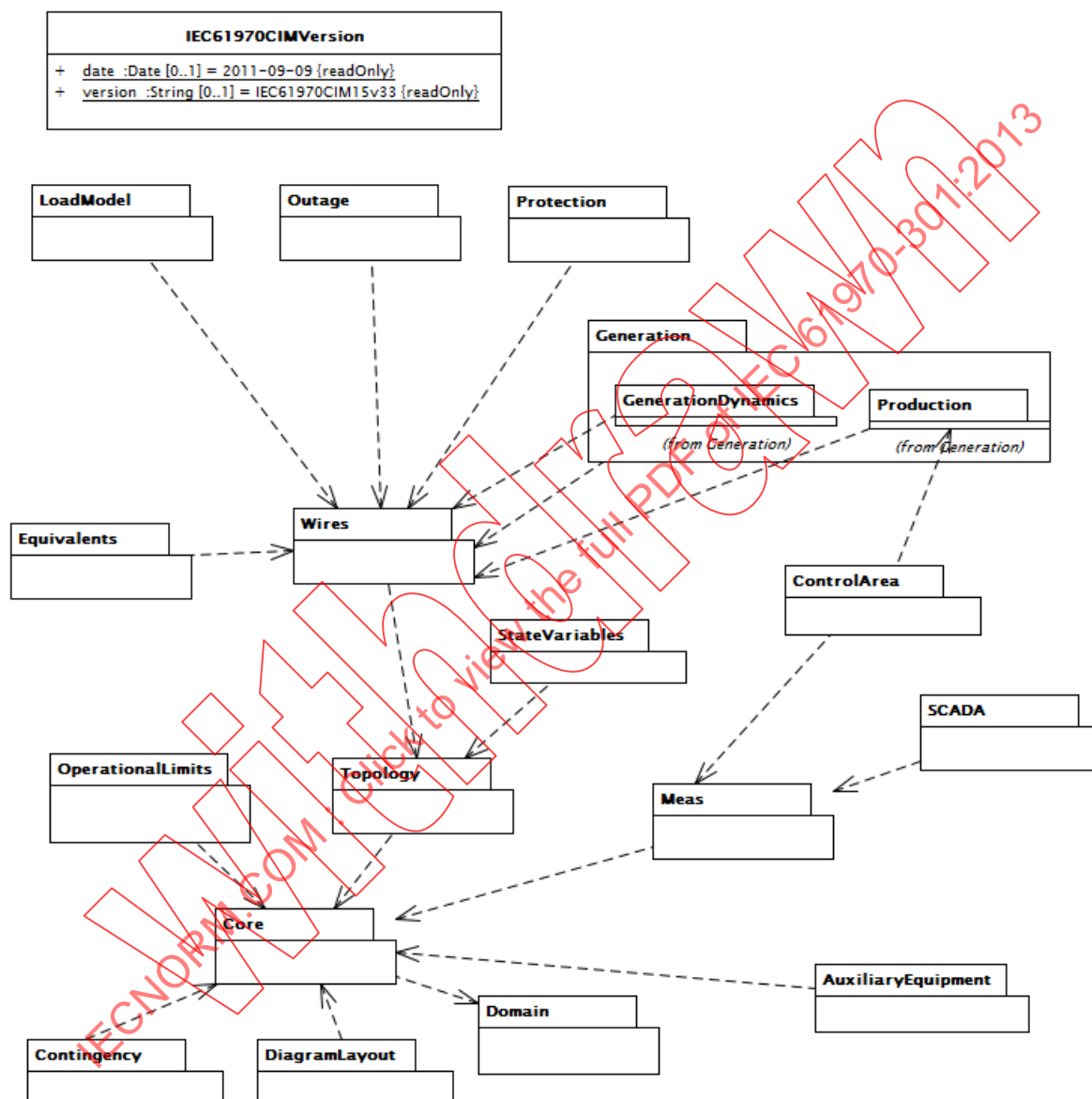


Figure 26 – Class diagram IEC61970::IEC61970Dependencies

This diagram shows all 61970 packages and their logical dependencies.

6.2 IEC61970CIMVersion root class

This is the IEC 61970 CIM version number assigned to this UML model.

Table 7 shows all attributes of IEC61970CIMVersion.

Table 7 – Attributes of IEC61970::IEC61970CIMVersion

name	type	description
date=2011-09-09 (const)	Date	Form is YYYY-MM-DD for example for January 5, 2009 it is 2009-01-05.
version=IEC61970CIM15v33 (const)	String	Form is IEC61970CIMXXvYY where XX is the major CIM package version and the YY is the minor version. For example IEC61970CIM13v18.

6.3 Package Domain

6.3.1 General

The domain package defines primitive datatypes that are used by classes in other packages. Stereotypes are used to describe the datatypes. The following stereotypes are defined:

<<enumeration>> A list of permissible constant values.

<<Primitive>> The most basic data types used to compose all other data types.

<<CIMDatatype>> A datatype that contains a value attribute, an optional unit of measure and a unit multiplier. The unit and multiplier may be specified as a static variable initialized to the allowed value.

<<Compound>> A composite of Primitive, enumeration, CIMDatatype or other Compound classes, as long as the Compound classes do not recurse.

Figure 27 shows class diagram BasicDatatypes.

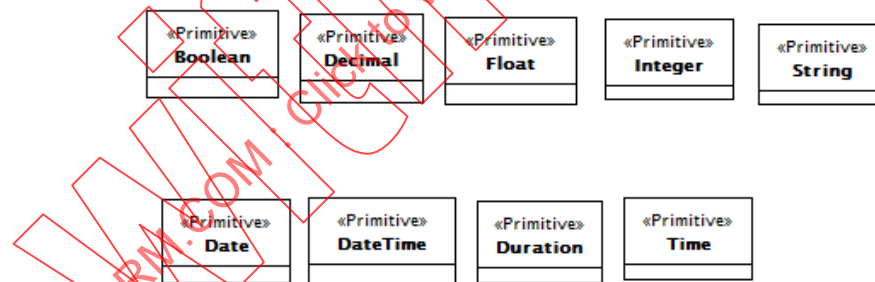


Figure 27 – Class diagram Domain::BasicDatatypes

This diagram shows basic datatypes that are the basis for all other datatypes.

Figure 28 shows class diagram ElectricityDatatypes.

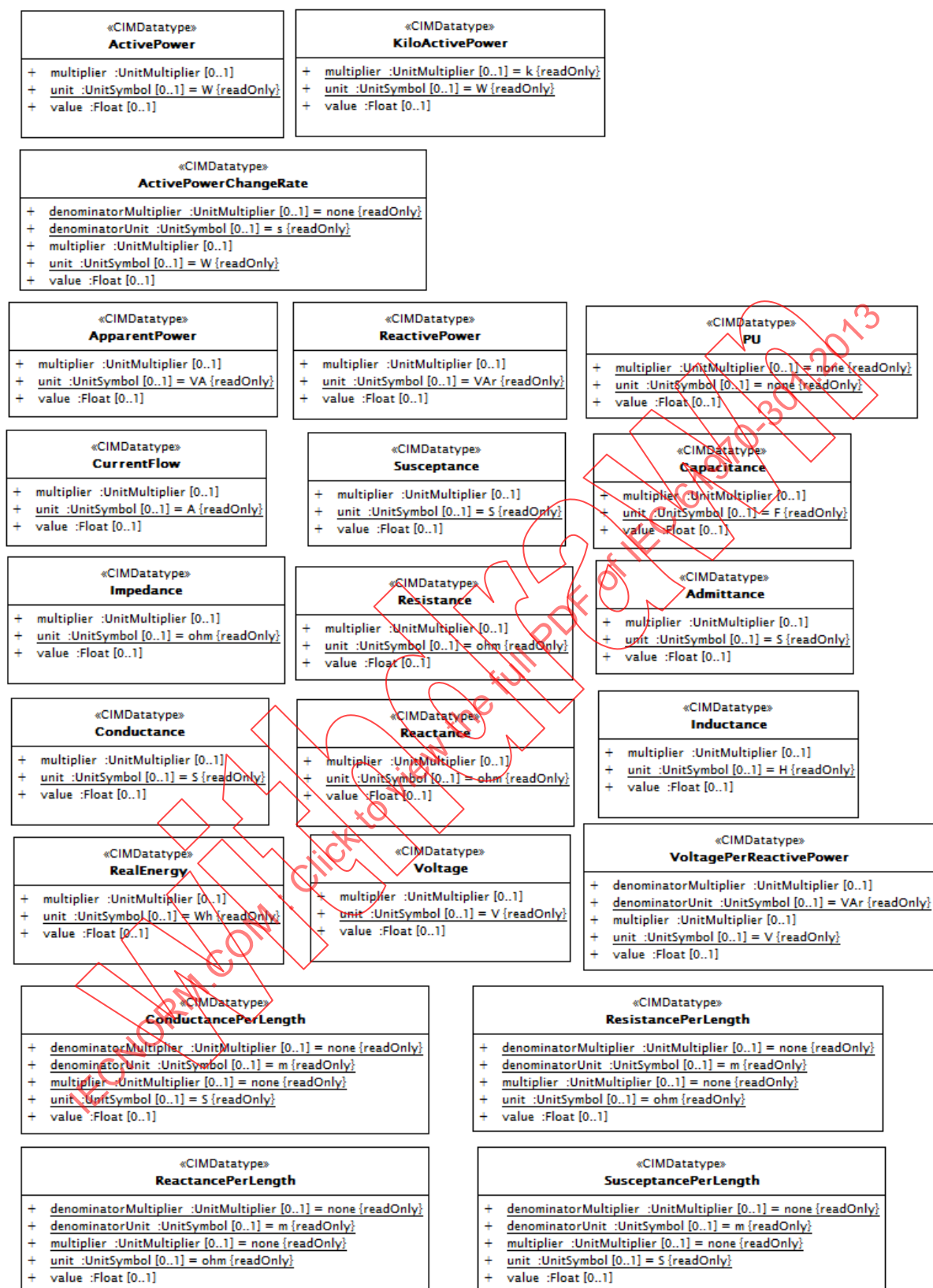


Figure 28 – Class diagram Domain::ElectricityDatatypes

This diagram shows electricity related data types.

Figure 29 shows class diagram EnumeratedUnitDatatypes.

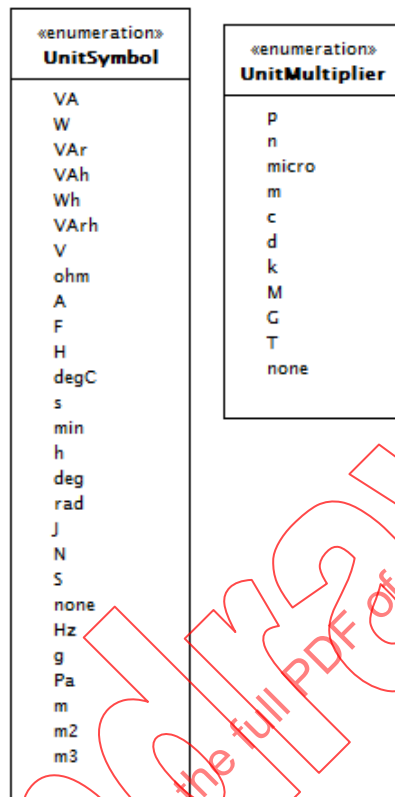


Figure 29 – Class diagram Domain::EnumeratedUnitDatatypes

This diagram shows enumerated data types related to the SI unit system.

Figure 30 shows class diagram GeneralDatatypes.

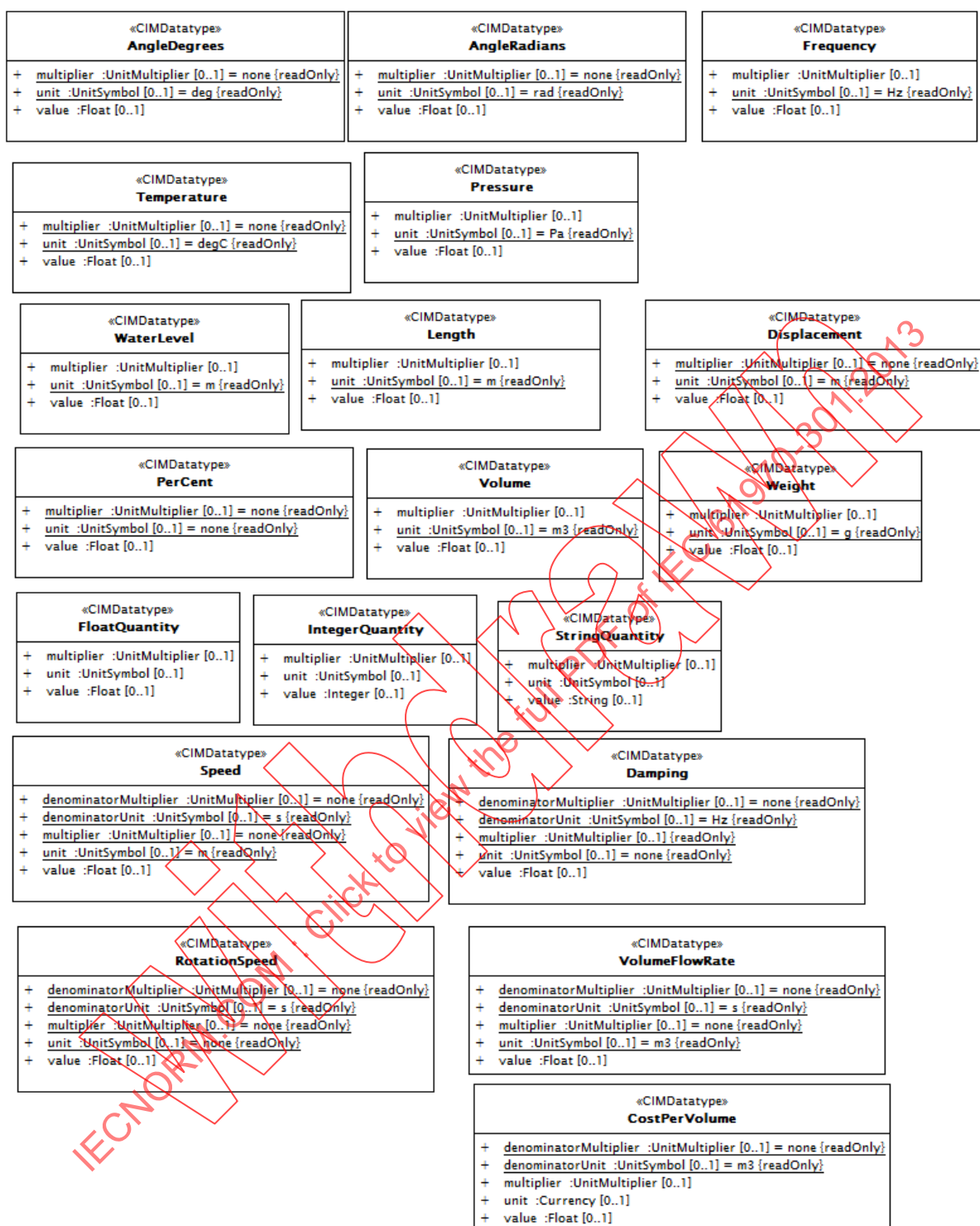


Figure 30 – Class diagram Domain::GeneralDatatypes

This diagram shows general data types.

Figure 31 shows class diagram MonetaryDatatypes.

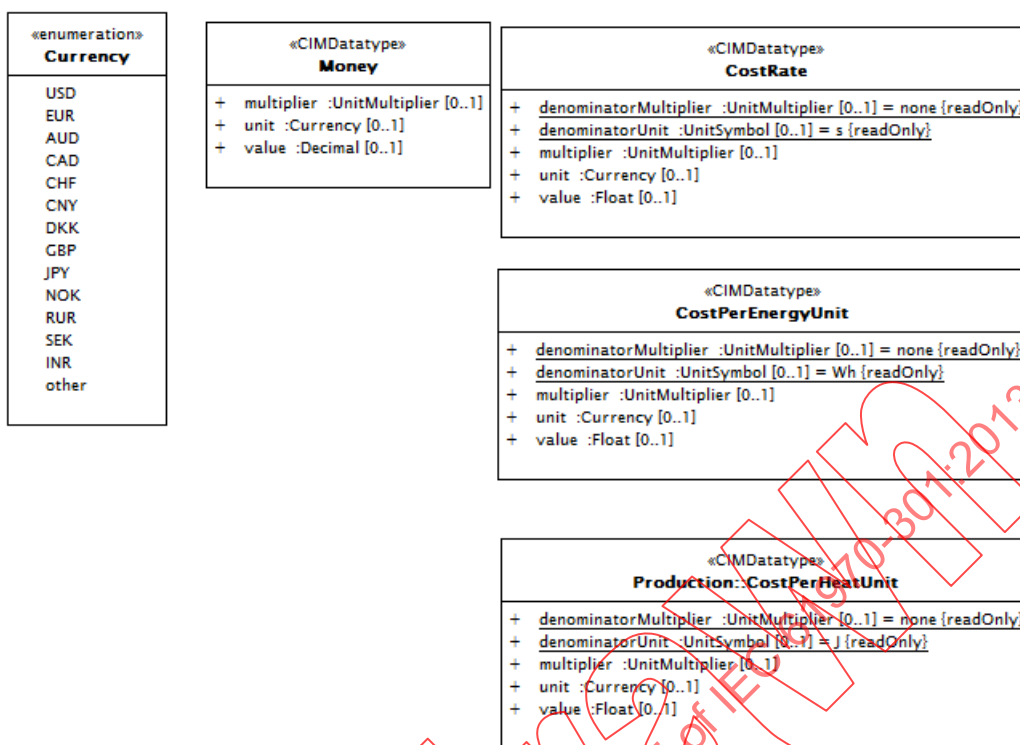


Figure 31 – Class diagram Domain::MonetaryDatatypes

This diagram shows monetary related data types.

Figure 32 shows class diagram TimeDatatypes.

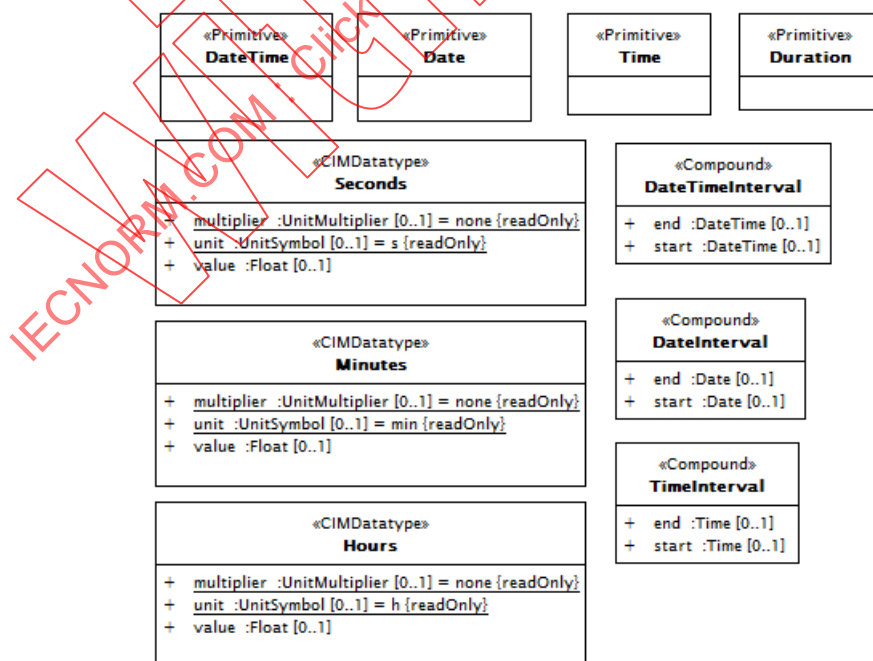


Figure 32 – Class diagram Domain::TimeDatatypes

This diagram shows time related data types.

6.3.2 ActivePower datatype

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

Table 8 shows all attributes of ActivePower.

Table 8 – Attributes of Domain::ActivePower

name	type	description
value	Float	
unit=W (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.3 ActivePowerChangeRate datatype

Rate of change of active power per time.

Table 9 shows all attributes of ActivePowerChangeRate.

Table 9 – Attributes of Domain::ActivePowerChangeRate

name	type	description
value	Float	
unit=W (const)	UnitSymbol	
multiplier	UnitMultiplier	
denominatorUnit=s (const)	UnitSymbol	
denominatorMultiplier=none (const)	UnitMultiplier	

6.3.4 Admittance datatype

Ratio of current to voltage.

Table 10 shows all attributes of Admittance.

Table 10 – Attributes of Domain::Admittance

name	type	description
value	Float	
unit=S (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.5 AngleDegrees datatype

Measurement of angle in degrees.

Table 11 shows all attributes of AngleDegrees.

Table 11 – Attributes of Domain::AngleDegrees

name	type	description
value	Float	
unit=deg (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	

6.3.6 AngleRadians datatype

Phase angle in radians.

Table 12 shows all attributes of AngleRadians.

Table 12 – Attributes of Domain::AngleRadians

name	type	description
value	Float	
unit=rad (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	

6.3.7 ApparentPower datatype

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

Table 13 shows all attributes of ApparentPower.

Table 13 – Attributes of Domain::ApparentPower

name	type	description
value	Float	
unit=VA (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.8 Boolean primitive

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

6.3.9 Capacitance datatype

Capacitive part of reactance (imaginary part of impedance), at rated frequency.

Table 14 shows all attributes of Capacitance.

Table 14 – Attributes of Domain::Capacitance

name	type	description
value	Float	
unit=F (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.10 Conductance datatype

Factor by which voltage must be multiplied to give corresponding power lost from a circuit. Real part of admittance.

Table 15 shows all attributes of Conductance.

Table 15 – Attributes of Domain::Conductance

name	type	description
value	Float	
unit=S (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.11 ConductancePerLength datatype

Real part of admittance per unit of length.

Table 16 shows all attributes of ConductancePerLength.

Table 16 – Attributes of Domain::ConductancePerLength

name	type	description
value	Float	
unit=S (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	
denominatorUnit=m (const)	UnitSymbol	
denominatorMultiplier=none (const)	UnitMultiplier	

6.3.12 CostPerEnergyUnit datatype

Cost, in units of currency, per quantity of electrical energy generated.

Table 17 shows all attributes of CostPerEnergyUnit.

Table 17 – Attributes of Domain::CostPerEnergyUnit

name	type	description
value	Float	
unit	Currency	
multiplier	UnitMultiplier	
denominatorUnit=Wh (const)	UnitSymbol	
denominatorMultiplier=none (const)	UnitMultiplier	

6.3.13 CostPerVolume datatype

Cost per unit volume.

Table 18 shows all attributes of CostPerVolume.

Table 18 – Attributes of Domain::CostPerVolume

name	type	description
denominatorMultiplier=none (const)	UnitMultiplier	
denominatorUnit=m3 (const)	UnitSymbol	
multiplier	UnitMultiplier	
unit	Currency	
value	Float	

6.3.14 CostRate datatype

Cost, in units of currency, per elapsed time.

Table 19 shows all attributes of CostRate.

Table 19 – Attributes of Domain::CostRate

name	type	description
value	Float	
unit	Currency	
multiplier	UnitMultiplier	
denominatorUnit=s (const)	UnitSymbol	
denominatorMultiplier=none (const)	UnitMultiplier	

6.3.15 Currency enumeration

Monetary currencies. Apologies for this list not being exhaustive.

Table 20 shows all literals of Currency.

Table 20 – Literals of Domain::Currency

literal	description
USD	US dollar
EUR	European euro
AUD	Australian dollar
CAD	Canadian dollar
CHF	Swiss francs
CNY	Chinese yuan renminbi
DKK	Danish crown
GBP	British pound
JPY	Japanese yen
NOK	Norwegian crown
RUR	Russian ruble
SEK	Swedish crown
INR	India rupees
other	Another type of currency.

6.3.16 CurrentFlow datatype

Electrical current with sign convention: positive flow is out of the conducting equipment into the connectivity node.

Table 21 shows all attributes of CurrentFlow.

Table 21 – Attributes of Domain::CurrentFlow

name	type	description
value	Float	
unit=A (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.17 Damping datatype

Per-unit active power variation with frequency referenced on the system apparent power base. Typical values are in range 1,0 – 2,0.

Table 22 shows all attributes of Damping.

Table 22 – Attributes of Domain::Damping

name	type	description
value	Float	
unit=none (const)	UnitSymbol	
multiplier	UnitMultiplier	
denominatorUnit=Hz (const)	UnitSymbol	
denominatorMultiplier=none (const)	UnitMultiplier	

6.3.18 Date primitive

Date and time as "yyyy-mm-dd", which conforms with ISO 8601. UTC time zone is specified as "yyyy-mm-ddZ". A local timezone relative UTC is specified as "yyyy-mm-dd(+/-)hh:mm".

6.3.19 DateInterval compound

Interval between two dates.

Table 23 shows all attributes of DateInterval.

Table 23 – Attributes of Domain::DateInterval

name	type	description
start	Date	Start date of this interval.
end	Date	End date of this interval.

6.3.20 DateTime primitive

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

6.3.21 DateTimeInterval compound

Interval between two date and time points.

Table 24 shows all attributes of DateTimeInterval.

Table 24 – Attributes of Domain::DateTimeInterval

name	type	description
start	DateTime	Start date and time of this interval.
end	DateTime	End date and time of this interval.

6.3.22 Decimal primitive

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

6.3.23 Displacement datatype

Unit of displacement relative a reference position, hence can be negative.

Table 25 shows all attributes of Displacement.

Table 25 – Attributes of Domain::Displacement

name	type	description
value	Float	
unit=m (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	

6.3.24 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.

6.3.25 Float primitive

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

6.3.26 FloatQuantity datatype

Quantity with float value and associated unit information.

Table 26 shows all attributes of FloatQuantity.

Table 26 – Attributes of Domain::FloatQuantity

name	type	description
value	Float	
unit	UnitSymbol	
multiplier	UnitMultiplier	

6.3.27 Frequency datatype

Cycles per second.

Table 27 shows all attributes of Frequency.

Table 27 – Attributes of Domain::Frequency

name	type	description
value	Float	
unit=Hz (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.28 Hours datatype

Time specified in hours.

Table 28 shows all attributes of Hours.

Table 28 – Attributes of Domain::Hours

name	type	description
value	Float	
unit=h (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	

6.3.29 Impedance datatype

Ratio of voltage to current.

Table 29 shows all attributes of Impedance.

Table 29 – Attributes of Domain::Impedance

name	type	description
value	Float	
unit=ohm (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.30 Inductance datatype

Inductive part of reactance (imaginary part of impedance), at rated frequency.

Table 30 shows all attributes of Inductance.

Table 30 – Attributes of Domain::Inductance

name	type	description
value	Float	
unit=H (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.31 Integer primitive

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

6.3.32 IntegerQuantity datatype

Quantity with integer value and associated unit information.

Table 31 shows all attributes of IntegerQuantity.

Table 31 – Attributes of Domain::IntegerQuantity

name	type	description
value	Integer	
unit	UnitSymbol	
multiplier	UnitMultiplier	

6.3.33 KiloActivePower datatype

Active power in kilowatts.

Table 32 shows all attributes of KiloActivePower.

Table 32 – Attributes of Domain::KiloActivePower

name	type	description
value	Float	
unit=W (const)	UnitSymbol	
multiplier=k (const)	UnitMultiplier	

6.3.34 Length datatype

Unit of length. Never negative.

Table 33 shows all attributes of Length.

Table 33 – Attributes of Domain::Length

name	type	description
multiplier	UnitMultiplier	
unit=m (const)	UnitSymbol	
value	Float	

6.3.35 Minutes datatype

Time in minutes.

Table 34 shows all attributes of Minutes.

Table 34 – Attributes of Domain::Minutes

name	type	description
value	Float	
unit=min (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	

6.3.36 Money datatype

Amount of money.

Table 35 shows all attributes of Money.

Table 35 – Attributes of Domain::Money

name	type	description
multiplier	UnitMultiplier	
unit	Currency	
value	Decimal	

6.3.37 PU datatype

Per Unit – a positive or negative value referred to a defined base. Values typically range from –10 to +10.

Table 36 shows all attributes of PU.

Table 36 – Attributes of Domain::PU

name	type	description
value	Float	
unit=none (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	

6.3.38 PerCent datatype

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

Table 37 shows all attributes of PerCent.

Table 37 – Attributes of Domain::PerCent

name	type	description
value	Float	Normally 0 – 100 on a defined base
unit=none (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	

6.3.39 Pressure datatype

Pressure in Pascal.

Table 38 shows all attributes of Pressure.

Table 38 – Attributes of Domain::Pressure

name	type	description
value	Float	
unit=Pa (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.40 Reactance datatype

Reactance (imaginary part of impedance), at rated frequency.

Table 39 shows all attributes of Reactance.

Table 39 – Attributes of Domain::Reactance

name	type	description
value	Float	
unit=ohm (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.41 ReactancePerLength datatype

Reactance (imaginary part of impedance) per unit of length, at rated frequency.

Table 40 shows all attributes of ReactancePerLength.

Table 40 – Attributes of Domain::ReactancePerLength

name	type	description
value	Float	
unit=ohm (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	
denominatorUnit=m (const)	UnitSymbol	
denominatorMultiplier=none (const)	UnitMultiplier	

6.3.42 ReactivePower datatype

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

Table 41 shows all attributes of ReactivePower.

Table 41 – Attributes of Domain::ReactivePower

name	type	description
value	Float	
unit=VAR (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.43 RealEnergy datatype

Real electrical energy.

Table 42 shows all attributes of RealEnergy.

Table 42 – Attributes of Domain::RealEnergy

name	type	description
value	Float	
unit=Wh (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.44 Resistance datatype

Resistance (real part of impedance).

Table 43 shows all attributes of Resistance.

Table 43 – Attributes of Domain::Resistance

name	type	description
value	Float	
unit=ohm (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.45 ResistancePerLength datatype

Resistance (real part of impedance) per unit of length.

Table 44 shows all attributes of ResistancePerLength.

Table 44 – Attributes of Domain::ResistancePerLength

name	type	description
value	Float	
unit=ohm (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	
denominatorUnit=m (const)	UnitSymbol	
denominatorMultiplier=none (const)	UnitMultiplier	

6.3.46 RotationSpeed datatype

Number of revolutions per second.

Table 45 shows all attributes of RotationSpeed.

Table 45 – Attributes of Domain::RotationSpeed

name	type	description
value	Float	
unit=none (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	
denominatorUnit=s (const)	UnitSymbol	
denominatorMultiplier=none (const)	UnitMultiplier	

6.3.47 Seconds datatype

Time, in seconds.

Table 46 shows all attributes of Seconds.

Table 46 – Attributes of Domain::Seconds

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

6.3.48 Speed datatype

Distance per unit of time.

Table 47 shows all attributes of Speed.

Table 47 – Attributes of Domain::Speed

name	type	description
value	Float	
unit=m (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	
denominatorUnit=s (const)	UnitSymbol	
denominatorMultiplier=none (const)	UnitMultiplier	

6.3.49 String primitive

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

6.3.50 StringQuantity datatype

Quantity with string value (when it is not important whether it is an integral or a floating point number) and associated unit information.

Table 48 shows all attributes of StringQuantity.

Table 48 – Attributes of Domain::StringQuantity

name	type	description
value	String	
unit	UnitSymbol	
multiplier	UnitMultiplier	

6.3.51 Susceptance datatype

Imaginary part of admittance.

Table 49 shows all attributes of Susceptance.

Table 49 – Attributes of Domain::Susceptance

name	type	description
value	Float	
unit=S (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.52 SusceptancePerLength datatype

Imaginary part of admittance per unit of length.

Table 50 shows all attributes of SusceptancePerLength.

Table 50 – Attributes of Domain::SusceptancePerLength

name	type	description
value	Float	
unit=S (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	
denominatorUnit=m (const)	UnitSymbol	
denominatorMultiplier=none (const)	UnitMultiplier	

6.3.53 Temperature datatype

Value of temperature in degrees Celsius.

Table 51 shows all attributes of Temperature.

Table 51 – Attributes of Domain::Temperature

name	type	description
multiplier=none (const)	UnitMultiplier	
unit=degC (const)	UnitSymbol	
value	Float	

6.3.54 Time primitive

Time as "hh:mm:ss.sss", which conforms with ISO 8601. UTC time zone is specified as "hh:mm:ss.sssZ". A local timezone relative UTC is specified as "hh:mm:ss.sss±hh:mm".

6.3.55 TimeInterval compound

Interval between two times.

Table 52 shows all attributes of TimeInterval.

Table 52 – Attributes of Domain::TimeInterval

name	type	description
start	Time	Start time of this interval.
end	Time	End time of this interval.

6.3.56 UnitMultiplier enumeration

The unit multipliers defined for the CIM.

Table 53 shows all literals of UnitMultiplier.

Table 53 – Literals of Domain::UnitMultiplier

literal	description
p	Pico 10^{*-12} .
n	Nano 10^{*-9} .
micro	Micro 10^{*-6} .
m	Milli 10^{*-3} .
c	Centi 10^{*-2} .
d	Deci 10^{*-1} .
k	Kilo 10^{*3} .
M	Mega 10^{*6} .
G	Giga 10^{*9} .
T	Tera 10^{*12} .
none	No multiplier or equivalently multiply by 1.

6.3.57 UnitSymbol enumeration

The units defined for usage in the CIM.

Table 54 shows all literals of UnitSymbol.

Table 54 – Literals of Domain::UnitSymbol

literal	description
VA	Apparent power in volts amperes.
W	Active power in watts.
VA _r	Reactive power in volts amperes reactive.
VA _h	Apparent energy in volt ampere hours.
Wh	Real energy in watt hours.
VA _r h	Reactive energy in volt ampere reactive hours.
V	Voltage in volts.
ohm	Resistance in ohms.
A	Current in amperes.
F	Capacitance in farads.
H	Inductance in henrys.

literal	description
degC	Relative temperature in degrees Celsius. In the SI unit system the symbol is °C. Electric charge is measured in coulomb that has the unit symbol C. To distinguish degree Celsius from coulomb the symbol used in the UML is degC. Reason for not using °C is the special character ° is difficult to manage in software.
s	Time in seconds.
min	Time in minutes.
h	Time in hours.
deg	Plane angle in degrees.
rad	Plane angle in radians.
J	Energy in joules.
N	Force in newtons.
S	Conductance in siemens.
none	Dimension less quantity, e.g. count, per unit, etc.
Hz	Frequency in hertz.
g	Mass in grams.
Pa	Pressure in pascals (N/m ²).
m	Length in metres.
m2	Area in square metres.
m3	Volume in cubic metres.

6.3.58 Voltage datatype

Electrical voltage.

Table 55 shows all attributes of Voltage.

Table 55 – Attributes of Domain::Voltage

name	type	description
value	Float	
unit=V (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.59 VoltagePerReactivePower datatype

Voltage variation with reactive power.

Table 56 shows all attributes of VoltagePerReactivePower.

Table 56 – Attributes of Domain::VoltagePerReactivePower

name	type	description
value	Float	
unit=V (const)	UnitSymbol	
multiplier	UnitMultiplier	
denominatorUnit=VA _r (const)	UnitSymbol	
denominatorMultiplier	UnitMultiplier	

6.3.60 Volume datatype

Volume.

Table 57 shows all attributes of Volume.

Table 57 – Attributes of Domain::Volume

name	type	description
value	Float	
unit=m3 (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.61 VolumeFlowRate datatype

Volume per time.

Table 58 shows all attributes of VolumeFlowRate.

Table 58 – Attributes of Domain::VolumeFlowRate

name	type	description
denominatorMultiplier=none (const)	UnitMultiplier	
denominatorUnit=s (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	
unit=m3 (const)	UnitSymbol	
value	Float	

6.3.62 WaterLevel datatype

Reservoir water level referred to a given datum such as mean sea level.

Table 59 shows all attributes of WaterLevel.

Table 59 – Attributes of Domain::WaterLevel

name	type	description
value	Float	
unit=m (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.63 Weight datatype

The weight of an object.

Table 60 shows all attributes of Weight.

Table 60 – Attributes of Domain::Weight

name	type	description
value	Float	
unit=g (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.4 Package Core

6.4.1 General

Contains the core PowerSystemResource and ConductingEquipment entities shared by all applications plus common collections of those entities. Not all applications require all the Core entities. This package does not depend on any other package except the Domain package, but most of the other packages have associations and generalizations that depend on it.

Figure 33 shows class diagram Main.

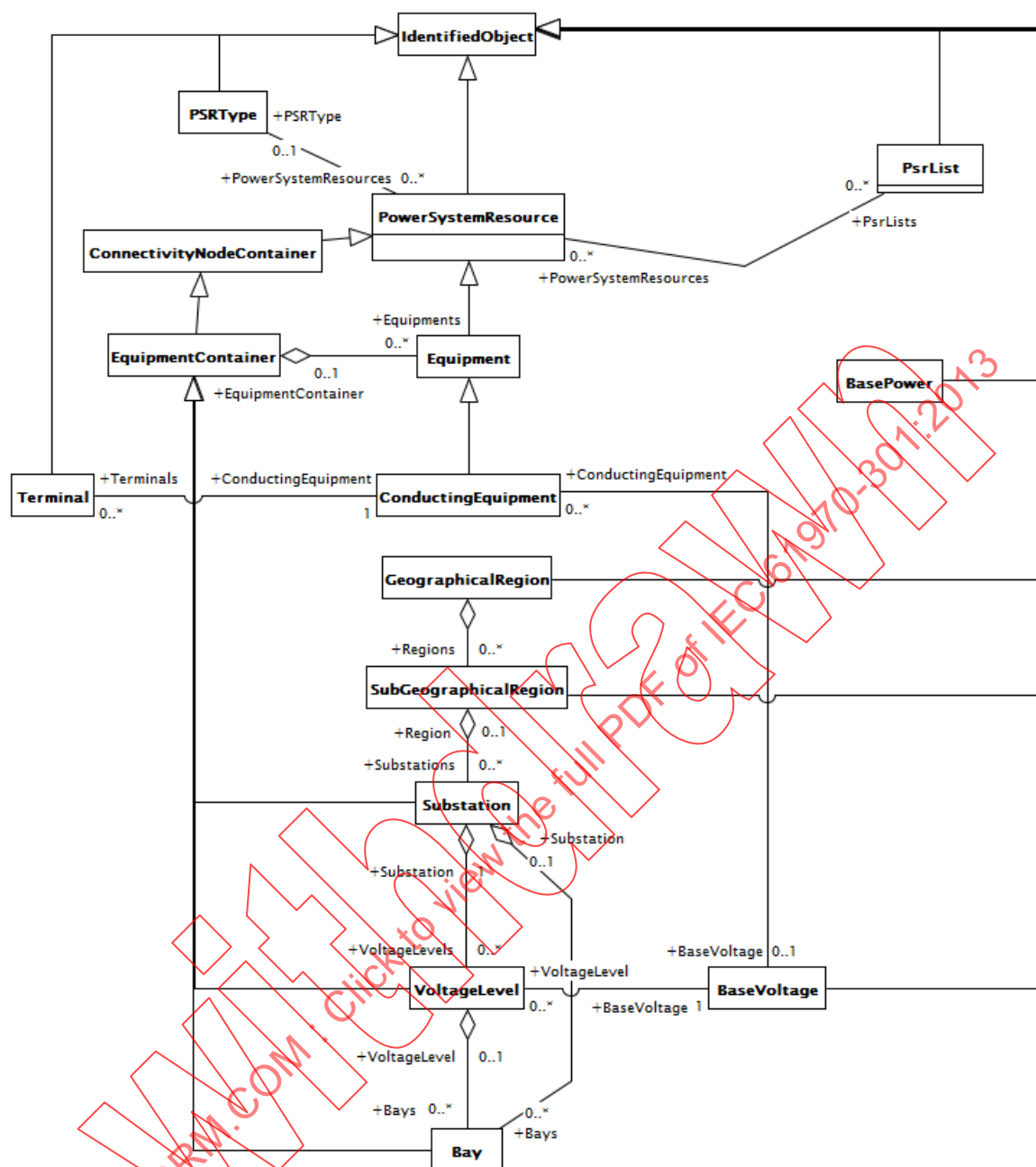


Figure 33 – Class diagram Core::Main

This diagram shows the main classes needed for any application of the CIM for modeling transmission and generation systems.

Figure 34 shows class diagram Names.

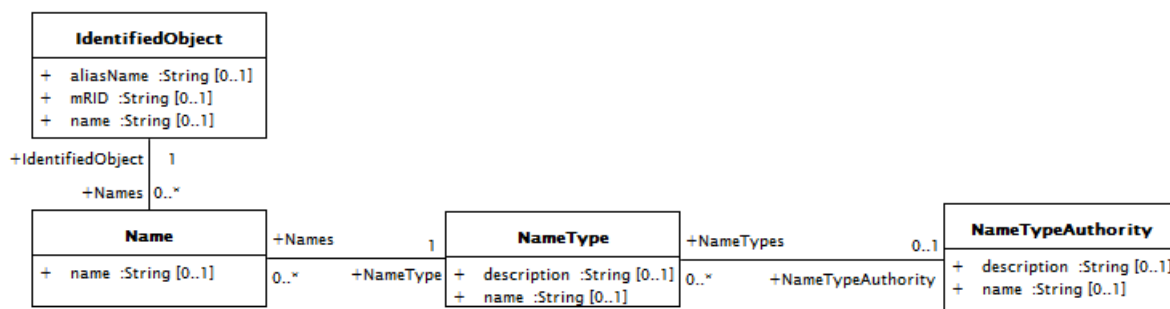


Figure 34 – Class diagram Core::Names

This diagram shows classes used to assign names to identified objects.

Figure 35 shows class diagram CurveSchedule.

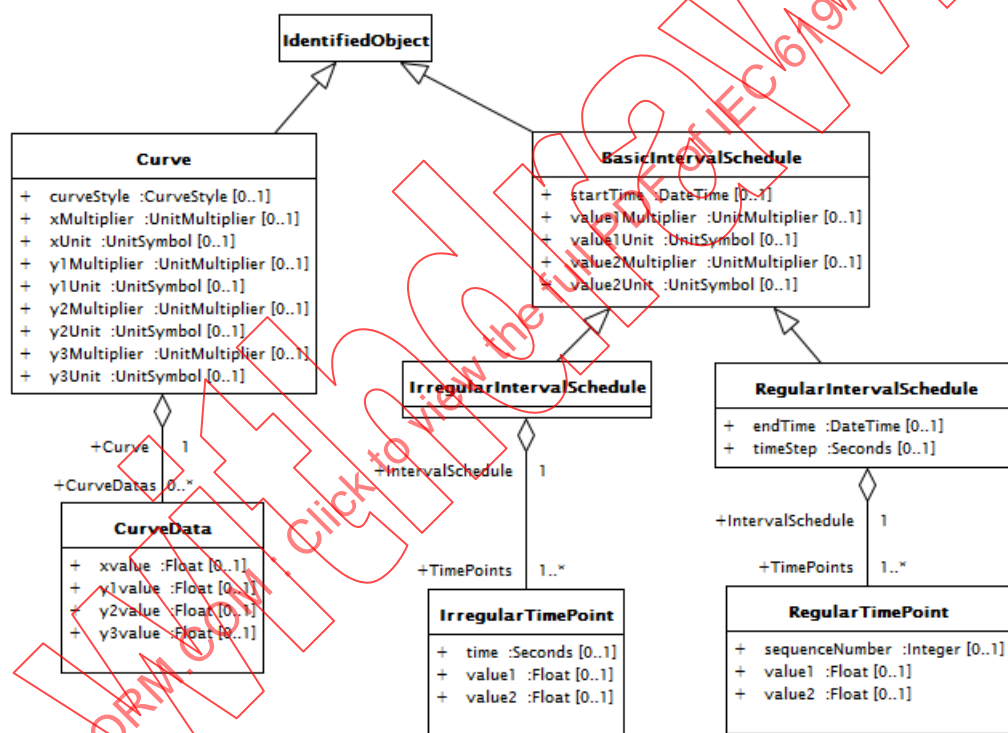


Figure 35 – Class diagram Core::CurveSchedule

This diagram shows the common model for curves and time dependent schedules.

Figure 36 shows class diagram Datatypes.

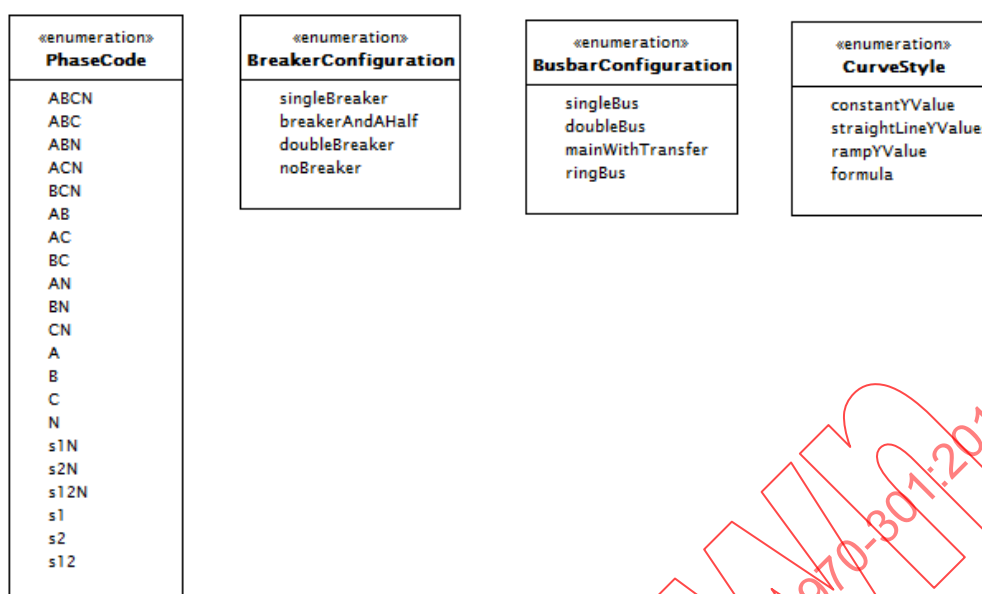


Figure 36 – Class diagram Core::Datatypes

This diagram shows the data types specific to the Core package.

Figure 37 shows class diagram Reporting.

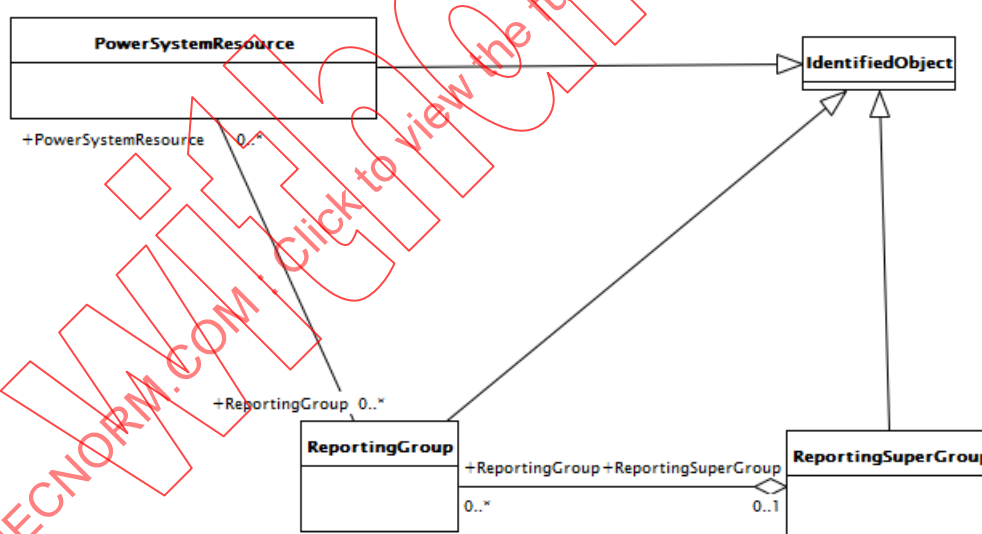


Figure 37 – Class diagram Core::Reporting

This diagram shows the common model for ad-hoc reporting.

Figure 38 shows class diagram OperatingShare.

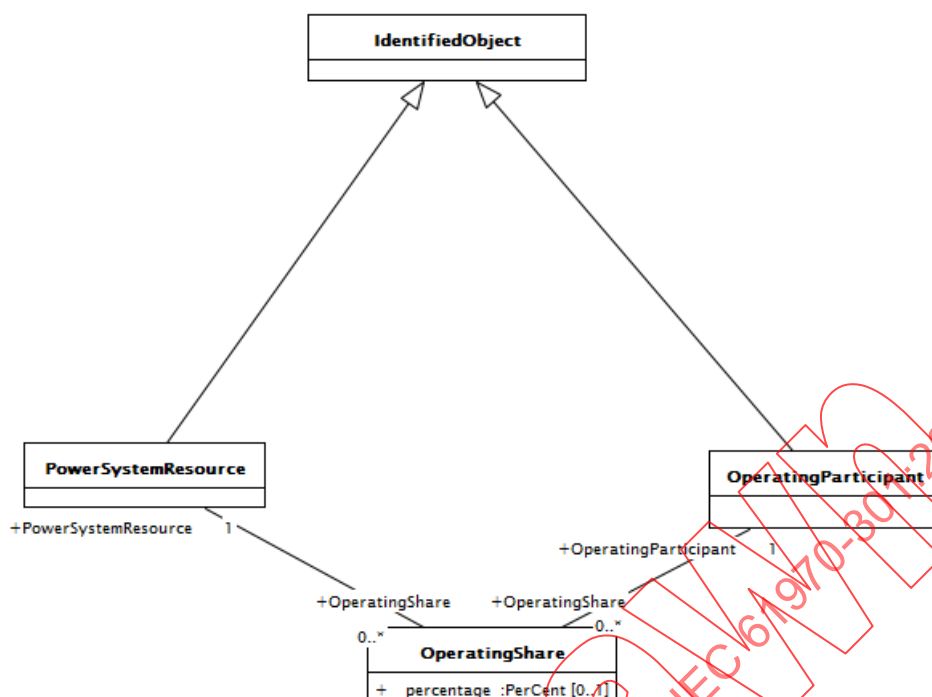


Figure 38 – Class diagram Core::OperatingShare

This diagram shows the common model for operating share.

6.4.2 BasePower

The BasePower class defines the base power used in the per unit calculations.

Table 61 shows all attributes of BasePower.

Table 61 – Attributes of Core::BasePower

name	type	description
basePower	ApparentPower	Value used as base power.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 62 shows all association ends of BasePower with other classes.

Table 62 – Association ends of Core::BasePower with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.4.3 BaseVoltage

Defines a system base voltage which is referenced.

Table 63 shows all attributes of BaseVoltage.

Table 63 – Attributes of Core::BaseVoltage

name	type	description
isDC	Boolean	If true, this is a direct current base voltage and items assigned to this base voltage are also associated with a direct current capabilities. False indicates alternating current.
nominalVoltage	Voltage	The power system resource's base voltage.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

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

Table 64 – Association ends of Core::BaseVoltage with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] ConductingEquipment	ConductingEquipment	All conducting equipment with this base voltage. Use only when there is no voltage level container used and only one base voltage applies. For example, not used for transformers.
[1..1]	[0..*] VoltageLevel	VoltageLevel	The voltage levels having this base voltage.
[0..1]	[0..*] TopologicalNode	TopologicalNode	The topological nodes at the base voltage.
[0..1]	[0..*] TransformerEnds	TransformerEnd	Transformer ends at the base voltage. This is essential for PU calculation.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.4.4 BasicIntervalSchedule

Schedule of values at points in time.

Table 65 shows all attributes of BasicIntervalSchedule.

Table 65 – Attributes of Core::BasicIntervalSchedule

name	type	description
startTime	DateTime	The time for the first time point.
value1Unit	UnitSymbol	Value1 units of measure.
value1Multiplier	UnitMultiplier	Multiplier for value1.
value2Unit	UnitSymbol	Value2 units of measure.
value2Multiplier	UnitMultiplier	Multiplier for value2.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 66 shows all association ends of BasicIntervalSchedule with other classes.

Table 66 – Association ends of Core::BasicIntervalSchedule with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.4.5 Bay

A collection of power system resources (within a given substation) including conducting equipment, protection relays, measurements, and telemetry. A bay typically represents a physical grouping related to modularization of equipment.

Table 67 shows all attributes of Bay.

Table 67 – Attributes of Core::Bay

name	type	description
bayEnergyMeasFlag	Boolean	Indicates the presence/absence of energy measurements.
bayPowerMeasFlag	Boolean	Indicates the presence/absence of active/reactive power measurements.
breakerConfiguration	BreakerConfiguration	Breaker configuration.
busBarConfiguration	BusbarConfiguration	Bus bar configuration.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 68 shows all association ends of Bay with other classes.

Table 68 – Association ends of Core::Bay with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] Substation	Substation	Substation containing the bay.
[0..*]	[0..1] VoltageLevel	VoltageLevel	The voltage level containing this bay.
[0..1]	[0..*] Equipments	Equipment	inherited from: EquipmentContainer
[1..1]	[0..*] ConnectivityNodes	ConnectivityNode	inherited from: ConnectivityNodeContainer
[0..1]	[0..*] TopologicalNode	TopologicalNode	inherited from: ConnectivityNodeContainer
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.4.6 BreakerConfiguration enumeration

Switching arrangement for bay.

Table 69 shows all literals of BreakerConfiguration.

Table 69 – Literals of Core::BreakerConfiguration

literal	description
singleBreaker	Single breaker.
breakerAndAHalf	Breaker and a half.
doubleBreaker	Double breaker.
noBreaker	No breaker.

6.4.7 BusbarConfiguration enumeration

Busbar layout for bay.

Table 70 shows all literals of BusbarConfiguration.

Table 70 – Literals of Core::BusbarConfiguration

literal	description
singleBus	Single bus.
doubleBus	Double bus.
mainWithTransfer	Main bus with transfer bus.
ringBus	Ring bus.

6.4.8 ConductingEquipment

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

Table 71 shows all attributes of ConductingEquipment.

Table 71 – Attributes of Core::ConductingEquipment

name	type	description
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 72 shows all association ends of ConductingEquipment with other classes.

Table 72 – Association ends of Core::ConductingEquipment with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] BaseVoltage	BaseVoltage	Base voltage of this conducting equipment. Use only when there is no voltage level container used and only one base voltage applies. For example, not used for transformers.
[1..1]	[0..*] Terminals	Terminal	Conducting equipment have terminals that may be connected to other conducting equipment terminals via connectivity nodes or topological nodes.
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	Protection equipment used to protect specific conducting equipment.
[1..1]	[0..*] ClearanceTags	ClearanceTag	Clearance tags for authorized field work on this conducting equipment.
[1..1]	[0..1] SvStatus	SvStatus	The status state variable associated with this conducting equipment.
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.4.9 ConnectivityNode

Connectivity nodes are points where terminals of conducting equipment are connected together with zero impedance.

Table 73 shows all attributes of ConnectivityNode.

Table 73 – Attributes of Core::ConnectivityNode

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 74 shows all association ends of ConnectivityNode with other classes.

Table 74 – Association ends of Core::ConnectivityNode with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] ConnectivityNodeContainer	ConnectivityNodeContainer	Container of this connectivity node.
[0..*]	[0..1] TopologicalNode	TopologicalNode	The topological node to which this connectivity node is assigned. May depend on the current state of switches in the network.
[0..1]	[0..*] Terminals	Terminal	Terminals interconnected with zero impedance at a this connectivity node.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.4.10 ConnectivityNodeContainer

A base class for all objects that may contain connectivity nodes or topological nodes.

Table 75 shows all attributes of ConnectivityNodeContainer.

Table 75 – Attributes of Core::ConnectivityNodeContainer

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 76 shows all association ends of ConnectivityNodeContainer with other classes.

Table 76 – Association ends of Core::ConnectivityNodeContainer with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] ConnectivityNodes	ConnectivityNode	Connectivity nodes which belong to this connectivity node container.
[0..1]	[0..*] TopologicalNode	TopologicalNode	The topological nodes which belong to this connectivity node container.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.4.11 Curve

A multi-purpose curve or functional relationship between an independent variable (X-axis) and dependent (Y-axis) variables.

Table 77 shows all attributes of Curve.

Table 77 – Attributes of Core::Curve

name	type	description
curveStyle	CurveStyle	The style or shape of the curve.
xMultiplier	UnitMultiplier	Multiplier for X-axis.
xUnit	UnitSymbol	The X-axis units of measure.
y1Multiplier	UnitMultiplier	Multiplier for Y1-axis.
y1Unit	UnitSymbol	The Y1-axis units of measure.
y2Multiplier	UnitMultiplier	Multiplier for Y2-axis.
y2Unit	UnitSymbol	The Y2-axis units of measure.
y3Multiplier	UnitMultiplier	Multiplier for Y3-axis.
y3Unit	UnitSymbol	The Y3-axis units of measure.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

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

Table 78 – Association ends of Core::Curve with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] CurveDatas	CurveData	The point data values that define this curve.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.4.12 CurveData root class

Multi-purpose data points for defining a curve. The use of this generic class is discouraged if a more specific class can be used to specify the x and y axis values along with their specific data types.

Table 79 shows all attributes of CurveData.

Table 79 – Attributes of Core::CurveData

name	type	description
xvalue	Float	The data value of the X-axis variable, depending on the X-axis units.
y1value	Float	The data value of the first Y-axis variable, depending on the Y-axis units.
y2value	Float	The data value of the second Y-axis variable (if present), depending on the Y-axis units.
y3value	Float	The data value of the third Y-axis variable (if present), depending on the Y-axis units.

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

Table 80 – Association ends of Core::CurveData with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Curve	Curve	The curve of this curve data point.

6.4.13 CurveStyle enumeration

Style or shape of curve.

Table 81 shows all literals of CurveStyle.

Table 81 – Literals of Core::CurveStyle

literal	description
constantYValue	The Y-axis values are assumed constant until the next curve point and prior to the first curve point.
straightLineYValues	The Y-axis values are assumed to be a straight line between values. Also known as linear interpolation.
rampYValue	The Y-axis values are assumed to ramp between points.
formula	An unspecified formula is assumed to compute the Y-axis value between points.

6.4.14 Equipment

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

Table 82 shows all attributes of Equipment.

Table 82 – Attributes of Core::Equipment

name	type	description
aggregate	Boolean	The single instance of equipment represents multiple pieces of equipment that have been modeled together as an aggregate. Examples would be power transformers or synchronous machines operating in parallel modeled as a single aggregate power transformer or aggregate synchronous machine. This is not to be used to indicate equipment that is part of a group of interdependent equipment produced by a network production program.
normallyInService	Boolean	If true, the equipment is normally in service.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 83 shows all association ends of Equipment with other classes.

Table 83 – Association ends of Core::Equipment with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	Container of this equipment.
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	The contingency equipments in which this equipment participates.
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	The operational limit sets associated with this equipment.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.4.15 EquipmentContainer

A modeling construct to provide a root class for containing equipment.

Table 84 shows all attributes of EquipmentContainer.

Table 84 – Attributes of Core::EquipmentContainer

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 85 shows all association ends of EquipmentContainer with other classes.

Table 85 – Association ends of Core::EquipmentContainer with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] Equipments	Equipment	Contained equipment.
[1..1]	[0..*] ConnectivityNodes	ConnectivityNode	inherited from: ConnectivityNodeContainer
[0..1]	[0..*] TopologicalNode	TopologicalNode	inherited from: ConnectivityNodeContainer
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.4.16 GeographicalRegion

A geographical region of a power system network model.

Table 86 shows all attributes of GeographicalRegion.

Table 86 – Attributes of Core::GeographicalRegion

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 87 shows all association ends of GeographicalRegion with other classes.

Table 87 – Association ends of Core::GeographicalRegion with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] Regions	SubGeographicalRegion	All sub-geographical regions within this geographical region.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.4.17 IdentifiedObject root class

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

Table 88 shows all attributes of IdentifiedObject.

Table 88 – Attributes of Core::IdentifiedObject

name	type	description
aliasName	String	The aliasName is free text human readable name of the object alternative to IdentifiedObject.name. It may be non-unique and may not correlate to a naming hierarchy. The attribute aliasName is retained because of backwards compatibility between CIM releases. It is however recommended to replace aliasName with the Name class as aliasName is planned for retirement at a future time.
mRID	String	Master resource identifier issued by a model authority. The mRID is globally unique within an exchange context. Global uniqueness is easily achieved by using a UUID for the mRID. It is strongly recommended to do this. For CIMXML data files in RDF syntax, the mRID is mapped to rdf:ID or rdf:about attributes that identify CIM object elements.
name	String	The name is any free human readable and possibly non-unique text naming the object.

Table 89 shows all association ends of IdentifiedObject with other classes.

Table 89 – Association ends of Core::IdentifiedObject with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] DiagramObjects	DiagramObject	The diagram objects that are associated with the domain object.
[1..1]	[0..*] Names	Name	All names of this identified object.

6.4.18 IrregularIntervalSchedule

The schedule has time points where the time between them varies.

Table 90 shows all attributes of IrregularIntervalSchedule.

Table 90 – Attributes of Core::IrregularIntervalSchedule

name	type	description
startTime	DateTime	inherited from: BasicIntervalSchedule
value1Unit	UnitSymbol	inherited from: BasicIntervalSchedule
value1Multiplier	UnitMultiplier	inherited from: BasicIntervalSchedule
value2Unit	UnitSymbol	inherited from: BasicIntervalSchedule
value2Multiplier	UnitMultiplier	inherited from: BasicIntervalSchedule
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 91 shows all association ends of IrregularIntervalSchedule with other classes.

Table 91 – Association ends of Core::IrregularIntervalSchedule with other classes

[mult from]	[mult to] name	type	description
[1..1]	[1..*] TimePoints	IrregularTimePoint	The point data values that define a curve.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.4.19 IrregularTimePoint root class

TimePoints for a schedule where the time between the points varies.

Table 92 shows all attributes of IrregularTimePoint.

Table 92 – Attributes of Core::IrregularTimePoint

name	type	description
time	Seconds	The time is relative to the schedule starting time.
value1	Float	The first value at the time. The meaning of the value is defined by the derived type of the associated schedule.
value2	Float	The second value at the time. The meaning of the value is defined by the derived type of the associated schedule.

Table 93 shows all association ends of IrregularTimePoint with other classes.

Table 93 – Association ends of Core::IrregularTimePoint with other classes

[mult from]	[mult to] name	type	description
[1..*]	[1..1] IntervalSchedule	IrregularIntervalSchedule	An IrregularTimePoint belongs to an IrregularIntervalSchedule.

6.4.20 Name root class

The Name class provides the means to define any number of human readable names for an object. A name is not to be used for defining inter-object relationships. For inter-object relationships instead use the object identification 'mRID'.

Table 94 shows all attributes of Name.

Table 94 – Attributes of Core::Name

name	type	description
name	String	Any free text that names the object.

Table 95 shows all association ends of Name with other classes.

Table 95 – Association ends of Core::Name with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] IdentifiedObject	IdentifiedObject	Identified object that this name designates.
[0..*]	[1..1] NameType	NameType	Type of this name.

6.4.21 NameType root class

Type of name. Possible values for attribute 'name' are implementation dependent but standard profiles may specify types. An enterprise may have multiple IT systems each having its own local name for the same object, e.g. a planning system may have different names from an EMS. An object may also have different names within the same IT system, e.g. localName as defined in CIM version 14. The definition from CIM14 is:

The localName is a human readable name of the object. It is a free text name local to a node in a naming hierarchy similar to a file directory structure. A power system related naming hierarchy may be: Substation, VoltageLevel, Equipment etc. Children of the same parent in such a hierarchy have names that typically are unique among them.

Table 96 shows all attributes of NameType.

Table 96 – Attributes of Core::NameType

name	type	description
name	String	Name of the name type.
description	String	Description of the name type.

Table 97 shows all association ends of NameType with other classes.

Table 97 – Association ends of Core::NameType with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] NameTypeAuthority	NameTypeAuthority	Authority responsible for managing names of this type.
[1..1]	[0..*] Names	Name	All names of this type.

6.4.22 NameTypeAuthority root class

Authority responsible for creation and management of names of a given type; typically an organization or an enterprise system.

Table 98 shows all attributes of NameTypeAuthority.

Table 98 – Attributes of Core::NameTypeAuthority

name	type	description
name	String	Name of the name type authority.
description	String	Description of the name type authority.

Table 99 shows all association ends of NameTypeAuthority with other classes.

Table 99 – Association ends of Core::NameTypeAuthority with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] NameTypes	NameType	All name types managed by this authority.

6.4.23 OperatingParticipant

An operator of multiple power system resource objects. Note multiple operating participants may operate the same power system resource object. This can be used for modeling jointly owned units where each owner operates as a contractual share.

Table 100 shows all attributes of OperatingParticipant.

Table 100 – Attributes of Core::OperatingParticipant

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 101 shows all association ends of OperatingParticipant with other classes.

Table 101 – Association ends of Core::OperatingParticipant with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] OperatingShare	OperatingShare	The operating shares of this operating participant. An operating participant can be resused for any number of power system resources.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.4.24 OperatingShare root class

Specifies the operations contract relationship between a power system resource and a contract participant.

Table 102 shows all attributes of OperatingShare.

Table 102 – Attributes of Core::OperatingShare

name	type	description
percentage	PerCent	Percentage operational ownership between the pair (power system resource and operating participant) associated with this share. The total percentage ownership for a power system resource should add to 100%.

Table 103 shows all association ends of OperatingShare with other classes.

Table 103 – Association ends of Core::OperatingShare with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] OperatingParticipant	OperatingParticipant	The operating participant having this share with the associated power system resource.
[0..*]	[1..1] PowerSystemResource	PowerSystemResource	The power system resource to which the share applies.

6.4.25 PhaseCode enumeration

Enumeration of phase identifiers. Allows designation of phases for both transmission and distribution equipment, circuits and loads.

Residential and small commercial loads are often served from single-phase, or split-phase, secondary circuits. For example of s12N, phases 1 and 2 refer to hot wires that are 180 degrees out of phase, while N refers to the neutral wire. Through single-phase transformer connections, these secondary circuits may be served from one or two of the primary phases A, B, and C. For three-phase loads, use the A, B, C phase codes instead of s12N.

Table 104 shows all literals of PhaseCode.

Table 104 – Literals of Core::PhaseCode

literal	description
ABCN	Phases A, B, C, and N.
ABC	Phases A, B, and C.
ABN	Phases A, B, and neutral.
ACN	Phases A, C and neutral.
BCN	Phases B, C, and neutral.
AB	Phases A and B.
AC	Phases A and C.
BC	Phases B and C.
AN	Phases A and neutral.
BN	Phases B and neutral.
CN	Phases C and neutral.
A	Phase A.
B	Phase B.
C	Phase C.
N	Neutral phase.
s1N	Secondary phase 1 and neutral.
s2N	Secondary phase 2 and neutral.
s12N	Secondary phases 1, 2, and neutral.
s1	Secondary phase 1.
s2	Secondary phase 2.
s12	Secondary phase 1 and 2.

6.4.26 PowerSystemResource

A power system resource can be an item of equipment such as a switch, an equipment container containing many individual items of equipment such as a substation, or an organisational entity such as sub-control area. Power system resources can have measurements associated.

Table 105 shows all attributes of PowerSystemResource.

Table 105 – Attributes of Core::PowerSystemResource

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

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

Table 106 – Association ends of Core::PowerSystemResource with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] PSRType	PSRType	Custom classification for this power system resource.
[0..1]	[0..*] Measurements	Measurement	The measurements associated with this power system resource.
[1..1]	[0..*] OperatingShare	OperatingShare	The operating shares of this power system resource.
[0..*]	[0..*] PsrLists	PsrList	The lists to which this power system resource belongs.
[1..1]	[0..1] OutageSchedule	OutageSchedule	A power system resource may have an outage schedule.
[0..*]	[0..*] ReportingGroup	ReportingGroup	Reporting groups to which this power system resource belongs.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.4.27 PsrList

Arbitrary list of power system resources. Can be used for various purposes, including grouping for report generation.

Table 107 shows all attributes of PsrList.

Table 107 – Attributes of Core::PsrList

name	type	description
typePSRList	String	Type of power system resources in this list.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

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

Table 108 – Association ends of Core::PsrList with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] PowerSystemResources	PowerSystemResource	The power system resources in the list.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.4.28 PSRType

Classifying instances of the same class, e.g. overhead and underground ACLineSegments. This classification mechanism is intended to provide flexibility outside the scope of this standard, i.e. provide customisation that is non standard.

Table 109 shows all attributes of PSRType.

Table 109 – Attributes of Core::PSRType

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

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

Table 110 – Association ends of Core::PSRType with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] PowerSystemResources	PowerSystemResource	Power system resources classified with this power system resource type.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.4.29 RegularIntervalSchedule

The schedule has time points where the time between them is constant.

Table 111 shows all attributes of RegularIntervalSchedule.

Table 111 – Attributes of Core::RegularIntervalSchedule

name	type	description
endTime	DateTime	The time for the last time point.
timeStep	Seconds	The time between each pair of subsequent regular time points in sequence order.
startTime	DateTime	inherited from: BasicIntervalSchedule
value1Unit	UnitSymbol	inherited from: BasicIntervalSchedule
value1Multiplier	UnitMultiplier	inherited from: BasicIntervalSchedule
value2Unit	UnitSymbol	inherited from: BasicIntervalSchedule
value2Multiplier	UnitMultiplier	inherited from: BasicIntervalSchedule
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 112 shows all association ends of RegularIntervalSchedule with other classes.

Table 112 – Association ends of Core::RegularIntervalSchedule with other classes

[mult from]	[mult to] name	type	description
[1..1]	[1..*] TimePoints	RegularTimePoint	The regular interval time point data values that define this schedule.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.4.30 RegularTimePoint root class

Time point for a schedule where the time between the consecutive points is constant.

Table 113 shows all attributes of RegularTimePoint.

Table 113 – Attributes of Core::RegularTimePoint

name	type	description
sequenceNumber	Integer	The position of the regular time point in the sequence. Note that time points do not have to be sequential, i.e. time points may be omitted. The actual time for a RegularTimePoint is computed by multiplying the associated regular interval schedule's time step with the regular time point sequence number and adding the associated schedules start time.
value1	Float	The first value at the time. The meaning of the value is defined by the derived type of the associated schedule.
value2	Float	The second value at the time. The meaning of the value is defined by the derived type of the associated schedule.

Table 114 shows all association ends of RegularTimePoint with other classes.

Table 114 – Association ends of Core::RegularTimePoint with other classes

[mult from]	[mult to] name	type	description
[1..*]	[1..1] IntervalSchedule	RegularIntervalSchedule	Regular interval schedule containing this time point.

6.4.31 ReportingGroup

A reporting group is used for various ad-hoc groupings used for reporting.

Table 115 shows all attributes of ReportingGroup.

Table 115 – Attributes of Core::ReportingGroup

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 116 shows all association ends of ReportingGroup with other classes.

Table 116 – Association ends of Core::ReportingGroup with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] PowerSystemResource	PowerSystemResource	Power system resources which belong to this reporting group.
[0..*]	[0..1] ReportingSuperGroup	ReportingSuperGroup	Reporting super group to which this reporting group belongs.
[0..1]	[0..*] BusNameMarker	BusNameMarker	The bus name markers that belong to this reporting group.
[0..1]	[0..*] TopologicalNode	TopologicalNode	The topological nodes that belong to the reporting group.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.4.32 ReportingSuperGroup

A reporting super group, groups reporting groups for a higher level report.

Table 117 shows all attributes of ReportingSuperGroup.

Table 117 – Attributes of Core::ReportingSuperGroup

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 118 shows all association ends of ReportingSuperGroup with other classes.

Table 118 – Association ends of Core::ReportingSuperGroup with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] ReportingGroup	ReportingGroup	Reporting groups that are grouped under this super group.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.4.33 SubGeographicalRegion

A subset of a geographical region of a power system network model.

Table 119 shows all attributes of SubGeographicalRegion.

Table 119 – Attributes of Core::SubGeographicalRegion

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 120 shows all association ends of SubGeographicalRegion with other classes.

Table 120 – Association ends of Core::SubGeographicalRegion with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] Substations	Substation	The substations in this sub-geographical region.
[0..1]	[0..*] Lines	Line	The lines within the sub-geographical region.
[0..*]	[0..1] Region	GeographicalRegion	The geographical region to which this sub-geographical region is within.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.4.34 Substation

A collection of equipment for purposes other than generation or utilization, through which electric energy in bulk is passed for the purposes of switching or modifying its characteristics.

Table 121 shows all attributes of Substation.

Table 121 – Attributes of Core::Substation

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 122 shows all association ends of Substation with other classes.

Table 122 – Association ends of Core::Substation with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] VoltageLevels	VoltageLevel	The voltage levels within this substation.
[0..1]	[0..*] Bays	Bay	Bays contained in the substation.
[0..*]	[0..1] Region	SubGeographicalRegion	The sub-geographical containing the the substation.
[0..1]	[0..*] Equipments	Equipment	inherited from: EquipmentContainer
[1..1]	[0..*] ConnectivityNodes	ConnectivityNode	inherited from: ConnectivityNodeContainer
[0..1]	[0..*] TopologicalNode	TopologicalNode	inherited from: ConnectivityNodeContainer
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.4.35 Terminal

An electrical connection point to a piece of conducting equipment. Terminals are connected at physical connection points called connectivity nodes.

Table 123 shows all attributes of Terminal.

Table 123 – Attributes of Core::Terminal

name	type	description
connected	Boolean	The connected status is related to a bus-branch model and the topological node to terminal relation. True implies the terminal is connected to the related topological node and false implies it is not. In a bus-branch model, the connected status is used to tell if equipment is disconnected without having to change the connectivity described by the topological node to terminal relation. A valid case is that conducting equipment can be connected in one end and open in the other. In particular for an AC line segment, where the reactive line charging can be significant, this is a relevant case.
phases	PhaseCode	Represents the normal network phasing condition.
sequenceNumber	Integer	The orientation of the terminal connections for a multiple terminal conducting equipment. The sequence numbering starts with 1 and additional terminals should follow in increasing order. The first terminal is the "starting point" for a two terminal branch.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 124 shows all association ends of Terminal with other classes.

Table 124 – Association ends of Core::Terminal with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] ConductingEquipment	ConductingEquipment	The conducting equipment of the terminal. Conducting equipment have terminals that may be connected to other conducting equipment terminals via connectivity nodes or topological nodes.
[0..*]	[0..1] ConnectivityNode	ConnectivityNode	The connectivity node to which this terminal connects with zero impedance.
[0..1]	[0..*] RegulatingControl	RegulatingControl	The controls regulating this terminal.
[0..*]	[0..1] TopologicalNode	TopologicalNode	The topological node associated with the terminal. This can be used as an alternative to the connectivity node path to topological node, thus making it unnecessary to model connectivity nodes in some cases. Note that if connectivity nodes are in the model, this association would probably not be used as an input specification.
[1..*]	[0..1] BusNameMarker	BusNameMarker	The bus name marker used to name the bus (topological node).
[1..1]	[0..*] BranchGroupTerminal	BranchGroupTerminal	The directed branch group terminals for which this terminal is monitored.
[0..1]	[0..*] Measurements	Measurement	Measurements associated with this terminal defining where the measurement is placed in the network topology. It may be used, for instance, to capture the sensor position, such as a voltage transformer (PT) at a busbar or a current transformer (CT) at the bar between a breaker and an isolator.
[1..1]	[0..2] TieFlow	TieFlow	The control area tie flows to which this terminal associates.
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	The operational limit sets that apply specifically to this terminal. Other operational limits sets may apply to this terminal through the association to equipment.
[1..1]	[0..*] HasSecond_MutualCoupling	MutualCoupling	Mutual couplings with the branch associated as the first branch.
[1..1]	[0..*] HasFirst_MutualCoupling	MutualCoupling	Mutual couplings associated with the branch as the first branch.
[1..1]	[0..1] SvPowerFlow	SvPowerFlow	The power flow state variable associated with the terminal.
[1..1]	[0..*] AuxiliaryEquipment	AuxiliaryEquipment	The auxiliary equipment connected to the terminal.
[0..1]	[0..*] TransformerEnd	TransformerEnd	All transformer ends connected at this terminal.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.4.36 VoltageLevel

A collection of equipment at one common system voltage forming a switchgear. The equipment typically consist of breakers, busbars, instrumentation, control, regulation and protection devices as well as assemblies of all these.

Table 125 shows all attributes of VoltageLevel.

Table 125 – Attributes of Core::VoltageLevel

name	type	description
highVoltageLimit	Voltage	The bus bar's high voltage limit
lowVoltageLimit	Voltage	The bus bar's low voltage limit
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 126 shows all association ends of VoltageLevel with other classes.

Table 126 – Association ends of Core::VoltageLevel with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] BaseVoltage	BaseVoltage	The base voltage used for all equipment within the voltage level.
[0..1]	[0..*] Bays	Bay	The bays within this voltage level.
[0..*]	[1..1] Substation	Substation	The substation of the voltage level.
[0..1]	[0..*] Equipments	Equipment	inherited from: EquipmentContainer
[1..1]	[0..*] ConnectivityNodes	ConnectivityNode	inherited from: ConnectivityNodeContainer
[0..1]	[0..*] TopologicalNode	TopologicalNode	inherited from: ConnectivityNodeContainer
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.5 Package DiagramLayout

6.5.1 General

This package describes diagram layout. This describes how objects are arranged in a coordinate system rather than how they are rendered.

Figure 39 shows class diagram DiagramLayout.

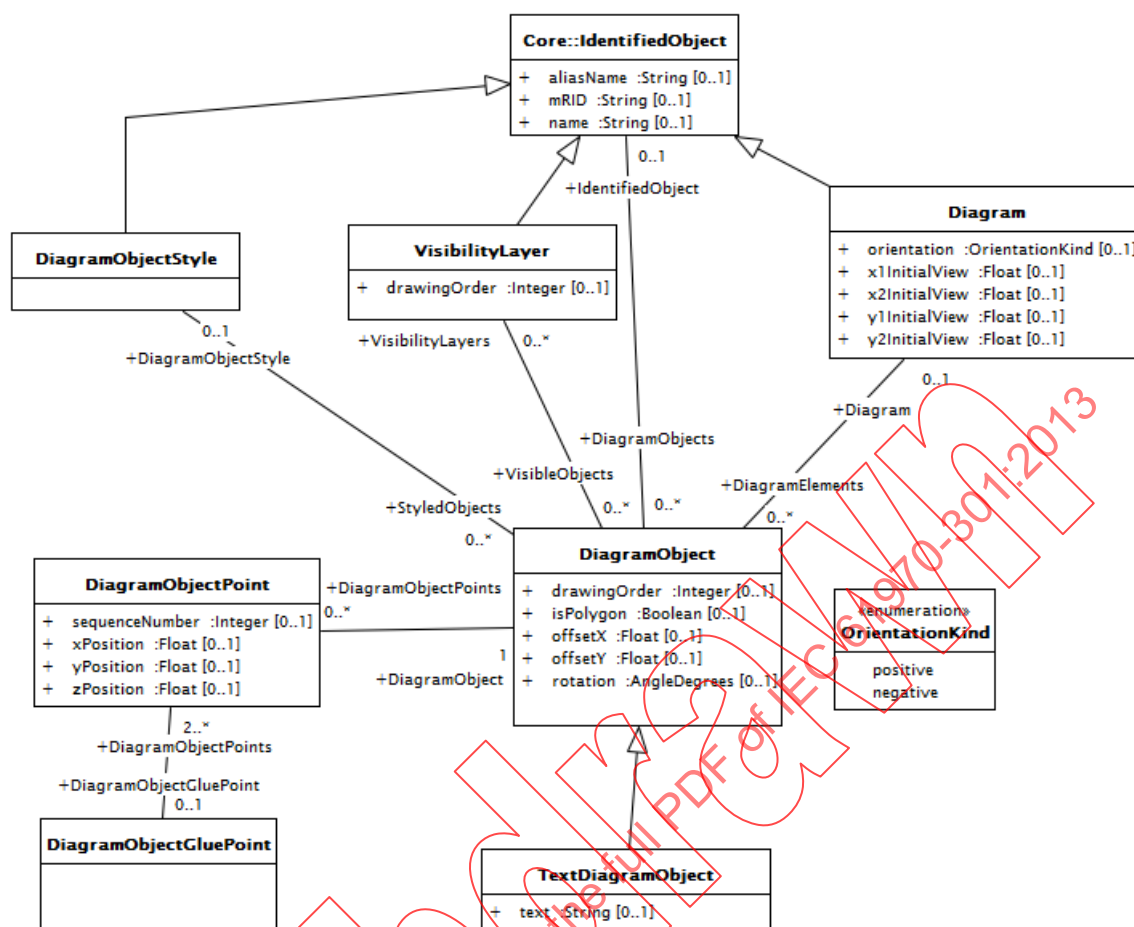


Figure 39 – Class diagram DiagramLayout::DiagramLayout

The diagram shows the diagram layout model.

6.5.2 Diagram

The diagram being exchanged. The coordinate system is a standard Cartesian coordinate system and the orientation attribute defines the orientation.

Table 127 shows all attributes of Diagram.

Table 127 – Attributes of DiagramLayout::Diagram

name	type	description
orientation	OrientationKind	Coordinate system orientation of the diagram.
x1InitialView	Float	X coordinate of the first corner of the initial view.
x2InitialView	Float	X coordinate of the second corner of the initial view.
y1InitialView	Float	Y coordinate of the first corner of the initial view.
y2InitialView	Float	Y coordinate of the second corner of the initial view.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 128 shows all association ends of Diagram with other classes.

Table 128 – Association ends of DiagramLayout::Diagram with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] DiagramElements	DiagramObject	A diagram is made up of multiple diagram objects.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.5.3 DiagramObject

An object that defines one or more points in a given space. This object can be associated with anything that specializes IdentifiedObject. For single line diagrams such objects typically include such items as analog values, breakers, disconnectors, power transformers, and transmission lines.

Table 129 shows all attributes of DiagramObject.

Table 129 – Attributes of DiagramLayout::DiagramObject

name	type	description
drawingOrder	Integer	The drawing order of this element. The higher the number, the later the element is drawn in sequence. This is used to ensure that elements that overlap are rendered in the correct order.
isPolygon	Boolean	Defines whether or not the diagram objects points define the boundaries of a polygon or the routing of a polyline. If this value is true then a receiving application should consider the first and last points to be connected.
offsetX	Float	The offset in the X direction. This is used for defining the offset from centre for rendering an icon (the default is that a single point specifies the centre of the icon). The offset is in per-unit with 0 indicating there is no offset from the horizontal centre of the icon. -0,5 indicates it is offset by 50 % to the left and 0,5 indicates an offset of 50 % to the right.
offsetY	Float	The offset in the Y direction. This is used for defining the offset from centre for rendering an icon (the default is that a single point specifies the centre of the icon). The offset is in per-unit with 0 indicating there is no offset from the vertical centre of the icon. The offset direction is dependent on the orientation of the diagram, with -0,5 and 0,5 indicating an offset of +/- 50 % on the vertical axis.
rotation	AngleDegrees	Sets the angle of rotation of the diagram object in a clockwise direction from the normal
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 130 shows all association ends of DiagramObject with other classes.

Table 130 – Association ends of DiagramLayout::DiagramObject with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] Diagram	Diagram	A diagram object is part of a diagram.
[0..*]	[0..*] VisibilityLayers	VisibilityLayer	A diagram object can be part of multiple visibility layers.
[1..1]	[0..*] DiagramObjectPoints	DiagramObjectPoint	A diagram object can have 0 or more points to reflect its layout position, routing (for polylines) or boundary (for polygons).
[0..*]	[0..1] DiagramObjectStyle	DiagramObjectStyle	A diagram object has a style associated that provides a reference for the style used in the originating system.
[0..*]	[0..1] IdentifiedObject	IdentifiedObject	The domain object to which this diagram object is associated.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.5.4 DiagramObjectGluePoint root class

This is used for grouping diagram object points from different diagram objects that are considered to be glued together in a diagram even if they are not at the exact same coordinates.

Table 131 shows all association ends of DiagramObjectGluePoint with other classes.

Table 131 – Association ends of DiagramLayout::DiagramObjectGluePoint with other classes

[mult from]	[mult to] name	type	description
[0..1]	[2..*] DiagramObjectPoints	DiagramObjectPoint	A diagram object glue point is associated with 2 or more object points that are considered to be 'glued' together.

6.5.5 DiagramObjectPoint root class

A point in a given space defined by 3 coordinates and associated to a diagram object. The coordinates may be positive or negative as the origin does not have to be in the corner of a diagram.

Table 132 shows all attributes of DiagramObjectPoint.

Table 132 – Attributes of DiagramLayout::DiagramObjectPoint

name	type	description
sequenceNumber	Integer	The sequence position of the point, used for defining the order of points for diagram objects acting as a polyline or polygon with more than one point.
xPosition	Float	The X coordinate of this point.
yPosition	Float	The Y coordinate of this point.
zPosition	Float	The Z coordinate of this point.

Table 133 shows all association ends of DiagramObjectPoint with other classes.

Table 133 – Association ends of DiagramLayout::DiagramObjectPoint with other classes

[mult from]	[mult to] name	type	description
[2..*]	[0..1] DiagramObjectGluePoint	DiagramObjectGluePoint	The 'glue' point to which this point is associated.
[0..*]	[1..1] DiagramObject	DiagramObject	The diagram object with which the points are associated.

6.5.6 DiagramObjectStyle

A reference to a style used by the originating system for a diagram object. A diagram object style describes information such as line thickness, shape such as circle or rectangle, etc., and color.

Table 134 shows all attributes of DiagramObjectStyle.

Table 134 – Attributes of DiagramLayout::DiagramObjectStyle

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 135 shows all association ends of DiagramObjectStyle with other classes.

Table 135 – Association ends of DiagramLayout::DiagramObjectStyle with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] StyledObjects	DiagramObject	A style can be assigned to multiple diagram objects.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.5.7 OrientationKind enumeration

The orientation of the coordinate system with respect to top, left, and the coordinate number system.

Table 136 shows all literals of OrientationKind.

Table 136 – Literals of DiagramLayout::OrientationKind

literal	description
positive	For 2D diagrams, a positive orientation will result in X values increasing from left to right and Y values increasing from bottom to top. This is also known as a right hand orientation.
negative	For 2D diagrams, a negative orientation gives the left-hand orientation (favoured by computer graphics displays) with X values increasing from left to right and Y values increasing from top to bottom. This is also known as a left hand orientation.

6.5.8 TextDiagramObject

A diagram object for placing free-text or text derived from an associated domain object.

Table 137 shows all attributes of TextDiagramObject.

Table 137 – Attributes of DiagramLayout::TextDiagramObject

name	type	description
text	String	The text that is displayed by this text diagram object.
drawingOrder	Integer	inherited from: DiagramObject
isPolygon	Boolean	inherited from: DiagramObject
offsetX	Float	inherited from: DiagramObject
offsetY	Float	inherited from: DiagramObject
rotation	AngleDegrees	inherited from: DiagramObject
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 138 shows all association ends of TextDiagramObject with other classes.

Table 138 – Association ends of DiagramLayout::TextDiagramObject with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] Diagram	Diagram	inherited from: DiagramObject
[0..*]	[0..*] VisibilityLayers	VisibilityLayer	inherited from: DiagramObject
[1..1]	[0..*] DiagramObjectPoints	DiagramObjectPoint	inherited from: DiagramObject
[0..*]	[0..1] DiagramObjectStyle	DiagramObjectStyle	inherited from: DiagramObject
[0..*]	[0..1] IdentifiedObject	IdentifiedObject	inherited from: DiagramObject
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.5.9 VisibilityLayer

Layers are typically used for grouping diagram objects according to themes and scales. Themes are used to display or hide certain information (e.g., lakes, borders), while scales are used for hiding or displaying information depending on the current zoom level (hide text when it is too small to be read, or when it exceeds the screen size). This is also called de-cluttering.

CIM based graphics exchange will support an m:n relationship between diagram objects and layers. It will be the task of the importing system to convert an m:n case into an appropriate 1:n representation if the importing system does not support m:n.

Table 139 shows all attributes of VisibilityLayer.

Table 139 – Attributes of DiagramLayout::VisibilityLayer

name	type	description
drawingOrder	Integer	The drawing order for this layer. The higher the number, the later the layer and the objects within it are rendered.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 140 shows all association ends of VisibilityLayer with other classes.

Table 140 – Association ends of DiagramLayout::VisibilityLayer with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] VisibleObjects	DiagramObject	A visibility layer can contain one or more diagram objects.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.6 Package OperationalLimits

6.6.1 General

This package models a specification of limits associated with equipment and other operational entities.

Figure 40 shows class diagram OperationalLimits.

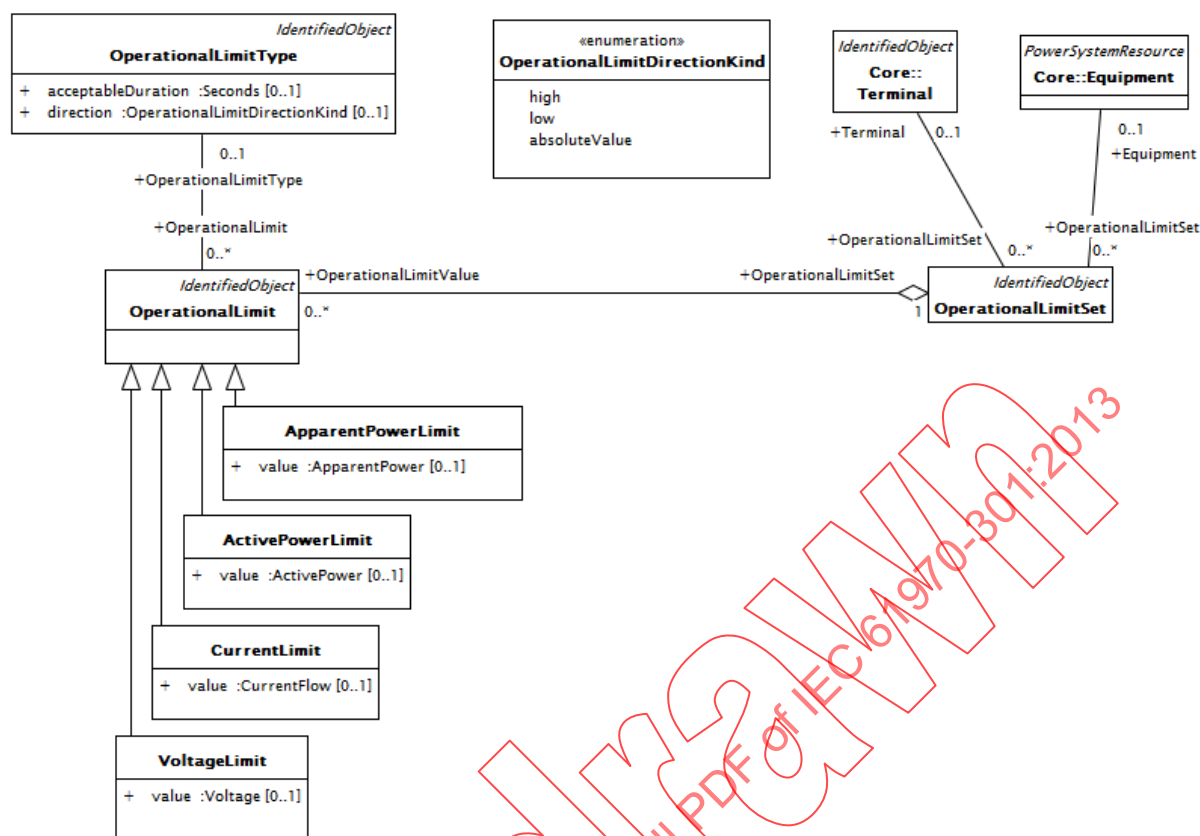


Figure 40 – Class diagram OperationalLimits::OperationalLimits

This diagram shows operational limits as they tie back into the core model.

Figure 41 shows class diagram BranchGroup.

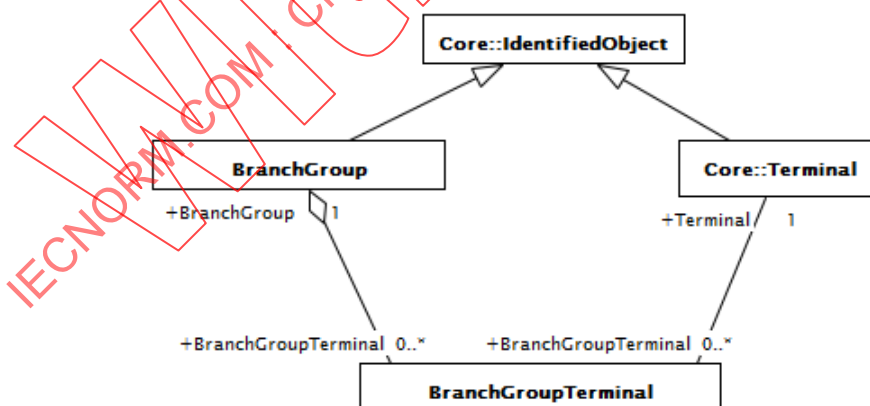


Figure 41 – Class diagram OperationalLimits::BranchGroup

This diagram shows the model for monitoring combined flows on groups of branches.

6.6.2 ActivePowerLimit

Limit on active power flow.

Table 141 shows all attributes of ActivePowerLimit.

Table 141 – Attributes of OperationalLimits::ActivePowerLimit

name	type	description
value	ActivePower	Value of active power limit.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 142 shows all association ends of ActivePowerLimit with other classes.

Table 142 – Association ends of OperationalLimits::ActivePowerLimit with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] OperationalLimitSet	OperationalLimitSet	inherited from: OperationalLimit
[0..*]	[0..1] OperationalLimitType	OperationalLimitType	inherited from: OperationalLimit
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.6.3 ApparentPowerLimit

Apparent power limit.

Table 143 shows all attributes of ApparentPowerLimit.

Table 143 – Attributes of OperationalLimits::ApparentPowerLimit

name	type	description
value	ApparentPower	The apparent power limit.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 144 shows all association ends of ApparentPowerLimit with other classes.

Table 144 – Association ends of OperationalLimits::ApparentPowerLimit with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] OperationalLimitSet	OperationalLimitSet	inherited from: OperationalLimit
[0..*]	[0..1] OperationalLimitType	OperationalLimitType	inherited from: OperationalLimit
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.6.4 BranchGroup

A group of branch terminals whose directed flow summation is to be monitored. A branch group need not form a cutset of the network.

Table 145 shows all attributes of BranchGroup.

Table 145 – Attributes of OperationalLimits::BranchGroup

name	type	description
maximumActivePower	ActivePower	The maximum active power flow.
maximumReactivePower	ReactivePower	The maximum reactive power flow.
minimumActivePower	ActivePower	The minimum active power flow.
minimumReactivePower	ReactivePower	The minimum reactive power flow.
monitorActivePower	Boolean	Monitor the active power flow.
monitorReactivePower	Boolean	Monitor the reactive power flow.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 146 shows all association ends of BranchGroup with other classes.

Table 146 – Association ends of OperationalLimits::BranchGroup with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] BranchGroupTerminal	BranchGroupTerminal	The directed branch group terminals to be summed.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.6.5 BranchGroupTerminal root class

A specific directed terminal flow for a branch group.

Table 147 shows all attributes of BranchGroupTerminal.

Table 147 – Attributes of OperationalLimits::BranchGroupTerminal

name	type	description
positiveFlowIn	Boolean	The flow into the terminal is summed if set true. The flow out of the terminal is summed if set false.

Table 148 shows all association ends of BranchGroupTerminal with other classes.

Table 148 – Association ends of OperationalLimits::BranchGroupTerminal with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Terminal	Terminal	The terminal to be summed.
[0..*]	[1..1] BranchGroup	BranchGroup	The branch group to which the directed branch group terminals belong.

6.6.6 CurrentLimit

Operational limit on current.

Table 149 shows all attributes of CurrentLimit.

Table 149 – Attributes of OperationalLimits::CurrentLimit

name	type	description
value	CurrentFlow	Limit on current flow.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 150 shows all association ends of CurrentLimit with other classes.

Table 150 – Association ends of OperationalLimits::CurrentLimit with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] OperationalLimitSet	OperationalLimitSet	inherited from: OperationalLimit
[0..*]	[0..1] OperationalLimitType	OperationalLimitType	inherited from: OperationalLimit
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.6.7 OperationalLimit

A value associated with a specific kind of limit.

Table 151 shows all attributes of OperationalLimit.

Table 151 – Attributes of OperationalLimits::OperationalLimit

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 152 shows all association ends of OperationalLimit with other classes.

Table 152 – Association ends of OperationalLimits::OperationalLimit with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] OperationalLimitSet	OperationalLimitSet	The limit set to which the limit values belong.
[0..*]	[0..1] OperationalLimitType	OperationalLimitType	The limit type associated with this limit.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.6.8 OperationalLimitDirectionKind enumeration

The direction of an operational limit.

Table 153 shows all literals of OperationalLimitDirectionKind.

Table 153 – Literals of OperationalLimits::OperationalLimitDirectionKind

literal	description
high	The limit is a high limit. If applied to a terminal flow, the positive direction is into the terminal.
low	The limit is a low limit. If applied to a terminal flow, the positive direction is into the terminal.
absoluteValue	If the absolute value of the monitored value is above the limit value, the limit is violated. In effect, the limit is both a high limit and its negative a low limit.

6.6.9 OperationalLimitSet

A set of limits associated with equipment. Sets of limits might apply to a specific temperature, or season for example. A set of limits may contain different severities of limit levels that would apply to the same equipment. The set may contain limits of different types such as apparent power and current limits or high and low voltage limits that are logically applied together as a set.

Table 154 shows all attributes of OperationalLimitSet.

Table 154 – Attributes of OperationalLimits::OperationalLimitSet

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 155 shows all association ends of OperationalLimitSet with other classes.

Table 155 – Association ends of OperationalLimits::OperationalLimitSet with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] Equipment	Equipment	The equipment to which the limit set applies.
[0..*]	[0..1] Terminal	Terminal	The terminal specifically associated to this operational limit set. If no terminal is associated, all terminals of the equipment are implied.
[1..1]	[0..*] OperationalLimitValue	OperationalLimit	Values of equipment limits.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.6.10 OperationalLimitType

The operational meaning of a category of limits.

Table 156 shows all attributes of OperationalLimitType.

Table 156 – Attributes of OperationalLimits::OperationalLimitType

name	type	description
acceptableDuration	Seconds	The nominal acceptable duration of the limit. Limits are commonly expressed in terms of the a time limit for which the limit is normally acceptable. The actual acceptable duration of a specific limit may depend on other local factors such as temperature or wind speed.
direction	OperationalLimitDirectionKind	The direction of the limit.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 157 shows all association ends of OperationalLimitType with other classes.

Table 157 – Association ends of OperationalLimits::OperationalLimitType with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] OperationalLimit	OperationalLimit	The operational limits associated with this type of limit.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.6.11 VoltageLimit

Operational limit applied to voltage.

Table 158 shows all attributes of VoltageLimit.

Table 158 – Attributes of OperationalLimits::VoltageLimit

name	type	description
value	Voltage	Limit on voltage. High or low limit nature of the limit depends upon the properties of the operational limit type.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 159 shows all association ends of VoltageLimit with other classes.

Table 159 – Association ends of OperationalLimits::VoltageLimit with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] OperationalLimitSet	OperationalLimitSet	inherited from: OperationalLimit
[0..*]	[0..1] OperationalLimitType	OperationalLimitType	inherited from: OperationalLimit
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.7 Package Topology

6.7.1 General

An extension to the Core Package that in association with the Terminal class models Connectivity, that is the physical definition of how equipment is connected together. In addition it models Topology, that is the logical definition of how equipment is connected via closed switches. The Topology definition is independent of the other electrical characteristics.

Figure 42 shows class diagram TopologicalNodeTerminal.

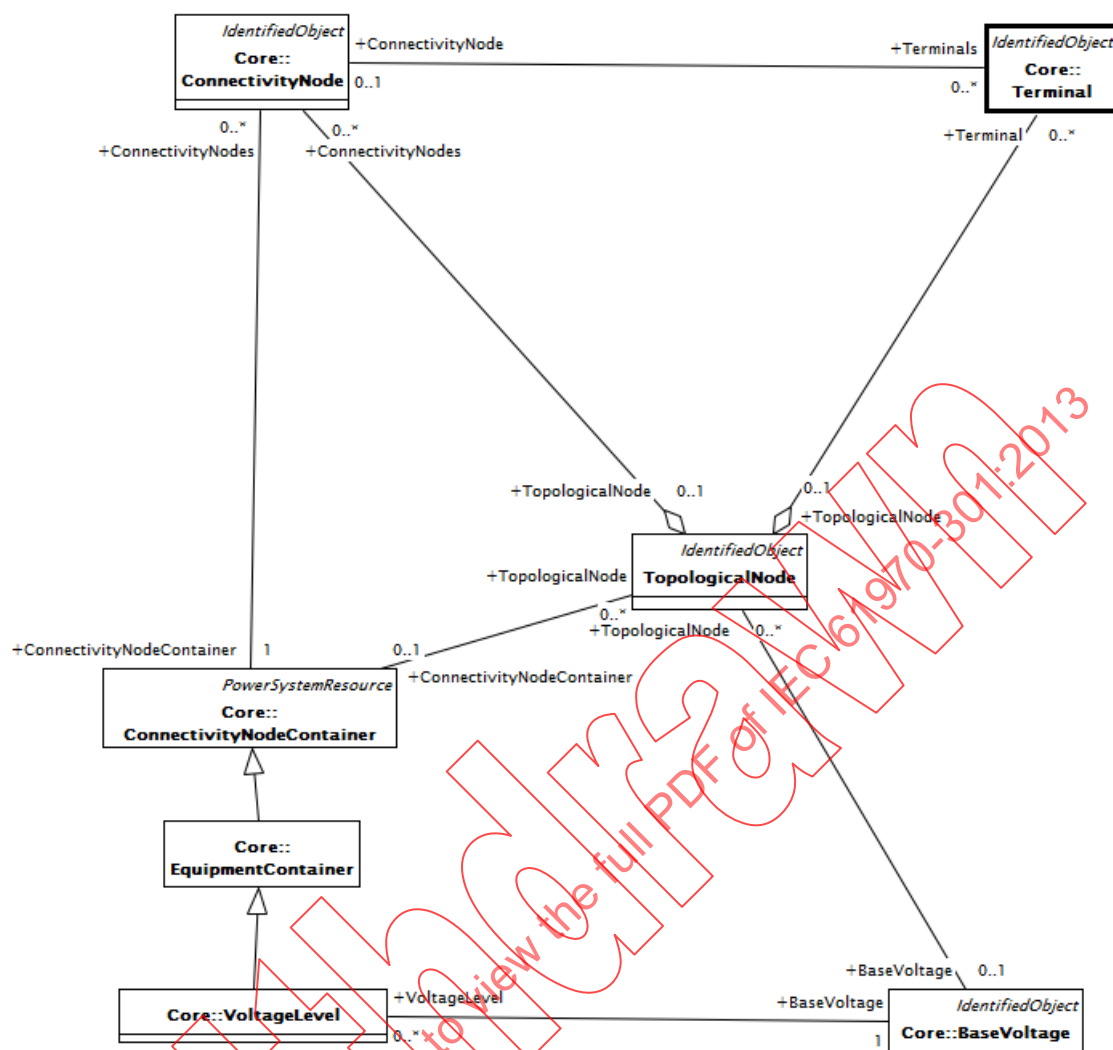


Figure 42 – Class diagram Topology::TopologicalNodeTerminal

Shows the relationships among topological nodes and terminals.

Figure 43 shows class diagram TopologyMeasRelations.

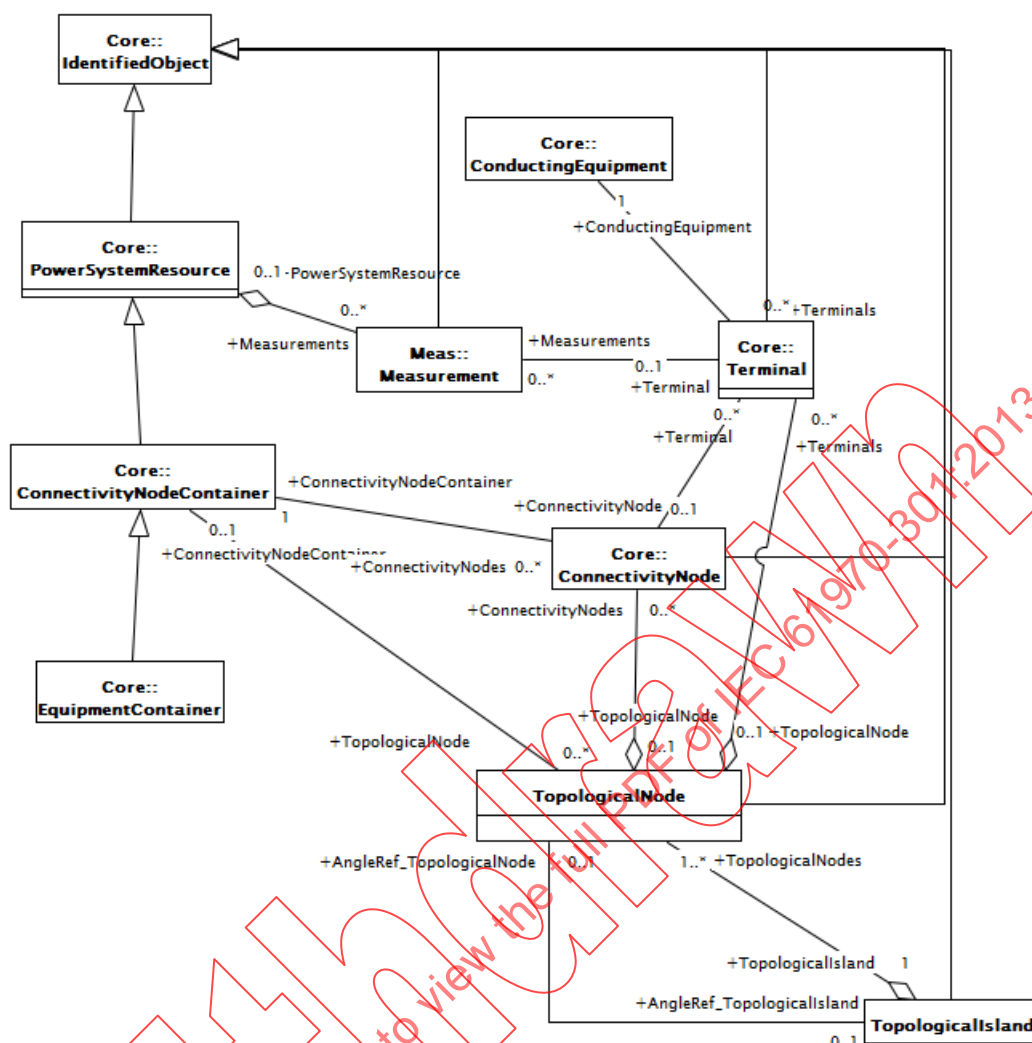


Figure 43 – Class diagram Topology::TopologyMeasRelations

Shows the relationship of measurements to terminals and topology.

Figure 44 shows class diagram TopologyReporting.

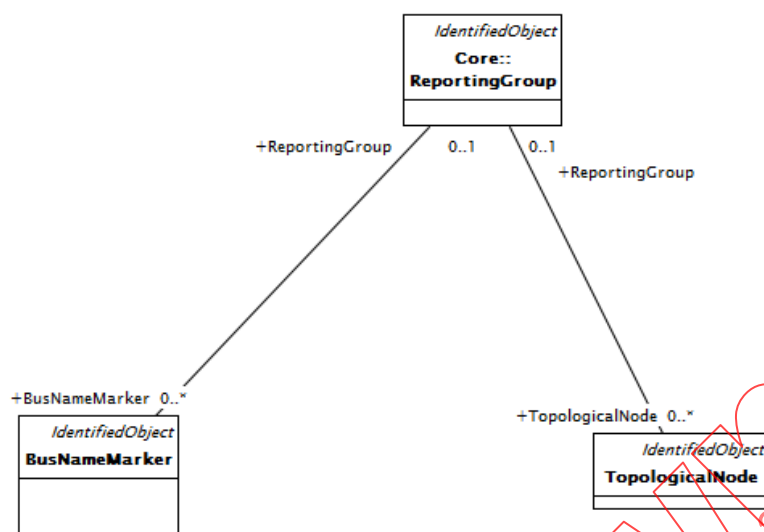


Figure 44 – Class diagram Topology::TopologyReporting

Shows topology reporting relationships.

Figure 45 shows class diagram Main.

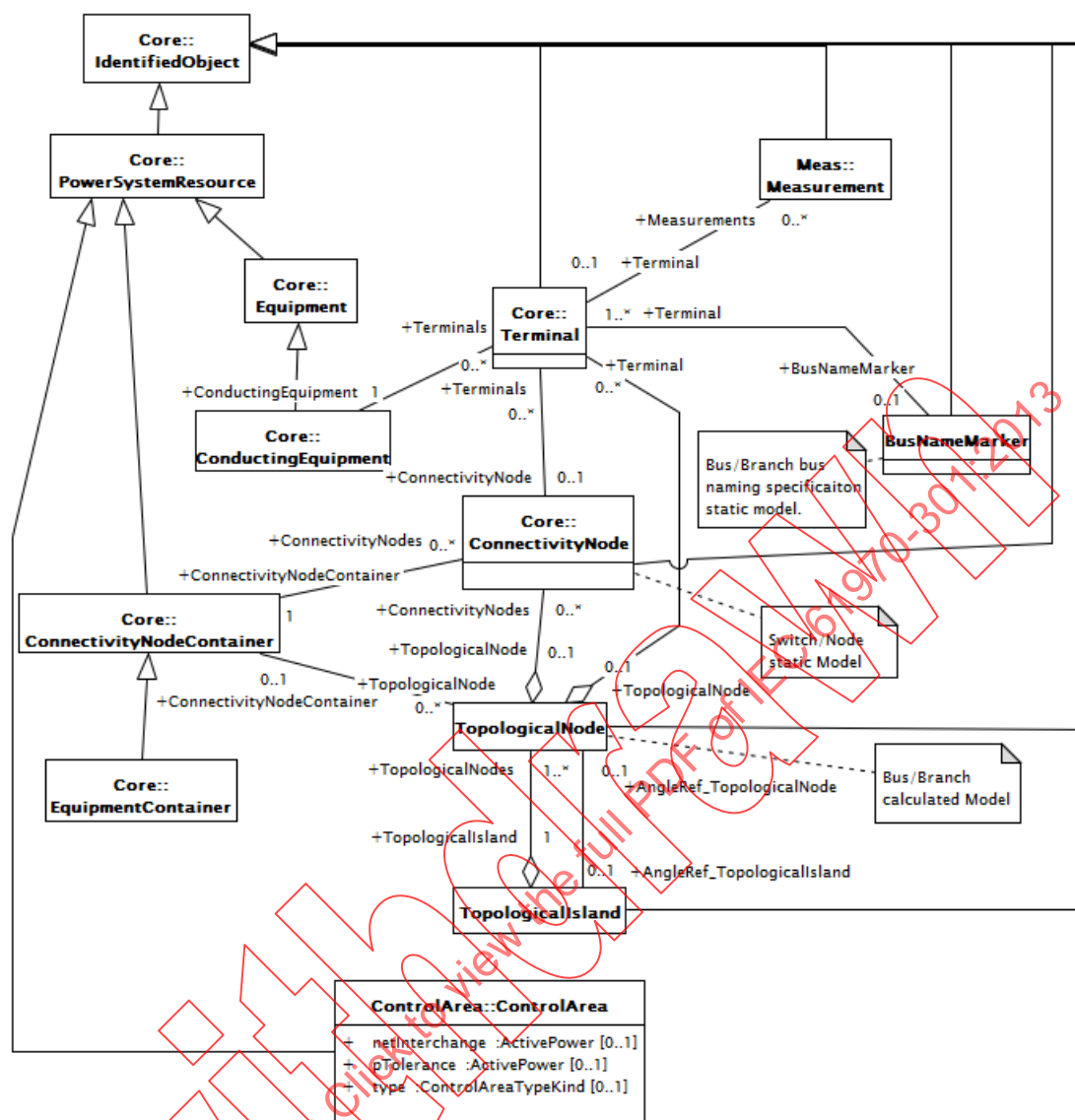


Figure 45 – Class diagram Topology::Main

This diagram shows all classes included in the Topology package as well as the key external classes that have associations with Topology classes.

6.7.2 BusNameMarker

Used to apply user standard names to topology buses. Typically used for "bus/branch" case generation. Associated with one or more **ConnectivityNodes** that are normally a part of the bus name. The associated **ConnectivityNodes** are to be connected by non-retained switches. For a ring bus station configuration, all busbar connectivity nodes in the ring are typically associated. For a breaker and a half scheme, both busbars would be associated. For a ring bus, all busbars would be associated. For a "straight" busbar configuration, only the main connectivity node at the busbar would be associated.

Table 160 shows all attributes of **BusNameMarker**.

Table 160 – Attributes of Topology::BusNameMarker

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 161 shows all association ends of BusNameMarker with other classes.

Table 161 – Association ends of Topology::BusNameMarker with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] ReportingGroup	ReportingGroup	The reporting group to which this bus name marker belongs.
[0..1]	[1..*] Terminal	Terminal	The terminals associated with this bus name marker.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.7.3 TopologicalIsland

An electrically connected subset of the network. Topological islands can change as the current network state changes: e.g. due to

- disconnect switches or breakers change state in a SCADA/EMS
- manual creation, change or deletion of topological nodes in a planning tool.

Table 162 shows all attributes of TopologicalIsland.

Table 162 – Attributes of Topology::TopologicalIsland

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 163 shows all association ends of TopologicalIsland with other classes.

Table 163 – Association ends of Topology::TopologicalIsland with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..1] AngleRef_TopologicalNode	TopologicalNode	The angle reference for the island. Normally there is one TopologicalNode that is selected as the angle reference for each island. Other reference schemes exist, so the association is typically optional.
[1..1]	[1..*] TopologicalNodes	TopologicalNode	A topological node belongs to a topological island.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.7.4 TopologicalNode

For a detailed substation model a topological node is a set of connectivity nodes that, in the current network state, are connected together through any type of closed switches, including jumpers. Topological nodes change as the current network state changes (i.e., switches, breakers, etc. change state).

For a planning model, switch statuses are not used to form topological nodes. Instead they are manually created or deleted in a model builder tool. Topological nodes maintained this way are also called "busses".

Table 164 shows all attributes of TopologicalNode.

Table 164 – Attributes of Topology::TopologicalNode

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 165 shows all association ends of TopologicalNode with other classes.

Table 165 – Association ends of Topology::TopologicalNode with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] BaseVoltage	BaseVoltage	The base voltage of the topological node.
[0..*]	[0..1] ConnectivityNodeContainer	ConnectivityNodeContainer	The connectivity node container to which the topological node belongs.
[0..*]	[0..1] ReportingGroup	ReportingGroup	The reporting group to which the topological node belongs.
[1..*]	[1..1] TopologicalIsland	TopologicalIsland	A topological node belongs to a topological island.
[0..1]	[0..*] ConnectivityNodes	ConnectivityNode	The connectivity nodes combine together to form this topological node. May depend on the current state of switches in the network.
[0..1]	[0..*] Terminal	Terminal	The terminals associated with the topological node. This can be used as an alternative to the connectivity node path to terminal, thus making it unnecessary to model connectivity nodes in some cases. Note that if connectivity nodes are in the model, this association would probably not be used as an input specification.
[0..1]	[0..1] AngleRef_TopologicalIsland	TopologicalIsland	The island for which the node is an angle reference. Normally there is one angle reference node for each island.
[1..1]	[0..1] SvVoltage	SvVoltage	The state voltage associated with the topological node.
[1..1]	[0..1] SvInjection	SvInjection	The injection flows state variables associated with the topological node.
[1..1]	[0..1] SvShortCircuit	SvShortCircuit	The state variables for short circuit associated with the topological node.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8 Package Wires

6.8.1 General

An extension to the Core and Topology package that models information on the electrical characteristics of Transmission and Distribution networks. This package is used by network applications such as State Estimation, Load Flow and Optimal Power Flow.

Figure 46 shows class diagram CutsAndJumpers.

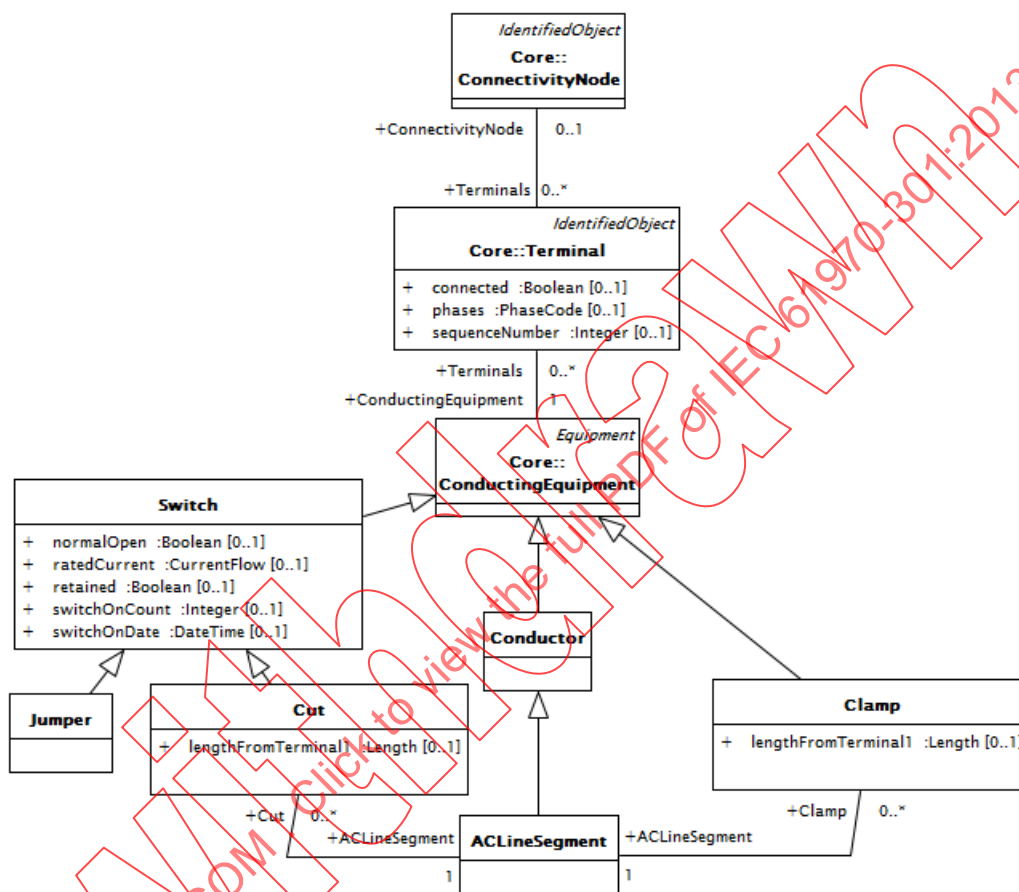


Figure 46 – Class diagram Wires::CutsAndJumpers

The diagram show the cuts and jumpers data model.

Figure 47 shows class diagram MutualCoupling.

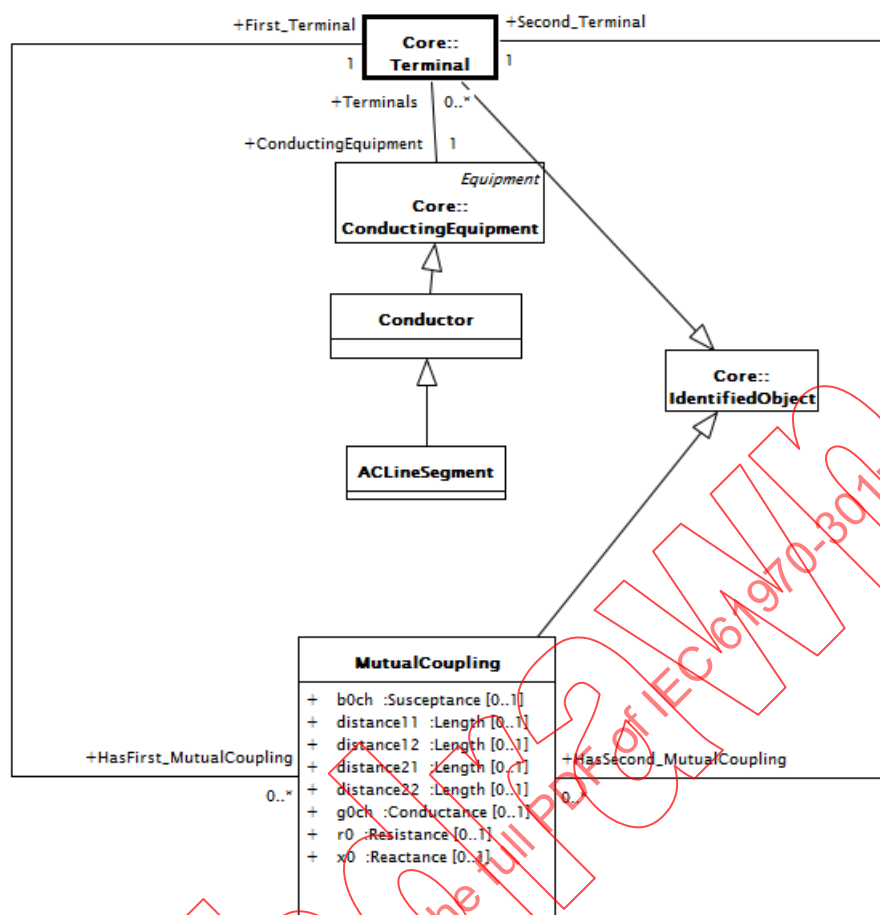


Figure 47 – Class diagram Wires::MutualCoupling

Shows mutual coupling models.

Figure 48 shows class diagram Schedules.

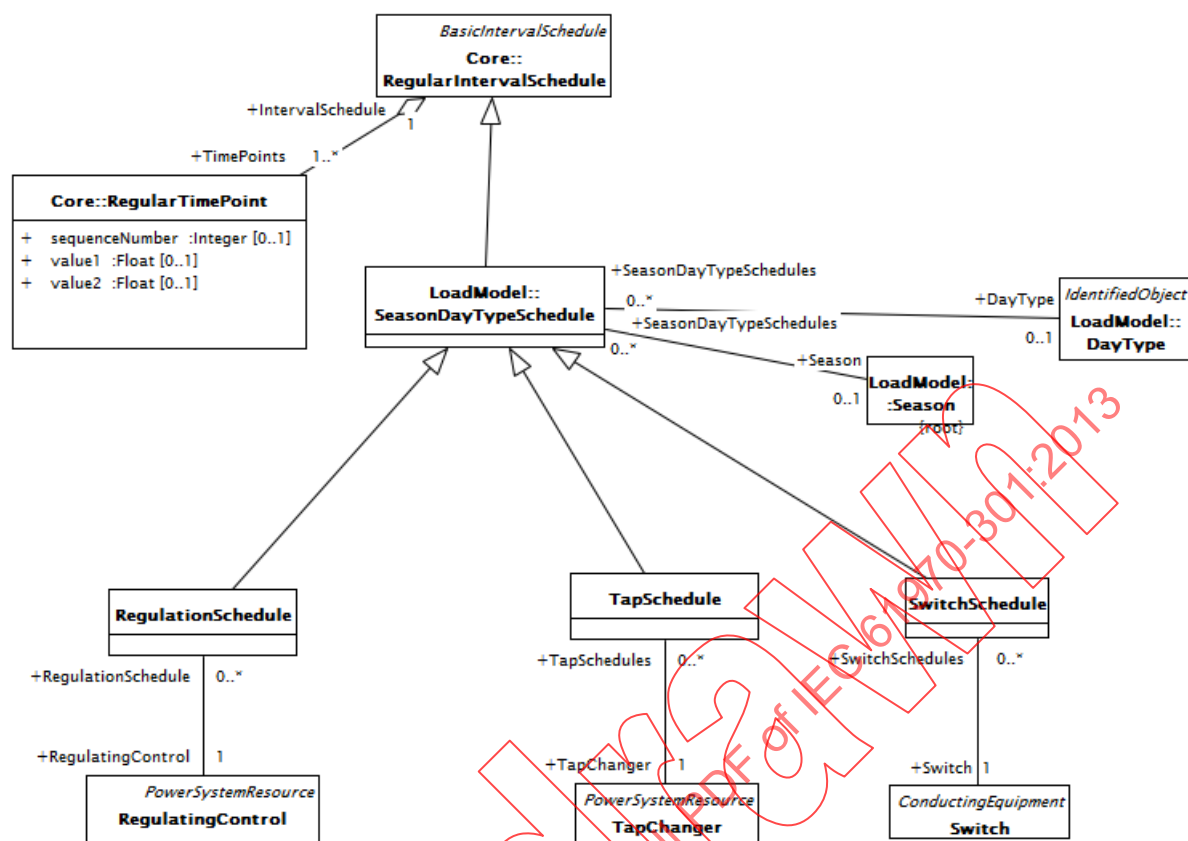


Figure 48 – Class diagram Wires::Schedules

Shows schedules of wires model.

Figure 49 shows class diagram SwitchingEquipment.

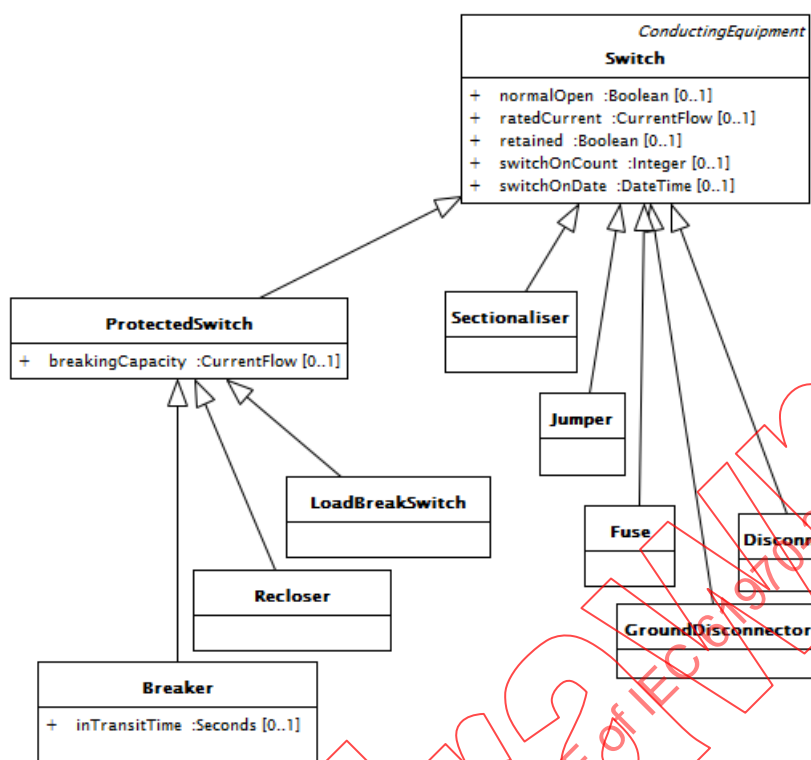


Figure 49 – Class diagram Wires::SwitchingEquipment

Shows switching equipment inheritance structure.

Figure 50 shows class diagram WiresPhaseModel.

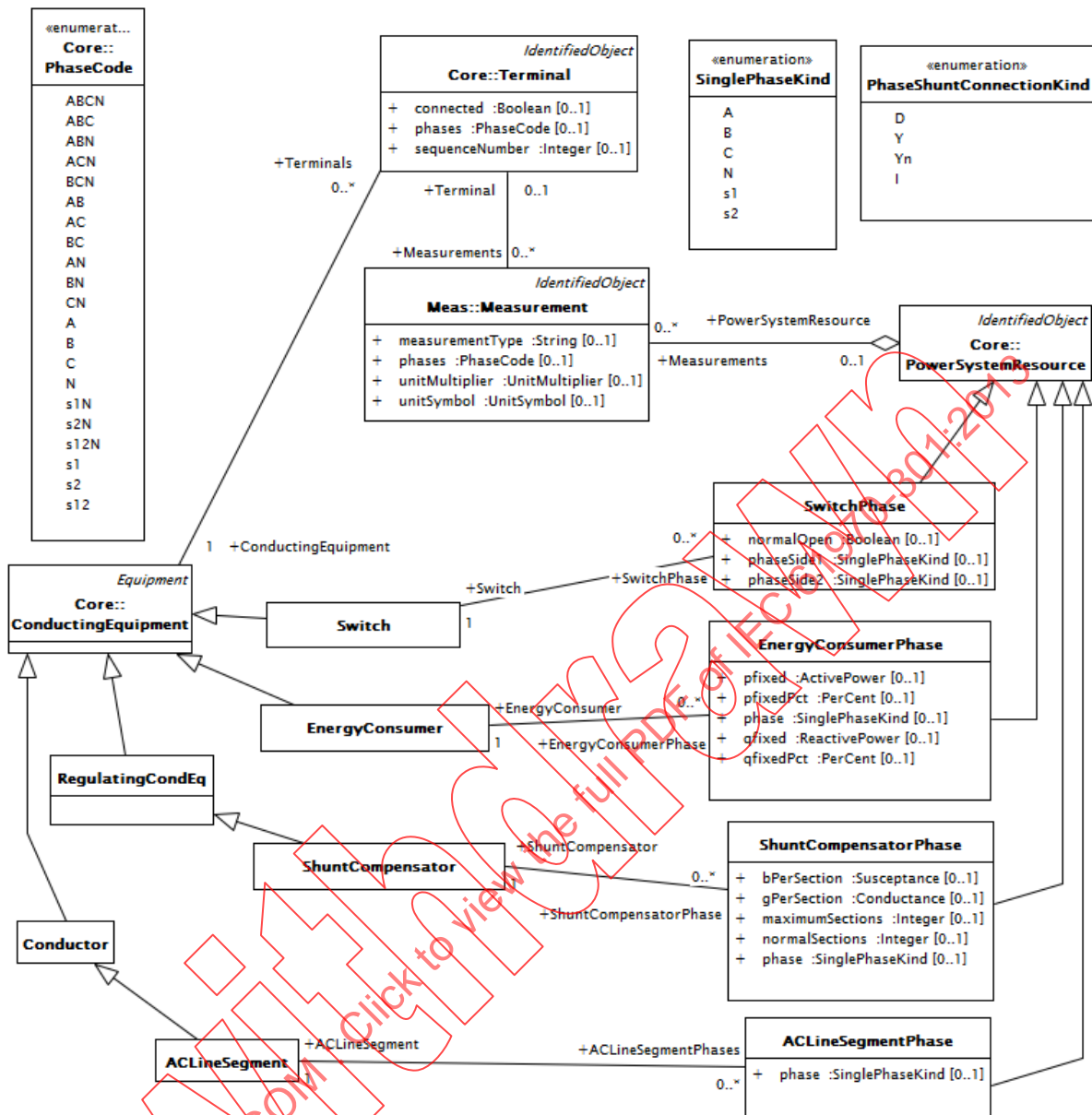


Figure 50 – Class diagram Wires::WiresPhaseModel

Shows the phase connection detail when modeling devices down to the phase components within a conducting equipment.

Figure 51 shows class diagram Datatypes.

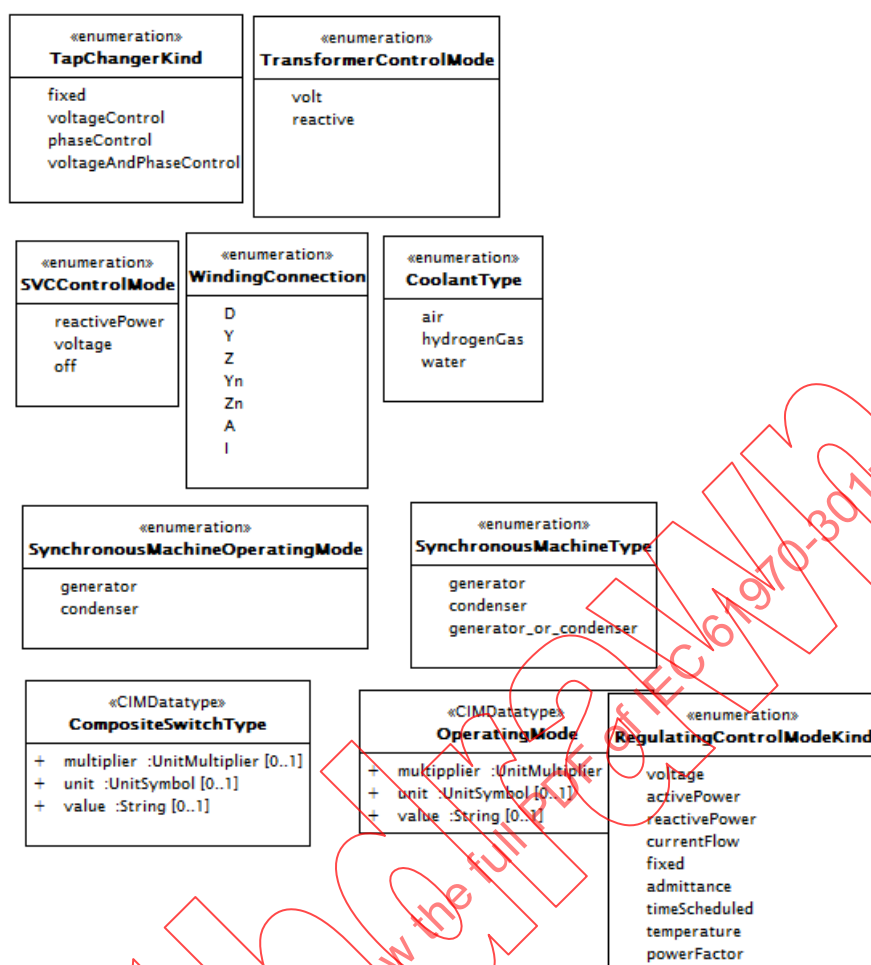


Figure 51 – Class diagram Wires::Datatypes

This diagram shows the data types specific to the Wires package.

Figure 52 shows class diagram InheritanceHierarchy.

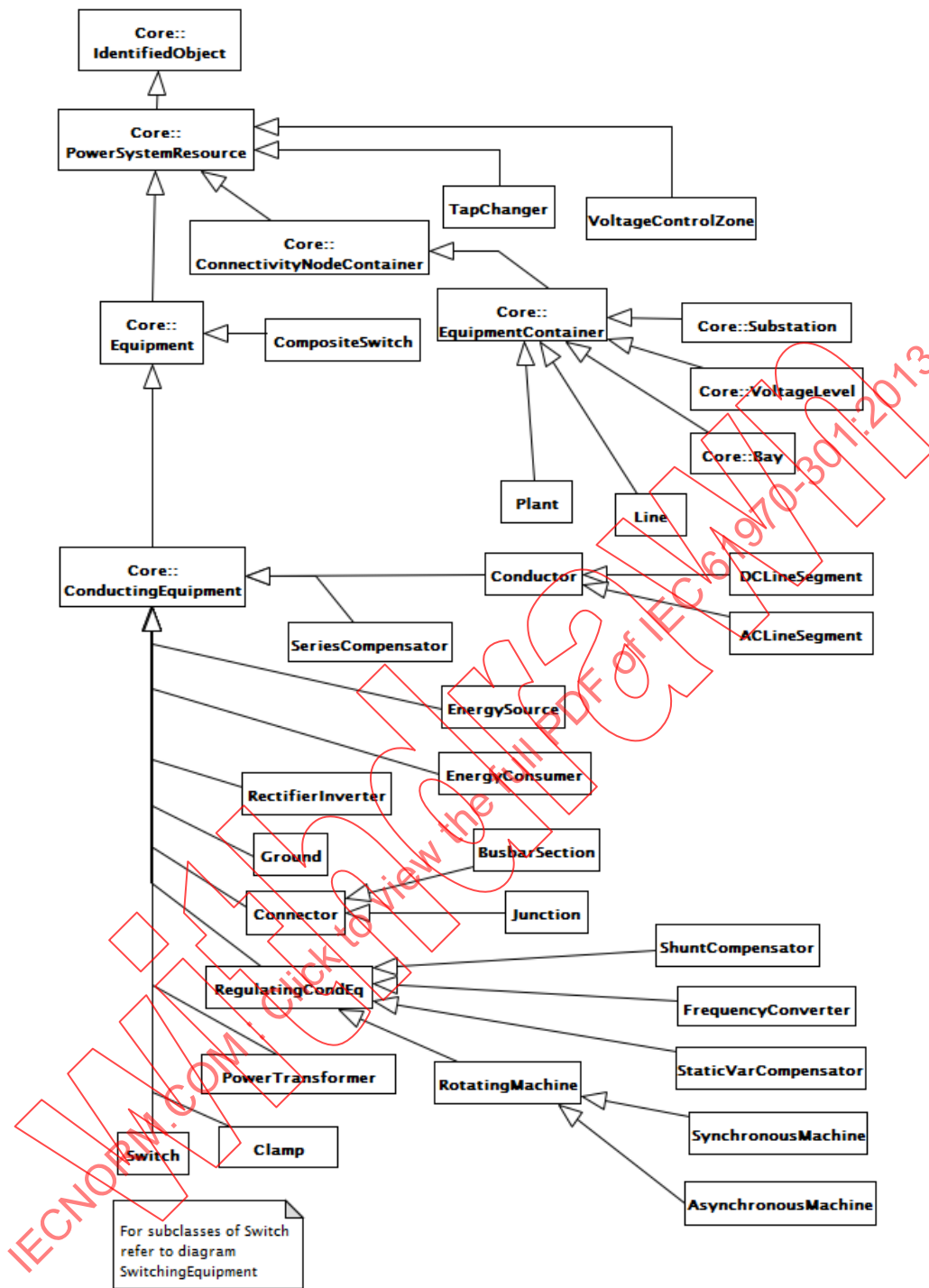


Figure 52 – Class diagram Wires::InheritanceHierarchy

This diagram describes inheritance between classes in and related to the Wires package.

Figure 53 shows class diagram LineModel.

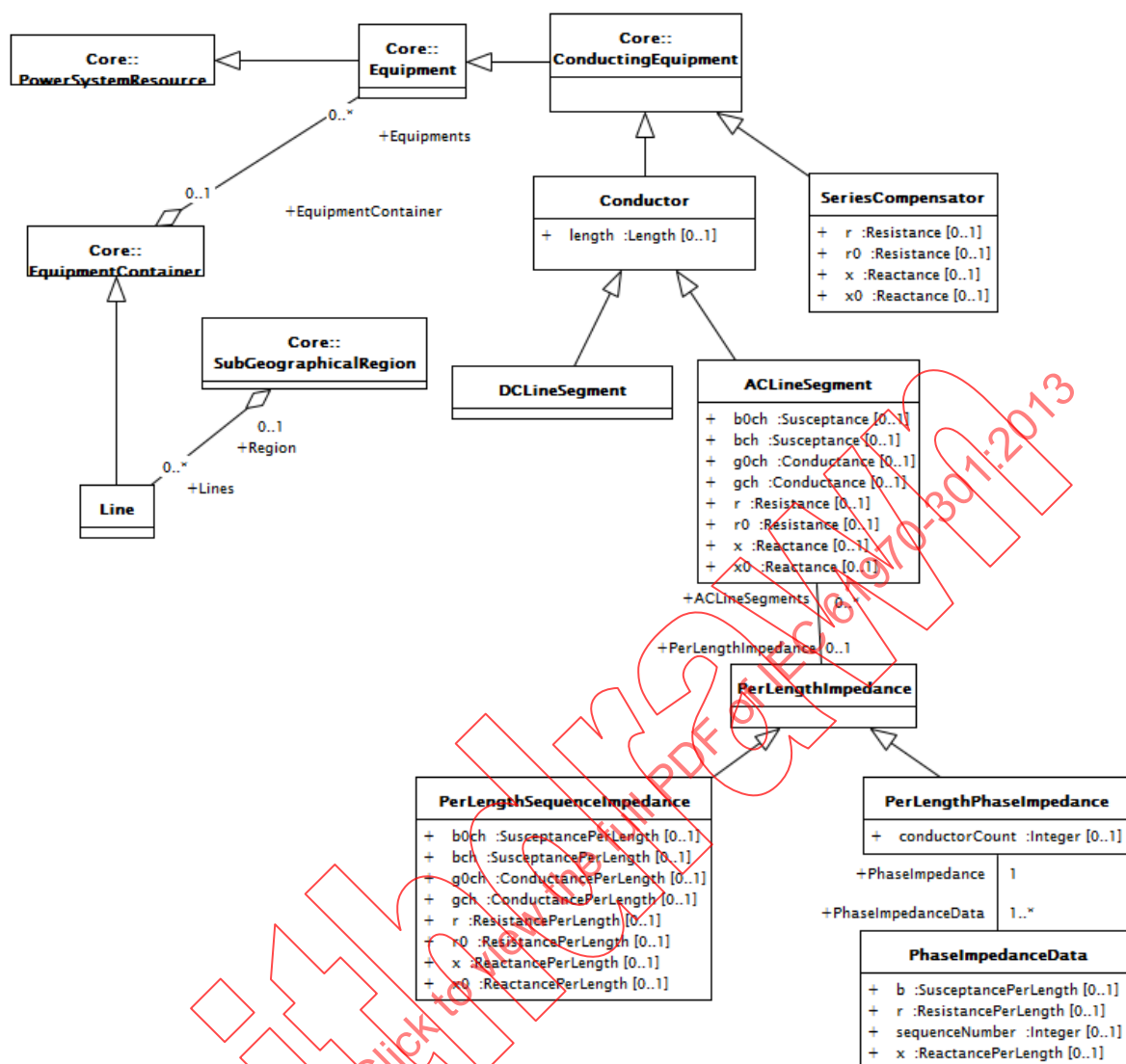


Figure 53 – Class diagram Wires::LineModel

This diagram shows all classes related to the transmission line model.

Figure 54 shows class diagram NamingHierarchyPart1.

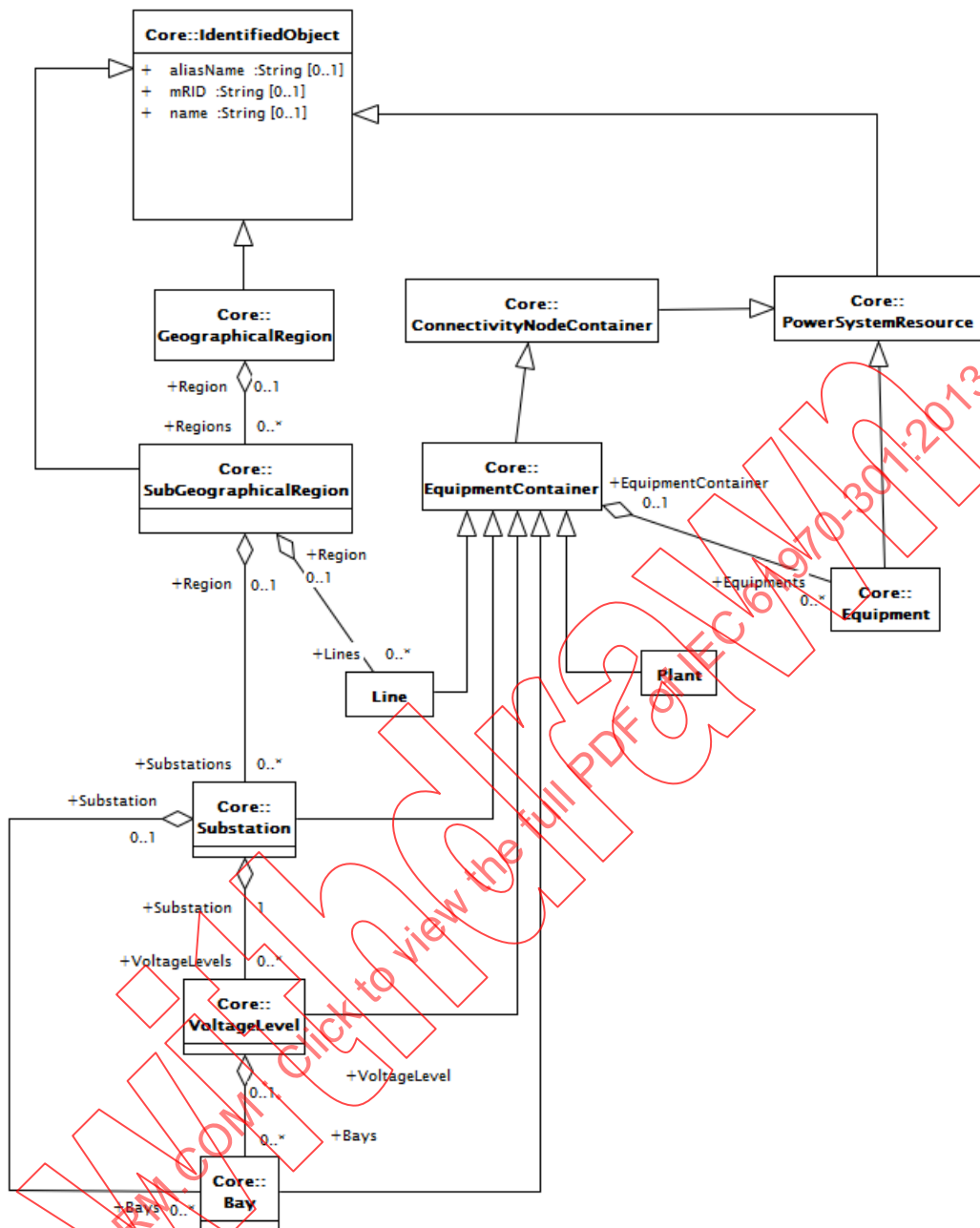


Figure 54 – Class diagram Wires::NamingHierarchyPart1

The diagram shows the upper part of the naming hierarchy. The hierarchy is a way to organize and name equipment. Hence it is tightly related to the IdentifiedObject and its attributes. The Equipment class is further specialized into many subtypes that are all contained by subclasses of the EquipmentContainer.

Figure 55 shows class diagram NamingHierarchyPart2.

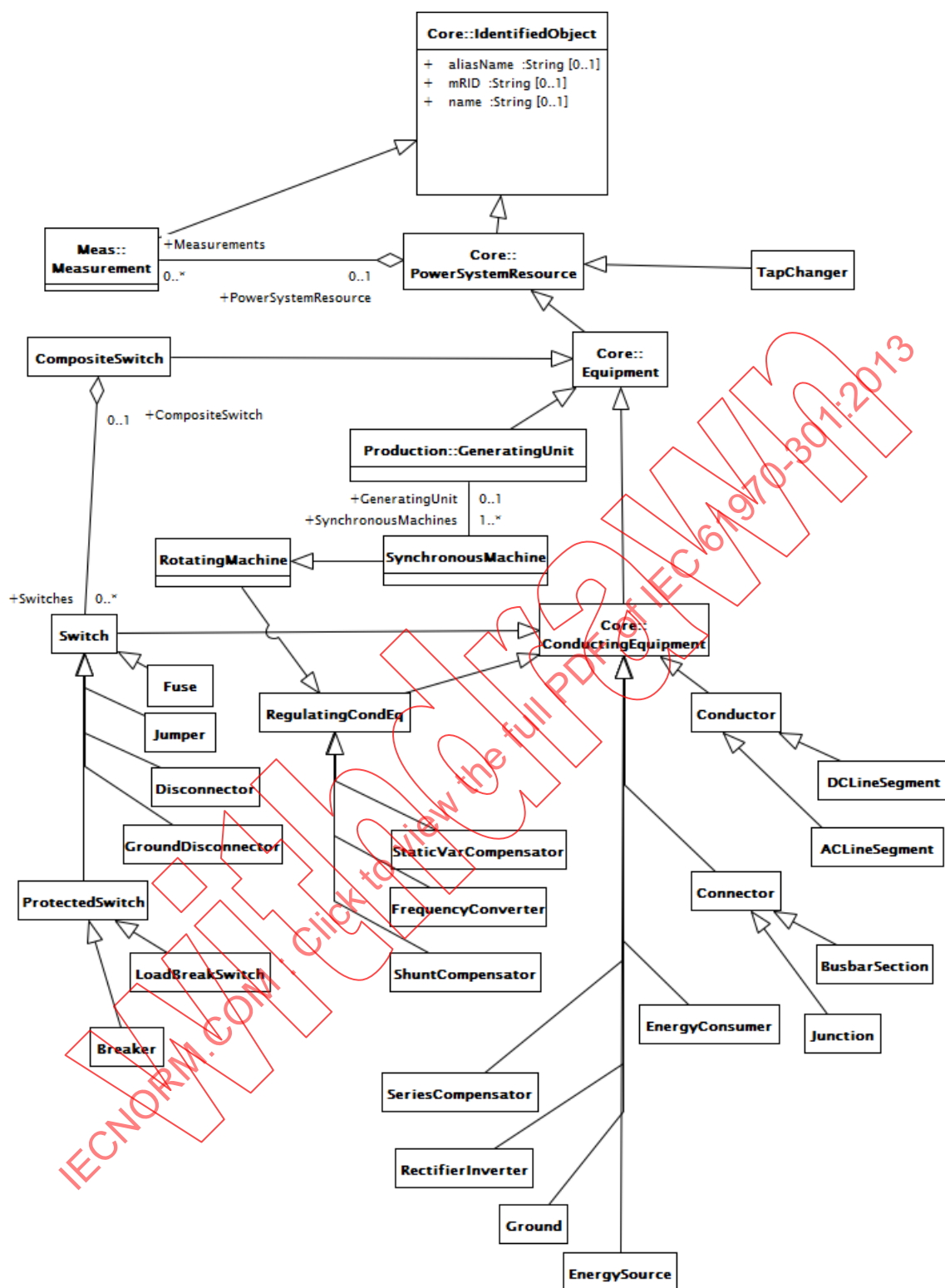


Figure 55 – Class diagram Wires::NamingHierarchyPart2

The diagram shows the lower part of the naming hierarchy. It shows how the Equipment class is further specialized into many subtypes that are all contained by subclasses of the EquipmentContainer. For the subclasses of equipment container refer to part 1 of the diagram.

Figure 56 shows class diagram RegulatingEquipment.

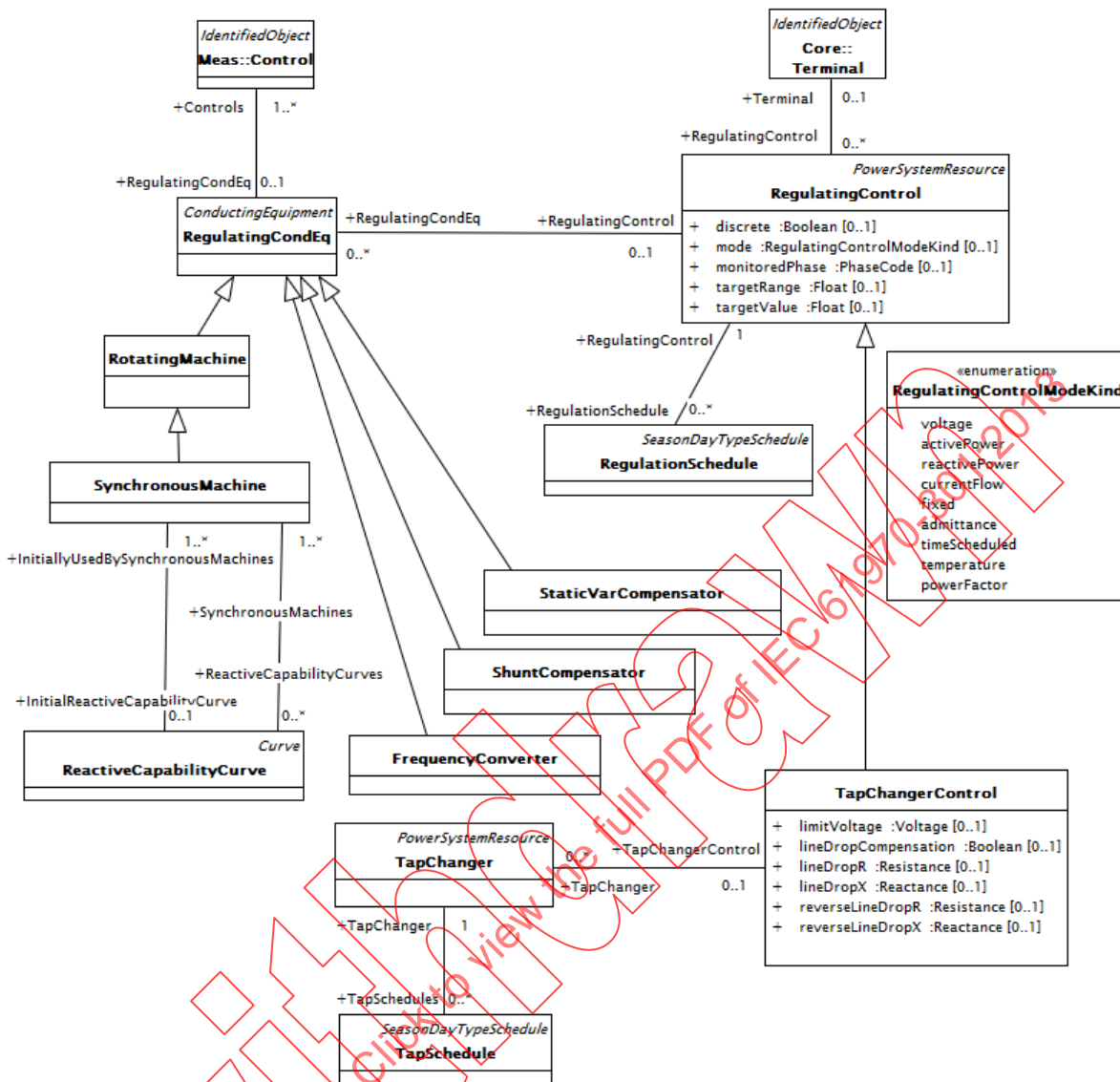


Figure 56 – Class diagram Wires::RegulatingEquipment

The diagram shows all classes related to equipment regulation and reactive power compensation.

Figure 57 shows class diagram TapChanger.

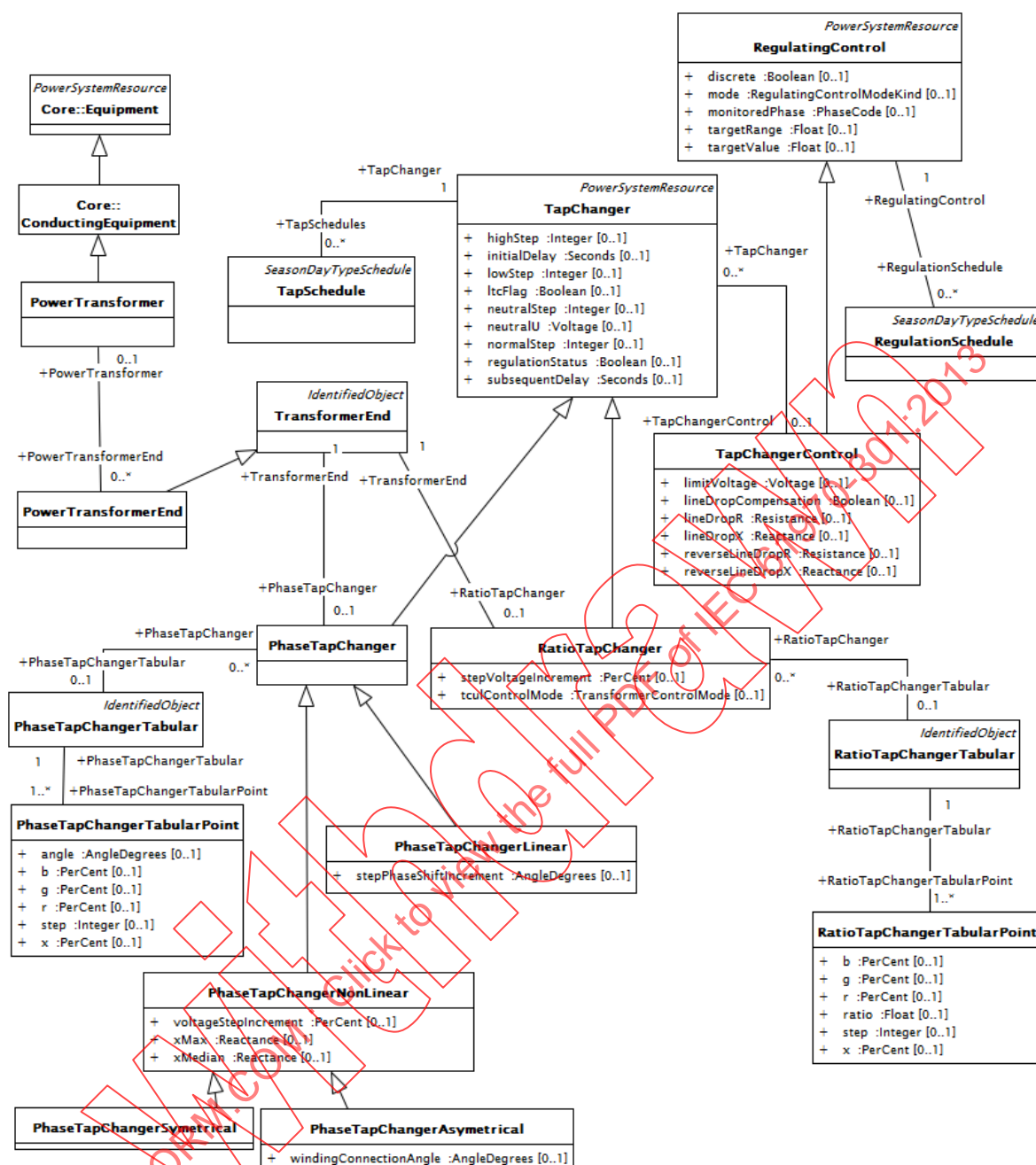


Figure 57 – Class diagram Wires::TapChanger

This diagram shows all classes related to the transformer tap model.

Figure 58 shows class diagram VoltageControl.

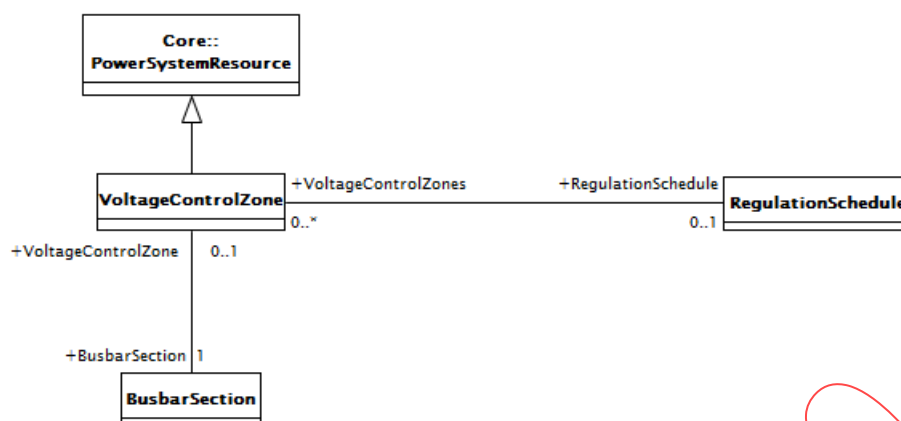


Figure 58 – Class diagram Wires::VoltageControl

This diagram shows all classes related to area voltage control.

Figure 59 shows class diagram Transformer.

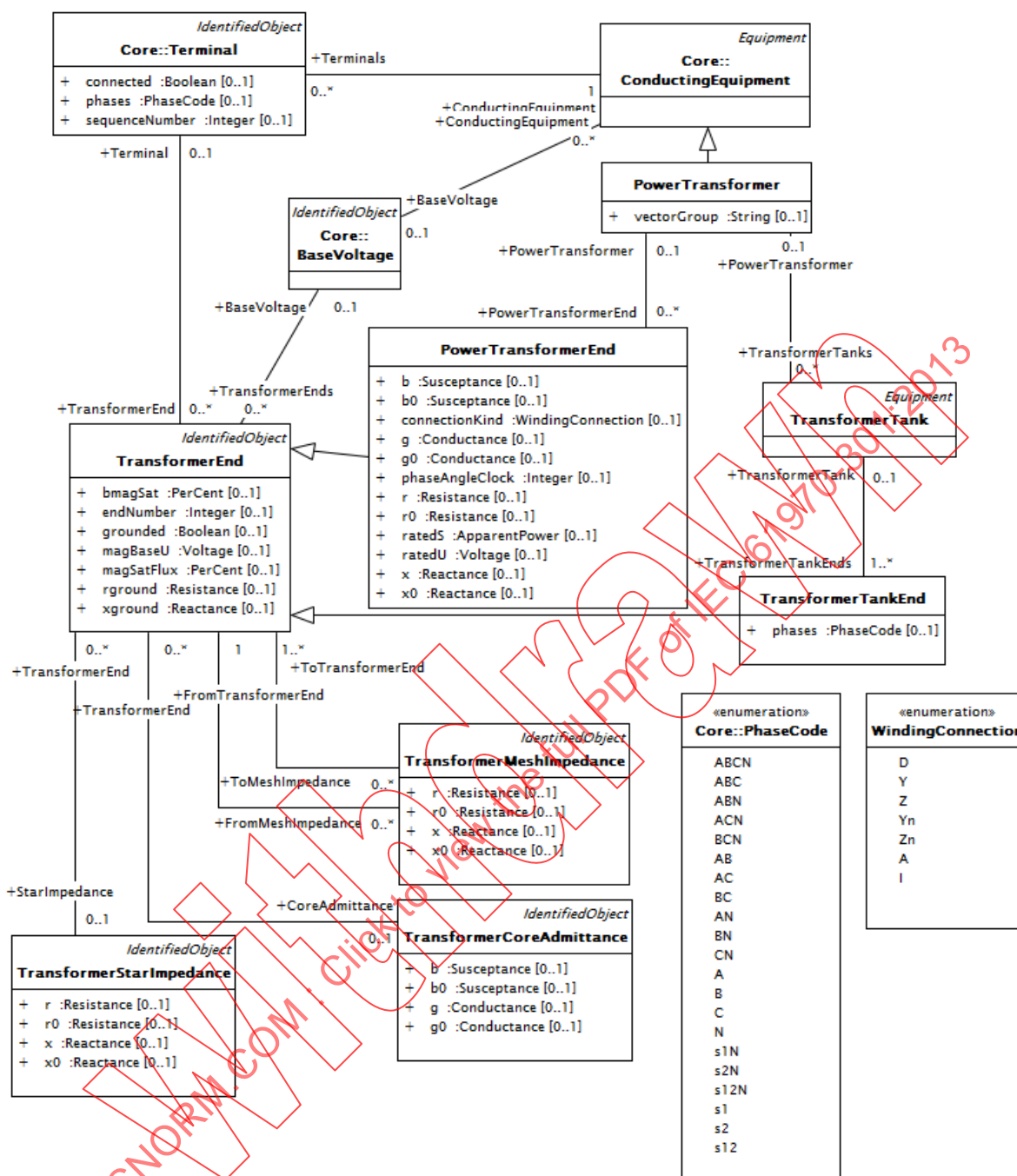


Figure 59 – Class diagram Wires::Transformer

This diagram shows classes related to the core transformer model, but does not show the tap changer details.

6.8.2 ACLineSegment

A wire or combination of wires, with consistent electrical characteristics, building a single electrical system, used to carry alternating current between points in the power system.

For symmetrical, transposed 3ph lines, it is sufficient to use attributes of the line segment, which describe impedances and admittances for the entire length of the segment. Additionally impedances can be computed by using length and associated per length impedances.

Table 166 shows all attributes of ACLineSegment.

Table 166 – Attributes of Wires::ACLineSegment

name	type	description
b0ch	Susceptance	Zero sequence shunt (charging) susceptance, uniformly distributed, of the entire line section.
bch	Susceptance	Positive sequence shunt (charging) susceptance, uniformly distributed, of the entire line section. This value represents the full charging over the full length of the line.
g0ch	Conductance	Zero sequence shunt (charging) conductance, uniformly distributed, of the entire line section.
gch	Conductance	Positive sequence shunt (charging) conductance, uniformly distributed, of the entire line section.
r	Resistance	Positive sequence series resistance of the entire line section.
r0	Resistance	Zero sequence series resistance of the entire line section.
x	Reactance	Positive sequence series reactance of the entire line section.
x0	Reactance	Zero sequence series reactance of the entire line section.
length	Length	inherited from: Conductor
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 167 shows all association ends of ACLineSegment with other classes.

Table 167 – Association ends of Wires::ACLineSegment with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] Clamp	Clamp	The clamps connected to the line segment.
[0..*]	[0..1] PerLengthImpedance	PerLengthImpedance	Per-length impedance of this line segment.
[1..1]	[0..*] Cut	Cut	Cuts applied to the line segment.
[1..1]	[0..*] ACLineSegmentPhases	ACLineSegmentPhase	The line segment phases which belong to the line segment.
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.3 ACLineSegmentPhase

Represents a single wire of an alternating current line segment.

Table 168 shows all attributes of ACLineSegmentPhase.

Table 168 – Attributes of Wires::ACLineSegmentPhase

name	type	description
phase	SinglePhaseKind	The phase connection of the wire at both ends.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 169 shows all association ends of ACLineSegmentPhase with other classes.

Table 169 – Association ends of Wires::ACLineSegmentPhase with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] ACLineSegment	ACLineSegment	The line segment to which the phase belongs.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.4 AsynchronousMachine

A rotating machine whose shaft rotates asynchronously with the electrical field. Also known as an induction machine with no external connection to the rotor windings, e.g squirrel-cage induction machine.

Table 170 shows all attributes of AsynchronousMachine.

Table 170 – Attributes of Wires::AsynchronousMachine

name	type	description
rr1	Resistance	Damper 1 winding resistance.
rr2	Resistance	Damper 2 winding resistance.
tpo	Seconds	Transient rotor time constant (greater than tppo).
tppo	Seconds	Sub-transient rotor time constant (greater than 0).
xlr1	Reactance	Damper 1 winding leakage reactance.
xlr2	Reactance	Damper 2 winding leakage reactance.
xm	Reactance	Magnetizing reactance.
xp	Reactance	Transient reactance (unsaturated) (greater than or equal to xpp).
xpp	Reactance	Sub-transient reactance (unsaturated) (greater than Xl).
xs	Reactance	Synchronous reactance (greater than xp).
damping	Float	inherited from: RotatingMachine
inertia	Seconds	inherited from: RotatingMachine
ratedS	ApparentPower	inherited from: RotatingMachine
saturationFactor	Float	inherited from: RotatingMachine
saturationFactor120	Float	inherited from: RotatingMachine
statorLeakageReactance	Reactance	inherited from: RotatingMachine
statorResistance	Resistance	inherited from: RotatingMachine
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 171 shows all association ends of AsynchronousMachine with other classes.

Table 171 – Association ends of Wires::AsynchronousMachine with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] RegulatingControl	RegulatingControl	inherited from: RegulatingCondEq
[0..1]	[1..*] Controls	Control	inherited from: RegulatingCondEq
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.5 Breaker

A mechanical switching device capable of making, carrying, and breaking currents under normal circuit conditions and also making, carrying for a specified time, and breaking currents under specified abnormal circuit conditions e.g. those of short circuit.

Table 172 shows all attributes of Breaker.

Table 172 – Attributes of Wires::Breaker

name	type	description
inTransitTime	Seconds	The transition time from open to close.
breakingCapacity	CurrentFlow	inherited from: ProtectedSwitch
normalOpen	Boolean	inherited from: Switch
ratedCurrent	CurrentFlow	inherited from: Switch
switchOnCount	Integer	inherited from: Switch
switchOnDate	DateTime	inherited from: Switch
retained	Boolean	inherited from: Switch
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 173 shows all association ends of Breaker with other classes.

Table 173 – Association ends of Wires::Breaker with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] OperatedByProtectionEquipment	ProtectionEquipment	inherited from: ProtectedSwitch
[1..1]	[0..*] RecloseSequences	RecloseSequence	inherited from: ProtectedSwitch
[1..1]	[0..*] SwitchSchedules	SwitchSchedule	inherited from: Switch
[1..1]	[0..*] SwitchPhase	SwitchPhase	inherited from: Switch
[0..*]	[0..1] CompositeSwitch	CompositeSwitch	inherited from: Switch
[0..*]	[0..*] SwitchingOperations	SwitchingOperation	inherited from: Switch
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.6 BusbarSection

A conductor, or group of conductors, with negligible impedance, that serve to connect other conducting equipment within a single substation.

Voltage measurements are typically obtained from VoltageTransformers that are connected to busbar sections. A bus bar section may have many physical terminals but for analysis is modelled with exactly one logical terminal.

Table 174 shows all attributes of BusbarSection.

Table 174 – Attributes of Wires::BusbarSection

name	type	description
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 175 shows all association ends of BusbarSection with other classes.

Table 175 – Association ends of Wires::BusbarSection with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..1] VoltageControlZone	VoltageControlZone	A VoltageControlZone is controlled by a designated BusbarSection.
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.7 Clamp

A Clamp is a galvanic connection at a line segment where other equipment is connected. A Clamp does not cut the line segment.

A Clamp is ConductingEquipment and has one Terminal with an associated ConnectivityNode. Any other ConductingEquipment can be connected to the Clamp ConnectivityNode.

Table 176 shows all attributes of Clamp.

Table 176 – Attributes of Wires::Clamp

name	type	description
lengthFromTerminal1	Length	The length to the place where the clamp is located starting from side one of the line segment, i.e. the line segment terminal with sequence number equal to 1.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 177 shows all association ends of Clamp with other classes.

Table 177 – Association ends of Wires::Clamp with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] ACLineSegment	ACLineSegment	The line segment to which the clamp is connected.
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.8 CompositeSwitch

A model of a set of individual Switches normally enclosed within the same cabinet and possibly with interlocks that restrict the combination of switch positions. These are typically found in medium voltage distribution networks.

A CompositeSwitch could represent a Ring-Main-Unit (RMU), or pad-mounted switchgear, with primitive internal devices such as an internal bus-bar plus 3 or 4 internal switches each of which may individually be open or closed. A CompositeSwitch and a set of contained Switches can also be used to represent a multi-position switch e.g. a switch that can connect a circuit to Ground, Open or Busbar.

Table 178 shows all attributes of CompositeSwitch.

Table 178 – Attributes of Wires::CompositeSwitch

name	type	description
compositeSwitchType	CompositeSwitchType	An alphanumeric code that can be used as a reference to extra information such as the description of the interlocking scheme if any.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 179 shows all association ends of CompositeSwitch with other classes.

Table 179 – Association ends of Wires::CompositeSwitch with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] Switches	Switch	Switches contained in this Composite switch.
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.9 CompositeSwitchType datatype

An alphanumeric code that can be used as a reference to extra information such as the description of the interlocking scheme if any.

Table 180 shows all attributes of CompositeSwitchType.

Table 180 – Attributes of Wires::CompositeSwitchType

name	type	description
multiplier	UnitMultiplier	Multiplier.
unit	UnitSymbol	Units.
value	String	Value.

6.8.10 Conductor

Combination of conducting material with consistent electrical characteristics, building a single electrical system, used to carry current between points in the power system.

Table 181 shows all attributes of Conductor.

Table 181 – Attributes of Wires::Conductor

name	type	description
length	Length	Segment length for calculating line section capabilities
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 182 shows all association ends of Conductor with other classes.

Table 182 – Association ends of Wires::Conductor with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.11 Connector

A conductor, or group of conductors, with negligible impedance, that serve to connect other conducting equipment within a single substation and are modelled with a single logical terminal.

Table 183 shows all attributes of Connector.

Table 183 – Attributes of Wires::Connector

name	type	description
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 184 shows all association ends of Connector with other classes.

Table 184 – Association ends of Wires::Connector with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.12 CoolantType enumeration

Method of cooling a machine.

Table 185 shows all literals of CoolantType.

Table 185 – Literals of Wires::CoolantType

literal	description
air	Air.
hydrogenGas	Hydrogen gas.
water	Water.

6.8.13 Cut

A cut separates a line segment into two parts. The cut appears as a switch inserted between these two parts and connects them together. As the cut is normally open there is no galvanic connection between the two line segment parts. But it is possible to close the cut to get galvanic connection.

The cut terminals are oriented towards the line segment terminals with the same sequence number. Hence the cut terminal with sequence number equal to 1 is oriented to the line segment's terminal with sequence number equal to 1.

The cut terminals also act as connection points for jumpers and other equipment, e.g. a mobile generator. To enable this, connectivity nodes are placed at the cut terminals. Once the connectivity nodes are in place any conducting equipment can be connected at them.

Table 186 shows all attributes of Cut.

Table 186 – Attributes of Wires::Cut

name	type	description
lengthFromTerminal1	Length	The length to the place where the cut is located starting from side one of the cut line segment, i.e. the line segment Terminal with sequenceNumber equal to 1.
normalOpen	Boolean	inherited from: Switch
ratedCurrent	CurrentFlow	inherited from: Switch
switchOnCount	Integer	inherited from: Switch
switchOnDate	DateTime	inherited from: Switch
retained	Boolean	inherited from: Switch
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 187 shows all association ends of Cut with other classes.

Table 187 – Association ends of Wires::Cut with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] ACLineSegment	ACLineSegment	The line segment to which the cut is applied.
[1..1]	[0..*] SwitchSchedules	SwitchSchedule	inherited from: Switch
[1..1]	[0..*] SwitchPhase	SwitchPhase	inherited from: Switch
[0..*]	[0..1] CompositeSwitch	CompositeSwitch	inherited from: Switch
[0..*]	[0..*] SwitchingOperations	SwitchingOperation	inherited from: Switch
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.14 DCLineSegment

A wire or combination of wires not insulated from one another, with consistent electrical characteristics, used to carry direct current between points in the DC region of the power system.

Table 188 shows all attributes of DCLineSegment.

Table 188 – Attributes of Wires::DCLineSegment

name	type	description
dcSegmentInductance	Inductance	Inductance of the DC line segment.
dcSegmentResistance	Resistance	Resistance of the DC line segment.
length	Length	inherited from: Conductor
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 189 shows all association ends of DCLineSegment with other classes.

Table 189 – Association ends of Wires::DCLineSegment with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.15 Disconnector

A manually operated or motor operated mechanical switching device used for changing the connections in a circuit, or for isolating a circuit or equipment from a source of power. It is required to open or close circuits when negligible current is broken or made.

Table 190 shows all attributes of Disconnector.

Table 190 – Attributes of Wires::Disconnector

name	type	description
normalOpen	Boolean	inherited from: Switch
ratedCurrent	CurrentFlow	inherited from: Switch
switchOnCount	Integer	inherited from: Switch
switchOnDate	DateTime	inherited from: Switch
retained	Boolean	inherited from: Switch
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 191 shows all association ends of Disconnector with other classes.

Table 191 – Association ends of Wires::Disconnector with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] SwitchSchedules	SwitchSchedule	inherited from: Switch
[1..1]	[0..*] SwitchPhase	SwitchPhase	inherited from: Switch
[0..*]	[0..1] CompositeSwitch	CompositeSwitch	inherited from: Switch
[0..*]	[0..*] SwitchingOperations	SwitchingOperation	inherited from: Switch
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.16 EnergyConsumer

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

Table 192 shows all attributes of EnergyConsumer.

Table 192 – Attributes of Wires::EnergyConsumer

name	type	description
customerCount	Integer	Number of individual customers represented by this demand.
grounded	WindingConnection	Used for Yn and Zn connections. True if the neutral is solidly grounded.
pfixed	ActivePower	Active power of the load that is a fixed quantity. Load sign convention is used, i.e. positive sign means flow out from a node.
pfixedPct	PerCent	Fixed active power as per cent of load group fixed active power. Load sign convention is used, i.e. positive sign means flow out from a node.
phaseConnection	PhaseShuntConnectionKind	The type of phase connection, such as wye or delta.
qfixed	ReactivePower	Reactive power of the load that is a fixed quantity. Load sign convention is used, i.e. positive sign means flow out from a node.
qfixedPct	PerCent	Fixed reactive power as per cent of load group fixed reactive power. Load sign convention is used, i.e. positive sign means flow out from a node.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 193 shows all association ends of EnergyConsumer with other classes.

Table 193 – Association ends of Wires::EnergyConsumer with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] LoadResponse	LoadResponseCharacteristic	The load response characteristic of this load. If missing, this load is assumed to be constant power.
[1..1]	[0..*] EnergyConsumerPhase	EnergyConsumerPhase	The individual phase models for this energy consumer.
[1..*]	[0..1] PowerCutZone	PowerCutZone	The energy consumer is assigned to this power cut zone.
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..1] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.17 EnergyConsumerPhase

A single phase of an energy consumer.

Table 194 shows all attributes of EnergyConsumerPhase.

Table 194 – Attributes of Wires::EnergyConsumerPhase

name	type	description
pfixed	ActivePower	Active power of the load that is a fixed quantity. Load sign convention is used, i.e. positive sign means flow out from a node.
pfixedPct	PerCent	Fixed active power as per cent of load group fixed active power. Load sign convention is used, i.e. positive sign means flow out from a node.
phase	SinglePhaseKind	Phase of this energy consumer component. If the energy consumer is wye connected, the connection is from the indicated phase to the central ground or neutral point. If the energy consumer is delta connected, the phase indicates an energy consumer connected from the indicated phase to the next logical non-neutral phase.
qfixed	ReactivePower	Reactive power of the load that is a fixed quantity. Load sign convention is used, i.e. positive sign means flow out from a node.
qfixedPct	PerCent	Fixed reactive power as per cent of load group fixed reactive power. Load sign convention is used, i.e. positive sign means flow out from a node.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 195 shows all association ends of EnergyConsumerPhase with other classes.

Table 195 – Association ends of Wires::EnergyConsumerPhase with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] EnergyConsumer	EnergyConsumer	The energy consumer to which this phase belongs.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.18 EnergySource

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

Table 196 shows all attributes of EnergySource.

Table 196 – Attributes of Wires::EnergySource

name	type	description
activePower	ActivePower	High voltage source load.
nominalVoltage	Voltage	Phase-to-phase nominal voltage.
r	Resistance	Positive sequence Thevenin resistance.
r0	Resistance	Zero sequence Thevenin resistance.
rn	Resistance	Negative sequence Thevenin resistance.
voltageAngle	AngleRadians	Phase angle of a phase open circuit.
voltageMagnitude	Voltage	Phase-to-phase open circuit voltage magnitude.
x	Reactance	Positive sequence Thevenin reactance.
x0	Reactance	Zero sequence Thevenin reactance.
xn	Reactance	Negative sequence Thevenin reactance.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 197 shows all association ends of EnergySource with other classes.

Table 197 – Association ends of Wires::EnergySource with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.19 FrequencyConverter

A device to convert from one frequency to another (e.g., frequency F1 to F2) comprises a pair of FrequencyConverter instances. One converts from F1 to DC, the other converts the DC to F2.

Table 198 shows all attributes of FrequencyConverter.

Table 198 – Attributes of Wires::FrequencyConverter

name	type	description
frequency	Frequency	Frequency on the AC side.
maxP	ActivePower	The maximum active power on the DC side at which the frequency converter should operate.
maxU	Voltage	The maximum voltage on the DC side at which the frequency converter should operate.
minP	ActivePower	The minimum active power on the DC side at which the frequency converter should operate.
minU	Voltage	The minimum voltage on the DC side at which the frequency converter should operate.
operatingMode	OperatingMode	Operating mode for the frequency converter
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 199 shows all association ends of FrequencyConverter with other classes.

Table 199 – Association ends of Wires::FrequencyConverter with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] RegulatingControl	RegulatingControl	inherited from: RegulatingCondEq
[0..1]	[1..*] Controls	Control	inherited from: RegulatingCondEq
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.20 Fuse

An overcurrent protective device with a circuit opening fusible part that is heated and severed by the passage of overcurrent through it. A fuse is considered a switching device because it breaks current.

Table 200 shows all attributes of Fuse.

Table 200 – Attributes of Wires::Fuse

name	type	description
normalOpen	Boolean	inherited from: Switch
ratedCurrent	CurrentFlow	inherited from: Switch
switchOnCount	Integer	inherited from: Switch
switchOnDate	DateTime	inherited from: Switch
retained	Boolean	inherited from: Switch
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 201 shows all association ends of Fuse with other classes.

Table 201 – Association ends of Wires::Fuse with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] SwitchSchedules	SwitchSchedule	inherited from: Switch
[1..1]	[0..*] SwitchPhase	SwitchPhase	inherited from: Switch
[0..*]	[0..1] CompositeSwitch	CompositeSwitch	inherited from: Switch
[0..*]	[0..*] SwitchingOperations	SwitchingOperation	inherited from: Switch
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.21 Ground

A common point for connecting grounded conducting equipment such as shunt capacitors. The power system model can have more than one ground.

Table 202 shows all attributes of Ground.

Table 202 – Attributes of Wires::Ground

name	type	description
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 203 shows all association ends of Ground with other classes.

Table 203 – Association ends of Wires::Ground with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.22 GroundDisconnector

A manually operated or motor operated mechanical switching device used for isolating a circuit or equipment from Ground.

Table 204 shows all attributes of GroundDisconnector.

Table 204 – Attributes of Wires::GroundDisconnector

name	type	description
normalOpen	Boolean	inherited from: Switch
ratedCurrent	CurrentFlow	inherited from: Switch
switchOnCount	Integer	inherited from: Switch
switchOnDate	DateTime	inherited from: Switch
retained	Boolean	inherited from: Switch
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 205 shows all association ends of GroundDisconnector with other classes.

Table 205 – Association ends of Wires::GroundDisconnector with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] SwitchSchedules	SwitchSchedule	inherited from: Switch
[1..1]	[0..*] SwitchPhase	SwitchPhase	inherited from: Switch
[0..*]	[0..1] CompositeSwitch	CompositeSwitch	inherited from: Switch
[0..*]	[0..*] SwitchingOperations	SwitchingOperation	inherited from: Switch
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.23 Jumper

A short section of conductor with negligible impedance which can be manually removed and replaced if the circuit is de-energized. Note that zero-impedance branches can potentially be modeled by other equipment types.

Table 206 shows all attributes of Jumper.

Table 206 – Attributes of Wires::Jumper

name	type	description
normalOpen	Boolean	inherited from: Switch
ratedCurrent	CurrentFlow	inherited from: Switch
switchOnCount	Integer	inherited from: Switch
switchOnDate	DateTime	inherited from: Switch
retained	Boolean	inherited from: Switch
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 207 shows all association ends of Jumper with other classes.

Table 207 – Association ends of Wires::Jumper with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] SwitchSchedules	SwitchSchedule	inherited from: Switch
[1..1]	[0..*] SwitchPhase	SwitchPhase	inherited from: Switch
[0..*]	[0..1] CompositeSwitch	CompositeSwitch	inherited from: Switch
[0..*]	[0..*] SwitchingOperations	SwitchingOperation	inherited from: Switch
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.24 Junction

A point where one or more conducting equipments are connected with zero resistance.

Table 208 shows all attributes of Junction.

Table 208 – Attributes of Wires::Junction

name	type	description
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 209 shows all association ends of Junction with other classes.

Table 209 – Association ends of Wires::Junction with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.25 Line

Contains equipment beyond a substation belonging to a power transmission line.

Table 210 shows all attributes of Line.

Table 210 – Attributes of Wires::Line

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 211 shows all association ends of Line with other classes.

Table 211 – Association ends of Wires::Line with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] Region	SubGeographicalRegion	The sub-geographical region of the line.
[0..1]	[0..*] Equipments	Equipment	inherited from: EquipmentContainer
[1..1]	[0..*] ConnectivityNodes	ConnectivityNode	inherited from: ConnectivityNodeContainer
[0..1]	[0..*] TopologicalNode	TopologicalNode	inherited from: ConnectivityNodeContainer
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.26 LoadBreakSwitch

A mechanical switching device capable of making, carrying, and breaking currents under normal operating conditions.

Table 212 shows all attributes of LoadBreakSwitch.

Table 212 – Attributes of Wires::LoadBreakSwitch

name	type	description
breakingCapacity	CurrentFlow	inherited from: ProtectedSwitch
normalOpen	Boolean	inherited from: Switch
ratedCurrent	CurrentFlow	inherited from: Switch
switchOnCount	Integer	inherited from: Switch
switchOnDate	DateTime	inherited from: Switch
retained	Boolean	inherited from: Switch
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 213 shows all association ends of LoadBreakSwitch with other classes.

Table 213 – Association ends of Wires::LoadBreakSwitch with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] OperatedByProtectionEquipment	ProtectionEquipment	inherited from: ProtectedSwitch
[1..1]	[0..*] RecloseSequences	RecloseSequence	inherited from: ProtectedSwitch
[1..1]	[0..*] SwitchSchedules	SwitchSchedule	inherited from: Switch
[1..1]	[0..*] SwitchPhase	SwitchPhase	inherited from: Switch
[0..*]	[0..1] CompositeSwitch	CompositeSwitch	inherited from: Switch
[0..*]	[0..*] SwitchingOperations	SwitchingOperation	inherited from: Switch
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.27 MutualCoupling

This class represents the zero sequence line mutual coupling.

Table 214 shows all attributes of MutualCoupling.

Table 214 – Attributes of Wires::MutualCoupling

name	type	description
b0ch	Susceptance	Zero sequence mutual coupling shunt (charging) susceptance, uniformly distributed, of the entire line section.
distance11	Length	Distance to the start of the coupled region from the first line's terminal having sequence number equal to 1.
distance12	Length	Distance to the end of the coupled region from the first line's terminal with sequence number equal to 1.
distance21	Length	Distance to the start of coupled region from the second line's terminal with sequence number equal to 1.
distance22	Length	Distance to the end of coupled region from the second line's terminal with sequence number equal to 1.
g0ch	Conductance	Zero sequence mutual coupling shunt (charging) conductance, uniformly distributed, of the entire line section.
r0	Resistance	Zero sequence branch-to-branch mutual impedance coupling, resistance.
x0	Reactance	Zero sequence branch-to-branch mutual impedance coupling, reactance.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 215 shows all association ends of MutualCoupling with other classes.

Table 215 – Association ends of Wires::MutualCoupling with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Second_Terminal	Terminal	The starting terminal for the calculation of distances along the second branch of the mutual coupling.
[0..*]	[1..1] First_Terminal	Terminal	The starting terminal for the calculation of distances along the first branch of the mutual coupling. Normally MutualCoupling would only be used for terminals of AC line segments. The first and second terminals of a mutual coupling should point to different AC line segments.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.28 OperatingMode datatype

Textual name for an operating mode.

Table 216 shows all attributes of OperatingMode.

Table 216 – Attributes of Wires::OperatingMode

name	type	description
multiplier	UnitMultiplier	
unit	UnitSymbol	
value	String	

6.8.29 PerLengthImpedance

Common type for per-length impedance electrical catalogues.

Table 217 shows all attributes of PerLengthImpedance.

Table 217 – Attributes of Wires::PerLengthImpedance

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 218 shows all association ends of PerLengthImpedance with other classes.

Table 218 – Association ends of Wires::PerLengthImpedance with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] ACLineSegments	ACLineSegment	All line segments described by this per-length impedance.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.30 PerLengthPhaseImpedance

Impedance and admittance parameters per unit length for n-wire unbalanced lines, in matrix form.

Table 219 shows all attributes of PerLengthPhaseImpedance.

Table 219 – Attributes of Wires::PerLengthPhaseImpedance

name	type	description
conductorCount	Integer	Number of phase, neutral, and other wires retained. Constrains the number of matrix elements and the phase codes that can be used with this matrix.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 220 shows all association ends of PerLengthPhaseImpedance with other classes.

Table 220 – Association ends of Wires::PerLengthPhaseImpedance with other classes

[mult from]	[mult to] name	type	description
[1..1]	[1..*] PhaseImpedanceData	PhaseImpedanceData	All data that belong to this conductor phase impedance.
[0..1]	[0..*] ACLineSegments	ACLineSegment	inherited from: PerLengthImpedance
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.31 PerLengthSequenceImpedance

Sequence impedance and admittance parameters per unit length, for transposed lines of 1, 2, or 3 phases. For 1-phase lines, define $x=x_0=x_{self}$. For 2-phase lines, define $x=x_s-x_m$ and $x_0=x_s+x_m$.

Table 221 shows all attributes of PerLengthSequenceImpedance.

Table 221 – Attributes of Wires::PerLengthSequenceImpedance

name	type	description
b0ch	SusceptancePerLength	Zero sequence shunt (charging) susceptance, per unit of length.
bch	SusceptancePerLength	Positive sequence shunt (charging) susceptance, per unit of length.
g0ch	ConductancePerLength	Zero sequence shunt (charging) conductance, per unit of length.
gch	ConductancePerLength	Positive sequence shunt (charging) conductance, per unit of length.
r	ResistancePerLength	Positive sequence series resistance, per unit of length.
r0	ResistancePerLength	Zero sequence series resistance, per unit of length.
x	ReactancePerLength	Positive sequence series reactance, per unit of length.
x0	ReactancePerLength	Zero sequence series reactance, per unit of length.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 222 shows all association ends of PerLengthSequenceImpedance with other classes.

Table 222 – Association ends of Wires::PerLengthSequenceImpedance with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] ACLineSegments	ACLineSegment	inherited from: PerLengthImpedance
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.32 PhaseImpedanceData root class

Triplet of resistance, reactance, and susceptance matrix element values.

Table 223 shows all attributes of PhaseImpedanceData.

Table 223 – Attributes of Wires::PhaseImpedanceData

name	type	description
sequenceNumber	Integer	Column-wise element index, assuming a symmetrical matrix. Ranges from 1 to $N + N*(N-1)/2$.
b	SusceptancePerLength	Susceptance matrix element value, per length of unit.
r	ResistancePerLength	Resistance matrix element value, per length of unit.
x	ReactancePerLength	Reactance matrix element value, per length of unit.

Table 224 shows all association ends of PhaseImpedanceData with other classes.

Table 224 – Association ends of Wires::PhaseImpedanceData with other classes

[mult from]	[mult to] name	type	description
[1..*]	[1..1] PhaseImpedance	PerLengthPhaseImpedance	Conductor phase impedance to which this data belongs.

6.8.33 PhaseShuntConnectionKind enumeration

The configuration of phase connections for a single terminal device such as a load or capacitor.

Table 225 shows all literals of PhaseShuntConnectionKind.

Table 225 – Literals of Wires::PhaseShuntConnectionKind

literal	description
D	Delta connection.
Y	Wye connection.
Yn	Wye, with neutral brought out for grounding.
I	Independent winding, for single-phase connections.

6.8.34 PhaseTapChanger

A transformer phase shifting tap model that controls the phase angle difference across the power transformer and potentially the active power flow through the power transformer. This phase tap model may also impact the voltage magnitude.

Table 226 shows all attributes of PhaseTapChanger.

Table 226 – Attributes of Wires::PhaseTapChanger

name	type	description
highStep	Integer	inherited from: TapChanger
initialDelay	Seconds	inherited from: TapChanger
lowStep	Integer	inherited from: TapChanger
ltcFlag	Boolean	inherited from: TapChanger
neutralStep	Integer	inherited from: TapChanger
neutralU	Voltage	inherited from: TapChanger
normalStep	Integer	inherited from: TapChanger
regulationStatus	Boolean	inherited from: TapChanger
subsequentDelay	Seconds	inherited from: TapChanger
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 227 shows all association ends of PhaseTapChanger with other classes.

Table 227 – Association ends of Wires::PhaseTapChanger with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] TransformerEnd	TransformerEnd	Transformer end to which this phase tap changer belongs.
[0..*]	[0..1] PhaseTapChangerTabular	PhaseTapChangerTabular	The phase tap changer table for this phase tap changer.
[0..*]	[0..1] TapChangerControl	TapChangerControl	inherited from: TapChanger
[1..1]	[0..1] SvTapStep	SvTapStep	inherited from: TapChanger
[1..1]	[0..*] TapSchedules	TapSchedule	inherited from: TapChanger
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.35 PhaseTapChangerAsymetrical

Describes the tap model for an asymmetrical phase shifting tranformer in which the difference voltage vector adds to the primary side voltage. The angle between the primary side voltage and the difference voltage is named the winding connection angle. The phase shift depends on both the difference voltage magnitude and the winding connection angle.

Table 228 shows all attributes of PhaseTapChangerAsymetrical.

Table 228 – Attributes of Wires::PhaseTapChangerAsymetrical

name	type	description
windingConnectionAngle	AngleDegrees	The phase angle between the in-phase winding and the out-of - phase winding used for creating phase shift. The out-of-phase winding produces what is known as the difference voltage. Setting this angle to 90 degrees is not the same as a symmemtrical transformer.
voltageStepIncrement	PerCent	inherited from: PhaseTapChangerNonLinear
xMax	Reactance	inherited from: PhaseTapChangerNonLinear
xMedian	Reactance	inherited from: PhaseTapChangerNonLinear
highStep	Integer	inherited from: TapChanger
initialDelay	Seconds	inherited from: TapChanger
lowStep	Integer	inherited from: TapChanger
ltcFlag	Boolean	inherited from: TapChanger
neutralStep	Integer	inherited from: TapChanger
neutralU	Voltage	inherited from: TapChanger
normalStep	Integer	inherited from: TapChanger
regulationStatus	Boolean	inherited from: TapChanger
subsequentDelay	Seconds	inherited from: TapChanger
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 229 shows all association ends of PhaseTapChangerAsymetrical with other classes.

Table 229 – Association ends of Wires::PhaseTapChangerAsymetrical with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] TransformerEnd	TransformerEnd	inherited from: PhaseTapChanger
[0..*]	[0..1] PhaseTapChangerTabular	PhaseTapChangerTabular	inherited from: PhaseTapChanger
[0..*]	[0..1] TapChangerControl	TapChangerControl	inherited from: TapChanger
[1..1]	[0..1] SvTapStep	SvTapStep	inherited from: TapChanger
[1..1]	[0..*] TapSchedules	TapSchedule	inherited from: TapChanger
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.36 PhaseTapChangerLinear

Describes a tap changer with a linear relation between the tap step and the phase angle difference across the transformer. This is a mathematical model that is an approximation of a real phase tap changer.

Table 230 shows all attributes of PhaseTapChangerLinear.

Table 230 – Attributes of Wires::PhaseTapChangerLinear

name	type	description
stepPhaseShiftIncrement	AngleDegrees	Phase shift per step position. A positive value indicates a positive phase shift from the winding where the tap is located to the other winding (for a two-winding transformer). The actual phase shift increment might be more accurately computed from the symmetrical or asymmetrical models or a tap step table lookup if those are available.
highStep	Integer	inherited from: TapChanger
initialDelay	Seconds	inherited from: TapChanger
lowStep	Integer	inherited from: TapChanger
ltcFlag	Boolean	inherited from: TapChanger
neutralStep	Integer	inherited from: TapChanger
neutralU	Voltage	inherited from: TapChanger
normalStep	Integer	inherited from: TapChanger
regulationStatus	Boolean	inherited from: TapChanger
subsequentDelay	Seconds	inherited from: TapChanger
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 231 shows all association ends of PhaseTapChangerLinear with other classes.

Table 231 – Association ends of Wires::PhaseTapChangerLinear with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] TransformerEnd	TransformerEnd	inherited from: PhaseTapChanger
[0..*]	[0..1] PhaseTapChangerTabular	PhaseTapChangerTabular	inherited from: PhaseTapChanger
[0..*]	[0..1] TapChangerControl	TapChangerControl	inherited from: TapChanger
[1..1]	[0..1] SvTapStep	SvTapStep	inherited from: TapChanger
[1..1]	[0..*] TapSchedules	TapSchedule	inherited from: TapChanger
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.37 PhaseTapChangerNonLinear

The non-linear phase tap changer describes the non-linear behavior of a phase tap changer. This is a base class for the symmetrical and asymmetrical phase tap changer models. The details of these models can be found in IEC 61970-301.

Table 232 shows all attributes of PhaseTapChangerNonLinear.

Table 232 – Attributes of Wires::PhaseTapChangerNonLinear

name	type	description
voltageStepIncrement	PerCent	The voltage step increment on the out of phase winding specified in percent of nominal voltage of the transformer end.
xMax	Reactance	The reactance at the maximum tap step.
xMedian	Reactance	The reactance at the mid tap step.
highStep	Integer	inherited from: TapChanger
initialDelay	Seconds	inherited from: TapChanger
lowStep	Integer	inherited from: TapChanger
ltcFlag	Boolean	inherited from: TapChanger
neutralStep	Integer	inherited from: TapChanger
neutralU	Voltage	inherited from: TapChanger
normalStep	Integer	inherited from: TapChanger
regulationStatus	Boolean	inherited from: TapChanger
subsequentDelay	Seconds	inherited from: TapChanger
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 233 shows all association ends of PhaseTapChangerNonLinear with other classes.

Table 233 – Association ends of Wires::PhaseTapChangerNonLinear with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] TransformerEnd	TransformerEnd	inherited from: PhaseTapChanger
[0..*]	[0..1] PhaseTapChangerTabular	PhaseTapChangerTabular	inherited from: PhaseTapChanger
[0..*]	[0..1] TapChangerControl	TapChangerControl	inherited from: TapChanger
[1..1]	[0..1] SvTapStep	SvTapStep	inherited from: TapChanger
[1..1]	[0..*] TapSchedules	TapSchedule	inherited from: TapChanger
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.38 PhaseTapChangerSymetrical

Describes a symmetrical phase shifting transformer tap model in which the secondary side voltage magnitude is the same as at the primary side. The difference voltage magnitude is the base in an equal-sided triangle where the sides corresponds to the primary and secondary voltages. The phase angle difference corresponds to the top angle and can be expressed as twice the arctangent of half the total difference voltage.

Table 234 shows all attributes of PhaseTapChangerSymetrical.

Table 234 – Attributes of Wires::PhaseTapChangerSymetrical

name	type	description
voltageStepIncrement	PerCent	inherited from: PhaseTapChangerNonLinear
xMax	Reactance	inherited from: PhaseTapChangerNonLinear
xMedian	Reactance	inherited from: PhaseTapChangerNonLinear
highStep	Integer	inherited from: TapChanger
initialDelay	Seconds	inherited from: TapChanger
lowStep	Integer	inherited from: TapChanger
ltcFlag	Boolean	inherited from: TapChanger
neutralStep	Integer	inherited from: TapChanger
neutralU	Voltage	inherited from: TapChanger
normalStep	Integer	inherited from: TapChanger
regulationStatus	Boolean	inherited from: TapChanger
subsequentDelay	Seconds	inherited from: TapChanger
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 235 shows all association ends of PhaseTapChangerSymetrical with other classes.

Table 235 – Association ends of Wires::PhaseTapChangerSymetrical with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] TransformerEnd	TransformerEnd	inherited from: PhaseTapChanger
[0..*]	[0..1] PhaseTapChangerTabular	PhaseTapChangerTabular	inherited from: PhaseTapChanger
[0..*]	[0..1] TapChangerControl	TapChangerControl	inherited from: TapChanger
[1..1]	[0..1] SvTapStep	SvTapStep	inherited from: TapChanger
[1..1]	[0..*] TapSchedules	TapSchedule	inherited from: TapChanger
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.39 PhaseTapChangerTabular

Describes a tabular curve for how the the phase angle difference and impedance varies with the tap step.

Table 236 shows all attributes of PhaseTapChangerTabular.

Table 236 – Attributes of Wires::PhaseTapChangerTabular

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 237 shows all association ends of PhaseTapChangerTabular with other classes.

Table 237 – Association ends of Wires::PhaseTapChangerTabular with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] PhaseTapChanger	PhaseTapChanger	The phase tap changers to which this phase tap table applies.
[1..1]	[1..*] PhaseTapChangerTabularPoint	PhaseTapChangerTabularPoint	The points of this table.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.40 PhaseTapChangerTabularPoint root class

Describes each tap step in the phase tap changer tabular curve.

Table 238 shows all attributes of PhaseTapChangerTabularPoint.

Table 238 – Attributes of Wires::PhaseTapChangerTabularPoint

name	type	description
angle	AngleDegrees	The angle difference in degrees.
b	PerCent	The magnetizing branch susceptance deviation in percent of nominal value. The actual susceptance is calculated as follows: calculated magnetizing susceptance = $b(\text{nominal}) * (1 + b(\text{from this class})/100)$. The $b(\text{nominal})$ is defined as the static magnetizing susceptance on the associated power transformer end or ends. This model assumes the star impedance (pi model) form.
g	PerCent	The magnetizing branch conductance deviation in percent of nominal value. The actual conductance is calculated as follows: calculated magnetizing conductance = $b(\text{nominal}) * (1 + g(\text{from this class})/100)$. The $g(\text{nominal})$ is defined as the static magnetizing conductance on the associated power transformer end or ends. This model assumes the star impedance (pi model) form.
r	PerCent	The resistance deviation in percent of nominal value. The actual resistance is calculated as follows: calculated resistance = $r(\text{nominal}) * (1 + r(\text{from this class})/100)$. The $r(\text{nominal})$ is defined as the static resistance on the associated power transformer end or ends. This model assumes the star impedance (pi model) form.
step	Integer	The tap step.
x	PerCent	The series reactance deviation in percent of nominal value. The actual reactance is calculated as follows: calculated reactance = $x(\text{nominal}) * (1 + x(\text{from this class})/100)$. The $x(\text{nominal})$ is defined as the static series reactance on the associated power transformer end or ends. This model assumes the star impedance (pi model) form.

Table 239 shows all association ends of PhaseTapChangerTabularPoint with other classes.

Table 239 – Association ends of Wires::PhaseTapChangerTabularPoint with other classes

[mult from]	[mult to] name	type	description
[1..*]	[1..1] PhaseTapChangerTabular	PhaseTapChangerTabular	The table of this point.

6.8.41 Plant

A Plant is a collection of equipment for purposes of generation.

Table 240 shows all attributes of Plant.

Table 240 – Attributes of Wires::Plant

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 241 shows all association ends of Plant with other classes.

Table 241 – Association ends of Wires::Plant with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] Equipments	Equipment	inherited from: EquipmentContainer
[1..1]	[0..*] ConnectivityNodes	ConnectivityNode	inherited from: ConnectivityNodeContainer
[0..1]	[0..*] TopologicalNode	TopologicalNode	inherited from: ConnectivityNodeContainer
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.42 PowerTransformer

An electrical device consisting of two or more coupled windings, with or without a magnetic core, for introducing mutual coupling between electric circuits. Transformers can be used to control voltage and phase shift (active power flow).

A power transformer may be composed of separate transformer tanks that need not be identical.

A power transformer can be modelled with or without tanks and is intended for use in both balanced and unbalanced representations. A power transformer typically has two terminals, but may have one (grounding), three or more terminals.

Table 242 shows all attributes of PowerTransformer.

Table 242 – Attributes of Wires::PowerTransformer

name	type	description
vectorGroup	String	Vector group of the transformer for protective relaying, e.g., Dyn1. For unbalanced transformers, this may not be simply determined from the constituent winding connections and phase angle displacements. The vectorGroup string consists of the following components in the order listed: high voltage winding connection, mid voltage winding connection (for three winding transformers), phase displacement clock number from 0 to 11, low voltage winding connection phase displacement clock number from 0 to 11. The winding connections are D (delta), Y (wye), YN (wye with neutral), Z (zigzag), ZN (zigzag with neutral), A (auto transformer). Upper case means the high voltage, lower case mid or low. The high voltage winding always has clock position 0 and is not included in the vector group string. Some examples: YNy0 (two winding wye to wye with no phase displacement), YNd11 (two winding wye to delta with 330 degrees phase displacement), YNyn0d5 (three winding transformer wye with neutral high voltage, wye with neutral mid voltage and no phase displacement, delta low voltage with 150 degrees displacement). Phase displacement is defined as the angular difference between the phasors representing the voltages between the neutral point (real or imaginary) and the corresponding terminals of two windings, a positive sequence voltage system being applied to the high-voltage terminals, following each other in alphabetical sequence if they are lettered, or in numerical sequence if they are numbered: the phasors are assumed to rotate in a counter-clockwise sense.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 243 shows all association ends of PowerTransformer with other classes.

Table 243 – Association ends of Wires::PowerTransformer with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] PowerTransformerEnd	PowerTransformerEnd	The ends of this power transformer.
[0..1]	[0..*] TransformerTanks	TransformerTank	All transformers that belong to this bank.
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.43 PowerTransformerEnd

A PowerTransformerEnd is associated with each Terminal of a PowerTransformer.

The impedance values r , $r0$, x , and $x0$ of a PowerTransformerEnd represents a star equivalent as follows

- for a two Terminal PowerTransformer the high voltage PowerTransformerEnd has non zero values on r , $r0$, x , and $x0$ while the low voltage PowerTransformerEnd has zero values for r , $r0$, x , and $x0$.
- for a three Terminal PowerTransformer the three PowerTransformerEnds represents a star equivalent with each leg in the star represented by r , $r0$, x , and $x0$ values.
- for a PowerTransformer with more than three Terminals the PowerTransformerEnd impedance values cannot be used. Instead use the TransformerMeshImpedance or split the transformer into multiple PowerTransformers.

Table 244 shows all attributes of PowerTransformerEnd.

Table 244 – Attributes of Wires::PowerTransformerEnd

name	type	description
b	Susceptance	Magnetizing branch susceptance (B mag). The value can be positive or negative.
b0	Susceptance	Zero sequence magnetizing branch susceptance.
connectionKind	WindingConnection	Kind of connection.
g	Conductance	Magnetizing branch conductance.
g0	Conductance	Zero sequence magnetizing branch conductance (star-model).
phaseAngleClock	Integer	Terminal voltage phase angle displacement where 360 degrees are represented with clock hours. The valid values are 0 to 11. For example, for the secondary side end of a transformer with vector group code of 'Dyn11', specify the connection kind as wye with neutral and specify the phase angle of the clock as 11. The clock value of the transformer end number specified as 1, is assumed to be zero. Note the transformer end number is not assumed to be the same as the terminal sequence number.
r	Resistance	Resistance (star-model) of the transformer end.
r0	Resistance	Zero sequence series resistance (star-model) of the transformer end.
ratedS	ApparentPower	Normal apparent power rating.
ratedU	Voltage	Rated voltage: phase-phase for three-phase windings, and either phase-phase or phase-neutral for single-phase windings.
x	Reactance	Positive sequence series reactance (star-model) of the transformer end.
x0	Reactance	Zero sequence series reactance of the transformer end.
bmagSat	PerCent	inherited from: TransformerEnd
endNumber	Integer	inherited from: TransformerEnd
grounded	Boolean	inherited from: TransformerEnd
magBaseU	Voltage	inherited from: TransformerEnd
magSatFlux	PerCent	inherited from: TransformerEnd
rground	Resistance	inherited from: TransformerEnd
xground	Reactance	inherited from: TransformerEnd
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 245 shows all association ends of PowerTransformerEnd with other classes.

Table 245 – Association ends of Wires::PowerTransformerEnd with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] PowerTransformer	PowerTransformer	The power transformer of this power transformer end.
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: TransformerEnd
[0..*]	[0..1] Terminal	Terminal	inherited from: TransformerEnd
[1..1]	[0..1] PhaseTapChanger	PhaseTapChanger	inherited from: TransformerEnd
[1..1]	[0..1] RatioTapChanger	RatioTapChanger	inherited from: TransformerEnd
[0..*]	[0..1] StarImpedance	TransformerStarImpedance	inherited from: TransformerEnd
[0..*]	[0..1] CoreAdmittance	TransformerCoreAdmittance	inherited from: TransformerEnd
[1..1]	[0..*] FromMeshImpedance	TransformerMeshImpedance	inherited from: TransformerEnd
[1..*]	[0..*] ToMeshImpedance	TransformerMeshImpedance	inherited from: TransformerEnd
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.44 ProtectedSwitch

A ProtectedSwitch is a switching device that can be operated by ProtectionEquipment.

Table 246 shows all attributes of ProtectedSwitch.

Table 246 – Attributes of Wires::ProtectedSwitch

name	type	description
breakingCapacity	CurrentFlow	The maximum fault current a breaking device can break safely under prescribed conditions of use.
normalOpen	Boolean	inherited from: Switch
ratedCurrent	CurrentFlow	inherited from: Switch
switchOnCount	Integer	inherited from: Switch
switchOnDate	DateTime	inherited from: Switch
retained	Boolean	inherited from: Switch
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 247 shows all association ends of ProtectedSwitch with other classes.

Table 247 – Association ends of Wires::ProtectedSwitch with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] OperatedByProtectionEquipment	ProtectionEquipment	Protection equipments that operate this ProtectedSwitch.
[1..1]	[0..*] RecloseSequences	RecloseSequence	A breaker may have zero or more automatic reclosures after a trip occurs.
[1..1]	[0..*] SwitchSchedules	SwitchSchedule	inherited from: Switch
[1..1]	[0..*] SwitchPhase	SwitchPhase	inherited from: Switch
[0..*]	[0..1] CompositeSwitch	CompositeSwitch	inherited from: Switch
[0..*]	[0..*] SwitchingOperations	SwitchingOperation	inherited from: Switch
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.45 RatioTapChanger

A tap changer that changes the voltage ratio impacting the voltage magnitude but not the phase angle across the transformer.

Table 248 shows all attributes of RatioTapChanger.

Table 248 – Attributes of Wires::RatioTapChanger

name	type	description
stepVoltageIncrement	PerCent	Tap step increment, in per cent of nominal voltage, per step position.
tcuControlMode	TransformerControlMode	Specifies the regulation control mode (voltage or reactive) of the RatioTapChanger.
highStep	Integer	inherited from: TapChanger
initialDelay	Seconds	inherited from: TapChanger
lowStep	Integer	inherited from: TapChanger
ltcFlag	Boolean	inherited from: TapChanger
neutralStep	Integer	inherited from: TapChanger
neutralU	Voltage	inherited from: TapChanger
normalStep	Integer	inherited from: TapChanger
regulationStatus	Boolean	inherited from: TapChanger
subsequentDelay	Seconds	inherited from: TapChanger
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 249 shows all association ends of RatioTapChanger with other classes.

Table 249 – Association ends of Wires::RatioTapChanger with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] RatioTapChangerTabular	RatioTapChangerTabular	The tap ratio table for this ratio tap changer.
[0..1]	[1..1] TransformerEnd	TransformerEnd	Transformer end to which this ratio tap changer belongs.
[0..*]	[0..1] TapChangerControl	TapChangerControl	inherited from: TapChanger
[1..1]	[0..1] SvTapStep	SvTapStep	inherited from: TapChanger
[1..1]	[0..*] TapSchedules	TapSchedule	inherited from: TapChanger
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..1] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.46 RatioTapChangerTabular

Describes a curve for how the voltage magnitude and impedance vary with the tap step.

Table 250 shows all attributes of RatioTapChangerTabular.

Table 250 – Attributes of Wires::RatioTapChangerTabular

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 251 shows all association ends of RatioTapChangerTabular with other classes.

Table 251 – Association ends of Wires::RatioTapChangerTabular with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] RatioTapChanger	RatioTapChanger	The ratio tap changer of this tap ratio table.
[1..1]	[1..*] RatioTapChangerTabularPoint	RatioTapChangerTabularPoint	Points of this table.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.47 RatioTapChangerTabularPoint root class

Describes each tap step in the ratio tap changer tabular curve.

Table 252 shows all attributes of RatioTapChangerTabularPoint.

Table 252 – Attributes of Wires::RatioTapChangerTabularPoint

name	type	description
b	PerCent	The magnetizing branch susceptance deviation in percent of nominal value. The actual susceptance is calculated as follows: calculated magnetizing susceptance = $b(\text{nominal}) * (1 + b(\text{from this class})/100)$. The $b(\text{nominal})$ is defined as the static magnetizing susceptance on the associated power transformer end or ends. This model assumes the star impedance (pi model) form.
g	PerCent	The magnetizing branch conductance deviation in percent of nominal value. The actual conductance is calculated as follows: calculated magnetizing conductance = $g(\text{nominal}) * (1 + g(\text{from this class})/100)$. The $g(\text{nominal})$ is defined as the static magnetizing conductance on the associated power transformer end or ends. This model assumes the star impedance (pi model) form.
r	PerCent	The resistance deviation in percent of nominal value. The actual reactance is calculated as follows: calculated resistance = $r(\text{nominal}) * (1 + r(\text{from this class})/100)$. The $r(\text{nominal})$ is defined as the static resistance on the associated power transformer end or ends. This model assumes the star impedance (pi model) form.
ratio	Float	The voltage ratio in per unit. Hence this is a value close to one.
step	Integer	The tap step.
x	PerCent	The series reactance deviation in percent of nominal value. The actual reactance is calculated as follows: calculated reactance = $x(\text{nominal}) * (1 + x(\text{from this class})/100)$. The $x(\text{nominal})$ is defined as the static series reactance on the associated power transformer end or ends. This model assumes the star impedance (pi model) form.

Table 253 shows all association ends of RatioTapChangerTabularPoint with other classes.

Table 253 – Association ends of Wires::RatioTapChangerTabularPoint with other classes

[mult from]	[mult to] name	type	description
[1..*]	[1..1] RatioTapChangerTabular	RatioTapChangerTabular	Table of this point.

6.8.48 ReactiveCapabilityCurve

Reactive power rating envelope versus the synchronous machine's active power, in both the generating and motoring modes. For each active power value there is a corresponding high and low reactive power limit value. Typically there will be a separate curve for each coolant condition, such as hydrogen pressure. The Y1 axis values represent reactive minimum and the Y2 axis values represent reactive maximum.

Table 254 shows all attributes of ReactiveCapabilityCurve.

Table 254 – Attributes of Wires::ReactiveCapabilityCurve

name	type	description
coolantTemperature	Temperature	The machine's coolant temperature (e.g., ambient air or stator circulating water).
hydrogenPressure	Pressure	The hydrogen coolant pressure
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 255 shows all association ends of ReactiveCapabilityCurve with other classes.

Table 255 – Association ends of Wires::ReactiveCapabilityCurve with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..*] InitiallyUsedBySynchronousMachines	SynchronousMachine	Synchronous machines using this curve as default.
[0..*]	[1..*] SynchronousMachines	SynchronousMachine	Synchronous machines using this curve.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.49 Recloser

Pole-mounted fault interrupter with built-in phase and ground relays, current transformer (CT), and supplemental controls.

Table 256 shows all attributes of Recloser.

Table 256 – Attributes of Wires::Recloser

name	type	description
breakingCapacity	CurrentFlow	inherited from: ProtectedSwitch
normalOpen	Boolean	inherited from: Switch
ratedCurrent	CurrentFlow	inherited from: Switch
switchOnCount	Integer	inherited from: Switch
switchOnDate	DateTime	inherited from: Switch
retained	Boolean	inherited from: Switch
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 257 shows all association ends of Recloser with other classes.

Table 257 – Association ends of Wires::Recloser with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] OperatedByProtectionEquipment	ProtectionEquipment	inherited from: ProtectedSwitch
[1..1]	[0..*] RecloseSequences	RecloseSequence	inherited from: ProtectedSwitch
[1..1]	[0..*] SwitchSchedules	SwitchSchedule	inherited from: Switch
[1..1]	[0..*] SwitchPhase	SwitchPhase	inherited from: Switch
[0..*]	[0..1] CompositeSwitch	CompositeSwitch	inherited from: Switch
[0..*]	[0..*] SwitchingOperations	SwitchingOperation	inherited from: Switch
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource

[mult from]	[mult to] name	type	description
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.50 RectifierInverter

Bi-directional AC-DC conversion equipment that can be used to control DC current, DC voltage, DC power flow, or firing angle.

Table 258 shows all attributes of RectifierInverter.

Table 258 – Attributes of Wires::RectifierInverter

name	type	description
bridges	Integer	Number of bridges
commutatingReactance	Reactance	Commutating reactance at AC bus frequency.
commutatingResistance	Resistance	Commutating resistance.
compoundResistance	Resistance	Compounding resistance.
frequency	Frequency	Frequency on the AC side.
maxP	ActivePower	The maximum active power on the DC side at which the fconverter should operate.
maxU	Voltage	The maximum voltage on the DC side at which the converter should operate.
minCompoundVoltage	Voltage	Minimum compounded DC voltage.
minP	ActivePower	The minimum active power on the DC side at which the converter should operate.
minU	Voltage	The minimum voltage on the DC side at which the converter should operate.
operatingMode	OperatingMode	Operating mode for the converter.
ratedU	Voltage	Rectifier/inverter primary base voltage.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 259 shows all association ends of RectifierInverter with other classes.

Table 259 – Association ends of Wires::RectifierInverter with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.51 RegulatingCondEq

A type of conducting equipment that can regulate a quantity (i.e. voltage or flow) at a specific point in the network.

Table 260 shows all attributes of RegulatingCondEq.

Table 260 – Attributes of Wires::RegulatingCondEq

name	type	description
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 261 shows all association ends of RegulatingCondEq with other classes.

Table 261 – Association ends of Wires::RegulatingCondEq with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] RegulatingControl	RegulatingControl	The regulating control scheme in which this equipment participates.
[0..1]	[1..*] Controls	Control	The controller outputs used to actually govern a regulating device, e.g. the magnetization of a synchronous machine or capacitor bank breaker actuator.
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.52 RegulatingControl

Specifies a set of equipment that works together to control a power system quantity such as voltage or flow.

Table 262 shows all attributes of RegulatingControl.

Table 262 – Attributes of Wires::RegulatingControl

name	type	description
mode	RegulatingControlModeKind	The regulating control mode presently available. This specifications allows for determining the kind of regulation without need for obtaining the units from a schedule.
targetRange	Float	This is the case input target range. This performs the same function as the value2 attribute on the regulation schedule in the case that schedules are not used. The units of those appropriate for the mode.
targetValue	Float	The target value specified for case input. This value can be used for the target value without the use of schedules. The value has the units appropriate to the mode attribute.
discrete	Boolean	The regulation is performed in a discrete mode.
monitoredPhase	PhaseCode	Phase voltage controlling this regulator, measured at regulator location.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 263 shows all association ends of RegulatingControl with other classes.

Table 263 – Association ends of Wires::RegulatingControl with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] Terminal	Terminal	The terminal associated with this regulating control. The terminal is associated instead of a node, since the terminal could connect into either a topological node (bus in bus-branch model) or a connectivity node (detailed switch model). Sometimes it is useful to model regulation at a terminal of a bus bar object since the bus bar can be present in both a bus-branch model or a model with switch detail.
[0..1]	[0..*] RegulatingCondEq	RegulatingCondEq	The equipment that participates in this regulating control scheme.
[1..1]	[0..*] RegulationSchedule	RegulationSchedule	Schedule for this regulating control.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.53 RegulatingControlModeKind enumeration

The kind of regulation model. For example regulating voltage, reactive power, active power, etc.

Table 264 shows all literals of RegulatingControlModeKind.

Table 264 – Literals of Wires::RegulatingControlModeKind

literal	description
voltage	Voltage is specified.
activePower	Active power is specified.
reactivePower	Reactive power is specified.
currentFlow	Current flow is specified.
fixed	The regulation mode is fixed, and thus not regulating.
admittance	Admittance is specified.
timeScheduled	Control switches on/off by time of day. The times may change on the weekend, or in different seasons.
temperature	Control switches on/off based on the local temperature (i.e., a thermostat).
powerFactor	Power factor is specified.

6.8.54 RegulationSchedule

A pre-established pattern over time for a controlled variable, e.g., busbar voltage.

Table 265 shows all attributes of RegulationSchedule.

Table 265 – Attributes of Wires::RegulationSchedule

name	type	description
endTime	DateTime	inherited from: RegularIntervalSchedule
timeStep	Seconds	inherited from: RegularIntervalSchedule
startTime	DateTime	inherited from: BasicIntervalSchedule
value1Unit	UnitSymbol	inherited from: BasicIntervalSchedule
value1Multiplier	UnitMultiplier	inherited from: BasicIntervalSchedule
value2Unit	UnitSymbol	inherited from: BasicIntervalSchedule
value2Multiplier	UnitMultiplier	inherited from: BasicIntervalSchedule
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 266 shows all association ends of RegulationSchedule with other classes.

Table 266 – Association ends of Wires::RegulationSchedule with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] RegulatingControl	RegulatingControl	Regulating controls that have this Schedule.
[0..1]	[0..*] VoltageControlZones	VoltageControlZone	A VoltageControlZone may have a voltage regulation schedule.
[0..*]	[0..1] DayType	DayType	inherited from: SeasonDayTypeSchedule
[0..*]	[0..1] Season	Season	inherited from: SeasonDayTypeSchedule
[1..1]	[1..*] TimePoints	RegularTimePoint	inherited from: RegularIntervalSchedule
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.55 Resistor

Resistor, typically used in filter configurations or as earthing resistor for transformers. Used for electrical model of distribution networks.

Table 267 shows all attributes of Resistor.

Table 267 – Attributes of Wires::Resistor

name	type	description
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 268 shows all association ends of Resistor with other classes.

Table 268 – Association ends of Wires::Resistor with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.56 RotatingMachine

A rotating machine which may be used as a generator or motor.

Table 269 shows all attributes of RotatingMachine.

Table 269 – Attributes of Wires::RotatingMachine

name	type	description

name	type	description
damping	Float	Damping torque coefficient, a proportionality constant that, when multiplied by the angular velocity of the rotor poles with respect to the magnetic field (frequency), results in the damping torque. This value is often zero when the sources of damping torques (generator damper windings, load damping effects, etc.) are modeled in detail.
inertia	Seconds	Inertia constant of generator or motor and mechanical load. Must be greater than zero. This is the specification for the stored energy in the rotating mass when operating at rated speed. For a generator, this includes the generator plus all other elements (turbine, exciter) on the same shaft and has units of MW-sec. For a motor, it includes the motor plus its mechanical load. Conventional units are per unit on the generator MVA base, usually expressed as MW-second/MVA or just second. This value is used in the accelerating power reference frame for operator training simulator solutions.
ratedS	ApparentPower	Nameplate apparent power rating for the unit.
saturationFactor	Float	Saturation factor at rated terminal voltage. Should be greater than or equal to zero.
saturationFactor120	Float	Saturation factor at 120% of rated terminal voltage. Should be greater than or equal to the saturation factor at rated terminal voltage.
statorLeakageReactance	Reactance	Stator leakage reactance. Should be greater than or equal to 0.
statorResistance	Resistance	Stator (armature) resistance. Should be greater than or equal to zero. Equivalent resistance when used for GenEquiv model.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 270 shows all association ends of RotatingMachine with other classes.

Table 270 – Association ends of Wires::RotatingMachine with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] RegulatingControl	RegulatingControl	inherited from: RegulatingCondEq
[0..1]	[1..*] Controls	Control	inherited from: RegulatingCondEq
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.57 Sectionaliser

Automatic switch that will lock open to isolate a faulted section. It may, or may not, have load breaking capability. Its primary purpose is to provide fault sectionalising at locations where the fault current is either too high, or too low, for proper coordination of fuses.

Table 271 shows all attributes of Sectionaliser.

Table 271 – Attributes of Wires::Sectionaliser

name	type	description
normalOpen	Boolean	inherited from: Switch
ratedCurrent	CurrentFlow	inherited from: Switch
switchOnCount	Integer	inherited from: Switch
switchOnDate	DateTime	inherited from: Switch
retained	Boolean	inherited from: Switch
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 272 shows all association ends of Sectionaliser with other classes.

Table 272 – Association ends of Wires::Sectionaliser with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] SwitchSchedules	SwitchSchedule	inherited from: Switch
[1..1]	[0..*] SwitchPhase	SwitchPhase	inherited from: Switch
[0..*]	[0..1] CompositeSwitch	CompositeSwitch	inherited from: Switch
[0..*]	[0..*] SwitchingOperations	SwitchingOperation	inherited from: Switch
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.58 SeriesCompensator

A Series Compensator is a series capacitor or reactor or an AC transmission line without charging susceptance. It is a two terminal device.

Table 273 shows all attributes of SeriesCompensator.

Table 273 – Attributes of Wires::SeriesCompensator

name	type	description
r	Resistance	Positive sequence resistance.
r0	Resistance	Zero sequence resistance.
x	Reactance	Positive sequence reactance.
x0	Reactance	Zero sequence reactance.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 274 shows all association ends of SeriesCompensator with other classes.

Table 274 – Association ends of Wires::SeriesCompensator with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.59 ShuntCompensator

A shunt capacitor or reactor or switchable bank of shunt capacitors or reactors. A section of a shunt compensator is an individual capacitor or reactor. A negative value for reactivePerSection indicates that the compensator is a reactor. ShuntCompensator is a single terminal device. Ground is implied.

Table 275 shows all attributes of ShuntCompensator.

Table 275 – Attributes of Wires::ShuntCompensator

name	type	description
aVRDelay	Seconds	Time delay required for the device to be connected or disconnected by automatic voltage regulation (AVR).
b0PerSection	Susceptance	Zero sequence shunt (charging) susceptance per section
bPerSection	Susceptance	Positive sequence shunt (charging) susceptance per section
g0PerSection	Conductance	Zero sequence shunt (charging) conductance per section
gPerSection	Conductance	Positive sequence shunt (charging) conductance per section
grounded	WindingConnection	Used for Yn and Zn connections. True if the neutral is solidly grounded.
maximumSections	Integer	The maximum number of sections that may be switched in.
nomU	Voltage	The voltage at which the nominal reactive power may be calculated. This should normally be within 10% of the voltage at which the capacitor is connected to the network.
normalSections	Integer	The normal number of sections switched in.
phaseConnection	PhaseShuntConnectionKind	The type of phase connection, such as wye or delta.
switchOnCount	Integer	The switch on count since the capacitor count was last reset or initialized.
switchOnDate	DateTime	The date and time when the capacitor bank was last switched on.
voltageSensitivity	VoltagePerReactivePower	Voltage sensitivity required for the device to regulate the bus voltage, in voltage/reactive power.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 276 shows all association ends of ShuntCompensator with other classes.

Table 276 – Association ends of Wires::ShuntCompensator with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] ShuntCompensatorPhase	ShuntCompensatorPhase	The individual phases models for the shunt compensator.
[1..1]	[0..1] SvShuntCompensatorSections	SvShuntCompensatorSections	The state for the number of shunt compensator sections in service.
[0..*]	[0..1] RegulatingControl	RegulatingControl	inherited from: RegulatingCondEq
[0..1]	[1..*] Controls	Control	inherited from: RegulatingCondEq
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.60 ShuntCompensatorPhase

Single phase of a multi-phase shunt compensator when its attributes might be different per phase.

Table 277 shows all attributes of ShuntCompensatorPhase.

Table 277 – Attributes of Wires::ShuntCompensatorPhase

name	type	description
bPerSection	Susceptance	Susceptance per section of the phase if shunt compensator is wye connected. Susceptance per section phase to phase if shunt compensator is delta connected.
gPerSection	Conductance	Conductance per section for this phase if shunt compensator is wye connected. Conductance per section phase to phase if shunt compensator is delta connected.
maximumSections	Integer	The maximum number of sections that may be switched in for this phase.
normalSections	Integer	For the capacitor phase, the normal number of sections switched in.
phase	SinglePhaseKind	Phase of this shunt compensator component. If the shunt compensator is wye connected, the connection is from the indicated phase to the central ground or neutral point. If the shunt compensator is delta connected, the phase indicates a shunt compensator connected from the indicated phase to the next logical non-neutral phase.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject

name	type	description
name	String	inherited from: IdentifiedObject

Table 278 shows all association ends of ShuntCompensatorPhase with other classes.

Table 278 – Association ends of Wires::ShuntCompensatorPhase with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] ShuntCompensator	ShuntCompensator	Shunt compensator of this shunt compensator phase.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.61 SinglePhaseKind enumeration

Enumeration of single phase identifiers. Allows designation of single phases for both transmission and distribution equipment, circuits and loads.

Table 279 shows all literals of SinglePhaseKind.

Table 279 – Literals of Wires::SinglePhaseKind

literal	description
A	Phase A.
B	Phase B.
C	Phase C.
N	Neutral.
s1	Secondary phase 1.
s2	Secondary phase 2.

6.8.62 StaticVarCompensator

A facility for providing variable and controllable shunt reactive power. The SVC typically consists of a stepdown transformer, filter, thyristor-controlled reactor, and thyristor-switched capacitor arms.

The SVC may operate in fixed MVar output mode or in voltage control mode. When in voltage control mode, the output of the SVC will be proportional to the deviation of voltage at the controlled bus from the voltage setpoint. The SVC characteristic slope defines the proportion. If the voltage at the controlled bus is equal to the voltage setpoint, the SVC MVar output is zero.

Table 280 shows all attributes of StaticVarCompensator.

Table 280 – Attributes of Wires::StaticVarCompensator

name	type	description
capacitiveRating	Reactance	Maximum available capacitive reactance.
inductiveRating	Reactance	Maximum available inductive reactance.
slope	VoltagePerReactivePower	The characteristics slope of an SVC defines how the reactive power output changes in proportion to the difference between the regulated bus voltage and the voltage setpoint.
sVCControlMode	SVCControlMode	SVC control mode.
voltageSetPoint	Voltage	The reactive power output of the SVC is proportional to the difference between the voltage at the regulated bus and the voltage setpoint. When the regulated bus voltage is equal to the voltage setpoint, the reactive power output is zero.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 281 shows all association ends of StaticVarCompensator with other classes.

Table 281 – Association ends of Wires::StaticVarCompensator with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] RegulatingControl	RegulatingControl	inherited from: RegulatingCondEq
[0..1]	[1..*] Controls	Control	inherited from: RegulatingCondEq
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.63 SVCControlMode enumeration

Static VAR Compensator control mode.

Table 282 shows all literals of SVCControlMode.

Table 282 – Literals of Wires::SVCControlMode

literal	description
reactivePower	
voltage	
off	

6.8.64 Switch

A generic device designed to close, or open, or both, one or more electric circuits.

Table 283 shows all attributes of Switch.

Table 283 – Attributes of Wires::Switch

name	type	description
normalOpen	Boolean	The attribute is used in cases when no Measurement for the status value is present. If the Switch has a status measurment the Discrete.normalValue is expected to match with the Switch.normalOpen.
ratedCurrent	CurrentFlow	The maximum continuous current carrying capacity in amps governed by the device material and construction.
switchOnCount	Integer	The switch on count since the switch was last reset or initialized.
switchOnDate	DateTime	The date and time when the switch was last switched on.
retained	Boolean	Branch is retained in a bus branch model.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 284 shows all association ends of Switch with other classes.

Table 284 – Association ends of Wires::Switch with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] SwitchSchedules	SwitchSchedule	A Switch can be associated with SwitchSchedules.
[1..1]	[0..*] SwitchPhase	SwitchPhase	The individual switch phases for the switch.
[0..*]	[0..1] CompositeSwitch	CompositeSwitch	Composite switch to which this Switch belongs.
[0..*]	[0..*] SwitchingOperations	SwitchingOperation	A switch may be operated by many schedules.
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRTYPE	PSRTYPE	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.65 SwitchPhase

Single phase of a multi-phase switch when its attributes might be different per phase.

Table 285 shows all attributes of SwitchPhase.

Table 285 – Attributes of Wires::SwitchPhase

name	type	description
normalOpen	Boolean	Used in cases when no Measurement for the status value is present. If the SwitchPhase has a status measurement the Discrete.normalValue is expected to match with this value.
phaseSide1	SinglePhaseKind	Phase of this SwitchPhase on the side with terminal sequence number equal 1. Should be a phase contained in that terminal's phases attribute.
phaseSide2	SinglePhaseKind	Phase of this SwitchPhase on the side with terminal sequence number equal 2. Should be a phase contained in that terminal's Terminal.phases attribute.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 286 shows all association ends of SwitchPhase with other classes.

Table 286 – Association ends of Wires::SwitchPhase with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Switch	Switch	The switch of the switch phase.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.66 SwitchSchedule

A schedule of switch positions. If RegularTimePoint.value1 is 0, the switch is open. If 1, the switch is closed.

Table 287 shows all attributes of SwitchSchedule.

Table 287 – Attributes of Wires::SwitchSchedule

name	type	description
endTime	DateTime	inherited from: RegularIntervalSchedule
timeStep	Seconds	inherited from: RegularIntervalSchedule
startTime	DateTime	inherited from: BasicIntervalSchedule
value1Unit	UnitSymbol	inherited from: BasicIntervalSchedule
value1Multiplier	UnitMultiplier	inherited from: BasicIntervalSchedule
value2Unit	UnitSymbol	inherited from: BasicIntervalSchedule
value2Multiplier	UnitMultiplier	inherited from: BasicIntervalSchedule
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 288 shows all association ends of SwitchSchedule with other classes.

Table 288 – Association ends of Wires::SwitchSchedule with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Switch	Switch	A SwitchSchedule is associated with a Switch.
[0..*]	[0..1] DayType	DayType	inherited from: SeasonDayTypeSchedule
[0..*]	[0..1] Season	Season	inherited from: SeasonDayTypeSchedule
[1..1]	[1..*] TimePoints	RegularTimePoint	inherited from: RegularIntervalSchedule
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.67 SynchronousGeneratorType enumeration

Type of synchronous generator as used in dynamic simulation applications.

Table 289 shows all literals of SynchronousGeneratorType.

Table 289 – Literals of Wires::SynchronousGeneratorType

literal	description
roundRotor	Also known as genrou.
salientPole	Also known as gensal.
transient	Also known as gentra.
typeF	Also known as typeF.
typeJ	Also known as typeJ.

6.8.68 SynchronousMachine

An electromechanical device that operates with shaft rotating synchronously with the network. It is a single machine operating either as a generator or synchronous condenser or pump.

Table 290 shows all attributes of SynchronousMachine.

Table 290 – Attributes of Wires::SynchronousMachine

name	type	description
aVRToManualLag	Seconds	Time delay required when switching from Automatic Voltage Regulation (AVR) to Manual for a lagging MVar violation.
aVRToManualLead	Seconds	Time delay required when switching from Automatic Voltage Regulation (AVR) to Manual for a leading MVar violation.
baseQ	ReactivePower	Default base reactive power value. This value represents the initial reactive power that can be used by any application function.
condenserP	ActivePower	Active power consumed when in condenser mode operation.
coolantCondition	Float	Temperature or pressure of coolant medium
coolantType	CoolantType	Method of cooling the machine.
manualToAVR	Seconds	Time delay required when switching from Manual to Automatic Voltage Regulation. This value is used in the accelerating power reference frame for powerflow solutions.
maxQ	ReactivePower	Maximum reactive power limit. This is the maximum (nameplate) limit for the unit.
maxU	Voltage	Maximum voltage limit for the unit.
minQ	ReactivePower	Minimum reactive power limit for the unit.
minU	Voltage	Minimum voltage limit for the unit.
operatingMode	SynchronousMachineOperatingMode	Current mode of operation.
qPercent	PerCent	Percent of the coordinated reactive control that comes from this machine.
r	Resistance	Positive sequence resistance of the synchronous machine.
r0	Resistance	Zero sequence resistance of the synchronous machine.
r2	Resistance	Negative sequence resistance.
referencePriority	Integer	Priority of unit for use as powerflow voltage phase angle reference bus selection. 0 = don't care (default) 1 = highest priority. 2 is less than 1 and so on.
synchronousGeneratorType	SynchronousGeneratorType	Type of Synchronous Generator used in Dynamic simulation applications.
tpdo	Seconds	Direct axis transient rotor time constant, also known as T'do
tppdo	Seconds	Direct axis sub-transient rotor time constant, also known as T''do
tppqo	Seconds	Quadrature axis subtransient rotor time constant, also known as T''qo
tpqo	Seconds	Quadrature axis transient rotor time constant, also known as T'qo
type	SynchronousMachineType	Modes that this synchronous machine can operate in.
x	Reactance	Positive sequence reactance of the synchronous machine.
x0	Reactance	Zero sequence reactance of the synchronous machine.
x2	Reactance	Negative sequence reactance.
xDirectSubtrans	Reactance	Direct-axis subtransient reactance, also known as X''d.

name	type	description
xDirectSync	Reactance	Direct-axis synchronous reactance. The quotient of a sustained value of that AC component of armature voltage that is produced by the total direct-axis flux due to direct-axis armature current and the value of the AC component of this current, the machine running at rated speed. (X_d)
xDirectTrans	Reactance	Direct-axis transient reactance, also known as X'_d .
xQuadSubtrans	Reactance	Quadrature-axis subtransient reactance, also known as X''_q .
xQuadSync	Reactance	Quadrature-axis synchronous reactance (X_q) , the ratio of the component of reactive armature voltage, due to the quadrature-axis component of armature current, to this component of current, under steady state conditions and at rated frequency.
xQuadTrans	Reactance	Quadrature-axis transient reactance, also known as X'_q .
damping	Float	inherited from: RotatingMachine
inertia	Seconds	inherited from: RotatingMachine
ratedS	ApparentPower	inherited from: RotatingMachine
saturationFactor	Float	inherited from: RotatingMachine
saturationFactor120	Float	inherited from: RotatingMachine
statorLeakageReactance	Reactance	inherited from: RotatingMachine
statorResistance	Resistance	inherited from: RotatingMachine
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 291 shows all association ends of SynchronousMachine with other classes.

Table 291 – Association ends of Wires::SynchronousMachine with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..1] HydroPump	HydroPump	The synchronous machine drives the turbine which moves the water from a low elevation to a higher elevation. The direction of machine rotation for pumping may or may not be the same as for generating.
[1..*]	[0..1] InitialReactiveCapabilityCurve	ReactiveCapabilityCurve	The default reactive capability curve for use by a synchronous machine.
[1..*]	[0..*] ReactiveCapabilityCurves	ReactiveCapabilityCurve	All available reactive capability curves for this synchronous machine.
[1..*]	[0..1] GeneratingUnit	GeneratingUnit	A synchronous machine may operate as a generator and as such becomes a member of a generating unit.
[0..*]	[0..*] PrimeMovers	PrimeMover	Prime movers that drive this SynchronousMachine.
[0..*]	[0..1] RegulatingControl	RegulatingControl	inherited from: RegulatingCondEq
[0..1]	[1..*] Controls	Control	inherited from: RegulatingCondEq
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.69 SynchronousMachineOperatingMode enumeration

Synchronous machine operating mode.

Table 292 shows all literals of SynchronousMachineOperatingMode.

Table 292 – Literals of Wires::SynchronousMachineOperatingMode

literal	description
generator	
condenser	

6.8.70 SynchronousMachineType enumeration

Synchronous machine type.

Table 293 shows all literals of SynchronousMachineType.

Table 293 – Literals of Wires::SynchronousMachineType

literal	description
generator	
condenser	
generator_or_condenser	

6.8.71 TapChanger

Mechanism for changing transformer winding tap positions.

Table 294 shows all attributes of TapChanger.

Table 294 – Attributes of Wires::TapChanger

name	type	description
highStep	Integer	Highest possible tap step position, advance from neutral
initialDelay	Seconds	For an LTC, the delay for initial tap changer operation (first step change)
lowStep	Integer	Lowest possible tap step position, retard from neutral
ltcFlag	Boolean	Specifies whether or not a TapChanger has load tap changing capabilities.
neutralStep	Integer	The neutral tap step position for this winding.
neutralU	Voltage	Voltage at which the winding operates at the neutral tap setting.
normalStep	Integer	The tap step position used in "normal" network operation for this winding. For a "Fixed" tap changer indicates the current physical tap setting.
regulationStatus	Boolean	Specifies the default regulation status of the TapChanger. True is regulating. False is not regulating.
subsequentDelay	Seconds	For an LTC, the delay for subsequent tap changer operation (second and later step changes)
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 295 shows all association ends of TapChanger with other classes.

Table 295 – Association ends of Wires::TapChanger with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] TapChangerControl	TapChangerControl	The regulating control scheme in which this tap changer participates.
[1..1]	[0..1] SvTapStep	SvTapStep	The tap step state associated with the tap changer.
[1..1]	[0..*] TapSchedules	TapSchedule	A TapChanger can have TapSchedules.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.72 TapChangerControl

Describes behaviour specific to tap changers, e.g. how the voltage at the end of a line varies with the load level and compensation of the voltage drop by tap adjustment.

Table 296 shows all attributes of TapChangerControl.

Table 296 – Attributes of Wires::TapChangerControl

name	type	description
limitVoltage	Voltage	Maximum allowed regulated voltage on the PT secondary, regardless of line drop compensation. Sometimes referred to as first-house protection.
lineDropCompensation	Boolean	If true, the line drop compensation is to be applied.
lineDropR	Resistance	Line drop compensator resistance setting for normal (forward) power flow.
lineDropX	Reactance	Line drop compensator reactance setting for normal (forward) power flow.
reverseLineDropR	Resistance	Line drop compensator resistance setting for reverse power flow.
reverseLineDropX	Reactance	Line drop compensator reactance setting for reverse power flow.
mode	RegulatingControlModeKind	inherited from: RegulatingControl
targetRange	Float	inherited from: RegulatingControl
targetValue	Float	inherited from: RegulatingControl
discrete	Boolean	inherited from: RegulatingControl
monitoredPhase	PhaseCode	inherited from: RegulatingControl
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 297 shows all association ends of TapChangerControl with other classes.

Table 297 – Association ends of Wires::TapChangerControl with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] TapChanger	TapChanger	The tap changer that participates in this regulating tap control scheme.
[0..*]	[0..1] Terminal	Terminal	inherited from: RegulatingControl
[0..1]	[0..*] RegulatingCondEq	RegulatingCondEq	inherited from: RegulatingControl
[1..1]	[0..*] RegulationSchedule	RegulationSchedule	inherited from: RegulatingControl
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.73 TapChangerKind enumeration

Transformer tap changer type. Indicates the capabilities of the tap changer independent of the operating mode.

Table 298 shows all literals of TapChangerKind.

Table 298 – Literals of Wires::TapChangerKind

literal	description
fixed	Not capable of control. This is also indicated by no association of TapChanger to a RegulatingControl.
voltageControl	Capable of voltage control.
phaseControl	Capable of phase control.
voltageAndPhaseControl	Capable of voltage and phase control.

6.8.74 TapSchedule

A pre-established pattern over time for a tap step.

Table 299 shows all attributes of TapSchedule.

Table 299 – Attributes of Wires::TapSchedule

name	type	description
endTime	DateTime	inherited from: RegularIntervalSchedule
timeStep	Seconds	inherited from: RegularIntervalSchedule
startTime	DateTime	inherited from: BasicIntervalSchedule
value1Unit	UnitSymbol	inherited from: BasicIntervalSchedule
value1Multiplier	UnitMultiplier	inherited from: BasicIntervalSchedule
value2Unit	UnitSymbol	inherited from: BasicIntervalSchedule
value2Multiplier	UnitMultiplier	inherited from: BasicIntervalSchedule
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 300 shows all association ends of TapSchedule with other classes.

Table 300 – Association ends of Wires::TapSchedule with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] TapChanger	TapChanger	A TapSchedule is associated with a TapChanger.
[0..*]	[0..1] DayType	DayType	inherited from: SeasonDayTypeSchedule
[0..*]	[0..1] Season	Season	inherited from: SeasonDayTypeSchedule
[1..1]	[1..*] TimePoints	RegularTimePoint	inherited from: RegularIntervalSchedule
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.75 TransformerControlMode enumeration

Control modes for a transformer.

Table 301 shows all literals of TransformerControlMode.

Table 301 – Literals of Wires::TransformerControlMode

literal	description
volt	Voltage control
reactive	Reactive power flow control

6.8.76 TransformerCoreAdmittance

The transformer core admittance. Used to specify the core admittance of a transformer in a manner that can be shared among power transformers.

Table 302 shows all attributes of TransformerCoreAdmittance.

Table 302 – Attributes of Wires::TransformerCoreAdmittance

name	type	description
b	Susceptance	Magnetizing branch susceptance (B mag). The value can be positive or negative.
b0	Susceptance	Zero sequence magnetizing branch susceptance.
g	Conductance	Magnetizing branch conductance (G mag).
g0	Conductance	Zero sequence magnetizing branch conductance.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 303 shows all association ends of TransformerCoreAdmittance with other classes.

Table 303 – Association ends of Wires::TransformerCoreAdmittance with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] TransformerEnd	TransformerEnd	All transformer ends having this core admittance.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.77 TransformerEnd

A conducting connection point of a power transformer. It corresponds to a physical transformer winding terminal. In earlier CIM versions, the TransformerWinding class served a similar purpose, but this class is more flexible because it associates to terminal but is not a specialization of ConductingEquipment.

Table 304 shows all attributes of TransformerEnd.

Table 304 – Attributes of Wires::TransformerEnd

name	type	description
bmagSat	PerCent	Core shunt magnetizing susceptance in the saturation region.
endNumber	Integer	Number for this transformer end, corresponding to the end's order in the power transformer vector group or phase angle clock number. Highest voltage winding should be 1. Each end within a power transformer should have a unique subsequent end number. Note the transformer end number need not match the terminal sequence number.
grounded	Boolean	(for Yn and Zn connections) True if the neutral is solidly grounded.
magBaseU	Voltage	The reference voltage at which the magnetizing saturation measurements were made
magSatFlux	PerCent	Core magnetizing saturation curve knee flux level.
rground	Resistance	(for Yn and Zn connections) Resistance part of neutral impedance where 'grounded' is true.
xground	Reactance	(for Yn and Zn connections) Reactive part of neutral impedance where 'grounded' is true.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 305 shows all association ends of TransformerEnd with other classes.

Table 305 – Association ends of Wires::TransformerEnd with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] BaseVoltage	BaseVoltage	Base voltage of the transformer end. This is essential for PU calculation.
[0..*]	[0..1] Terminal	Terminal	Terminal of the power transformer to which this transformer end belongs.
[1..1]	[0..1] PhaseTapChanger	PhaseTapChanger	Phase tap changer associated with this transformer end.
[1..1]	[0..1] RatioTapChanger	RatioTapChanger	Ratio tap changer associated with this transformer end.
[0..*]	[0..1] StarImpedance	TransformerStarImpedance	(accurate for 2- or 3-winding transformers only) Pi-model impedances of this transformer end. By convention, for a two winding transformer, the full values of the transformer should be entered on the high voltage end (endNumber=1).
[0..*]	[0..1] CoreAdmittance	TransformerCoreAdmittance	Core admittance of this transformer end, representing magnetising current and core losses. The full values of the transformer should be supplied for one transformer end only.
[1..1]	[0..*] FromMeshImpedance	TransformerMeshImpedance	All mesh impedances between this 'to' and other 'from' transformer ends.
[1..*]	[0..*] ToMeshImpedance	TransformerMeshImpedance	All mesh impedances between this 'from' and other 'to' transformer ends.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.78 TransformerMeshImpedance

Transformer mesh impedance (Delta-model) between transformer ends.

The typical case is that this class describes the impedance between two transformer ends pair-wise, i.e. the cardinalities at both transformer end associations are 1. But in cases where two or more transformer ends are modeled the cardinalities are larger than 1.

Table 306 shows all attributes of TransformerMeshImpedance.

Table 306 – Attributes of Wires::TransformerMeshImpedance

name	type	description
r	Resistance	Resistance between the 'from' and the 'to' end, seen from the 'from' end.
r0	Resistance	Zero-sequence resistance between the 'from' and the 'to' end, seen from the 'from' end.
x	Reactance	Reactance between the 'from' and the 'to' end, seen from the 'from' end.
x0	Reactance	Zero-sequence reactance between the 'from' and the 'to' end, seen from the 'from' end.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 307 shows all association ends of TransformerMeshImpedance with other classes.

Table 307 – Association ends of Wires::TransformerMeshImpedance with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] FromTransformerEnd	TransformerEnd	From end this mesh impedance is connected to. It determines the voltage reference.
[0..*]	[1..*] ToTransformerEnd	TransformerEnd	All transformer ends this mesh impedance is connected to.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.79 TransformerStarImpedance

Transformer star impedance (Pi-model) that accurately reflects impedance for transformers with 2 or 3 windings. For transformers with 4 or more windings, you must use TransformerMeshImpedance class.

For transmission networks use PowerTransformerEnd impedances (r, r0, x, x0, b, b0, g and g0).

Table 308 shows all attributes of TransformerStarImpedance.

Table 308 – Attributes of Wires::TransformerStarImpedance

name	type	description
r	Resistance	Resistance of the transformer end.
r0	Resistance	Zero sequence series resistance of the transformer end.
x	Reactance	Positive sequence series reactance of the transformer end.
x0	Reactance	Zero sequence series reactance of the transformer end.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 309 shows all association ends of TransformerStarImpedance with other classes.

Table 309 – Association ends of Wires::TransformerStarImpedance with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] TransformerEnd	TransformerEnd	All transformer ends having this star impedance.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.80 TransformerTank

An assembly of two or more coupled windings that transform electrical power between voltage levels. These windings are bound on a common core and place in the same tank. Transformer tank can be used to model both single-phase and 3-phase transformers.

Table 310 shows all attributes of TransformerTank.

Table 310 – Attributes of Wires::TransformerTank

name	type	description
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 311 shows all association ends of TransformerTank with other classes.

Table 311 – Association ends of Wires::TransformerTank with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] PowerTransformer	PowerTransformer	Bank this transformer belongs to.
[0..1]	[1..*] TransformerTankEnds	TransformerTankEnd	All windings of this transformer.
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.81 TransformerTankEnd

Transformer tank end represents an individual winding for unbalanced models or for transformer tanks connected into a bank (and bank is modelled with the PowerTransformer).

Table 312 shows all attributes of TransformerTankEnd.

Table 312 – Attributes of Wires::TransformerTankEnd

name	type	description
phases	PhaseCode	Describes the phases carried by a conducting equipment.
bmagSat	PerCent	inherited from: TransformerEnd
endNumber	Integer	inherited from: TransformerEnd
grounded	Boolean	inherited from: TransformerEnd
magBaseU	Voltage	inherited from: TransformerEnd
magSatFlux	PerCent	inherited from: TransformerEnd
rground	Resistance	inherited from: TransformerEnd
xground	Reactance	inherited from: TransformerEnd
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 313 shows all association ends of TransformerTankEnd with other classes.

Table 313 – Association ends of Wires::TransformerTankEnd with other classes

[mult from]	[mult to] name	type	description
[1..*]	[0..1] TransformerTank	TransformerTank	Transformer this winding belongs to.
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: TransformerEnd
[0..*]	[0..1] Terminal	Terminal	inherited from: TransformerEnd
[1..1]	[0..1] PhaseTapChanger	PhaseTapChanger	inherited from: TransformerEnd
[1..1]	[0..1] RatioTapChanger	RatioTapChanger	inherited from: TransformerEnd
[0..*]	[0..1] StarImpedance	TransformerStarImpedance	inherited from: TransformerEnd
[0..*]	[0..1] CoreAdmittance	TransformerCoreAdmittance	inherited from: TransformerEnd
[1..1]	[0..*] FromMeshImpedance	TransformerMeshImpedance	inherited from: TransformerEnd
[1..*]	[0..*] ToMeshImpedance	TransformerMeshImpedance	inherited from: TransformerEnd
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.82 VoltageControlZone

An area of the power system network which is defined for secondary voltage control purposes. A voltage control zone consists of a collection of substations with a designated bus bar section whose voltage will be controlled.

Table 314 shows all attributes of VoltageControlZone.

Table 314 – Attributes of Wires::VoltageControlZone

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 315 shows all association ends of VoltageControlZone with other classes.

Table 315 – Association ends of Wires::VoltageControlZone with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] BusbarSection	BusbarSection	A VoltageControlZone is controlled by a designated BusbarSection.
[0..*]	[0..1] RegulationSchedule	RegulationSchedule	A VoltageControlZone may have a voltage regulation schedule.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.83 WindingConnection enumeration

Winding connection type.

Table 316 shows all literals of WindingConnection.

Table 316 – Literals of Wires::WindingConnection

literal	description
D	Delta
Y	Wye
Z	ZigZag
Yn	Wye, with neutral brought out for grounding.
Zn	ZigZag, with neutral brought out for grounding.
A	Autotransformer common winding
I	Independent winding, for single-phase connections

6.9 Package Generation

6.9.1 General

This package contains packages that have information for Unit Commitment and Economic Dispatch of Hydro and Thermal Generating Units, Load Forecasting, Automatic Generation Control, and Unit Modeling for Dynamic Training Simulator.

Figure 60 shows class diagram Main.

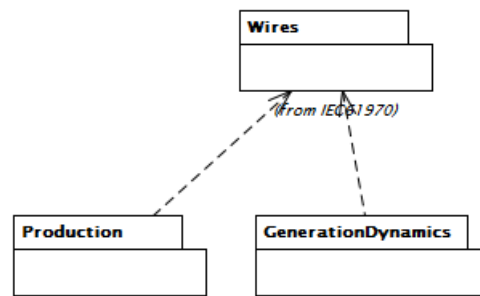


Figure 60 – Class diagram Generation::Main

This diagram shows the main generation packages and their dependency relationships to other packages.

6.9.2 Package GenerationDynamics

6.9.2.1 General

The Generation Dynamics package contains prime movers, such as turbines and boilers, which are needed for simulation and educational purposes.

Figure 61 shows class diagram Main.

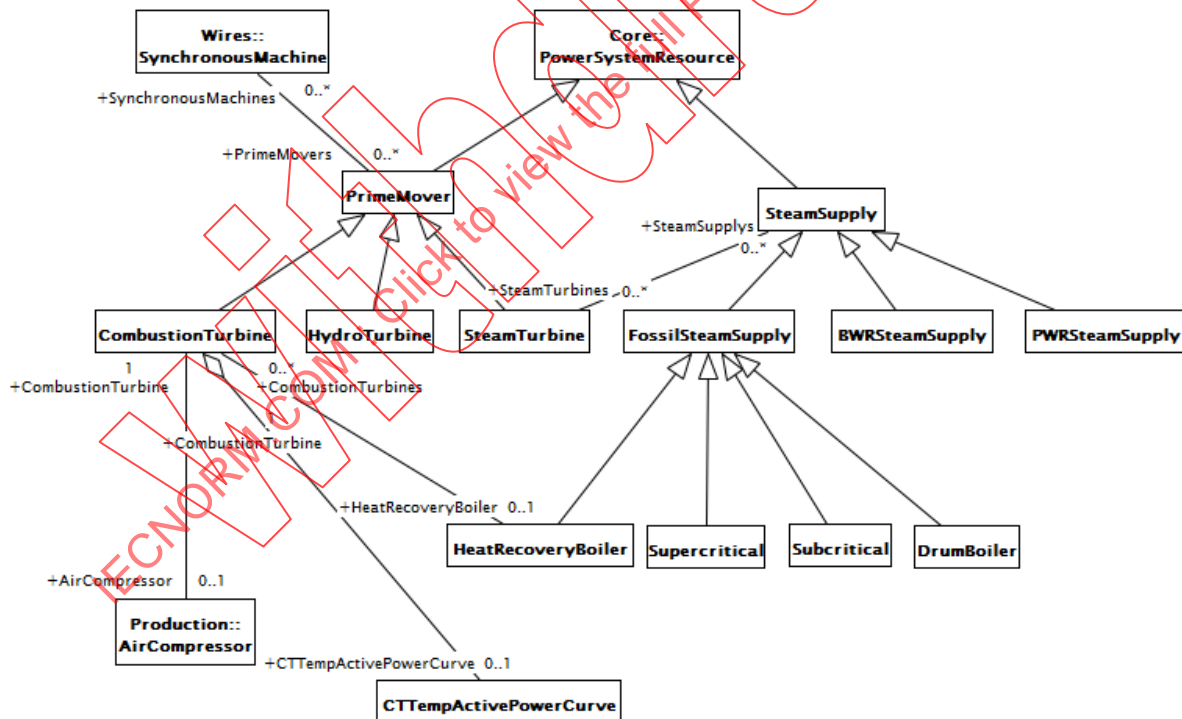


Figure 61 – Class diagram GenerationDynamics::Main

This diagram shows all classes included in the GenerationDynamics package as well as the key external classes that have associations with these classes.

Figure 62 shows class diagram Datatypes.

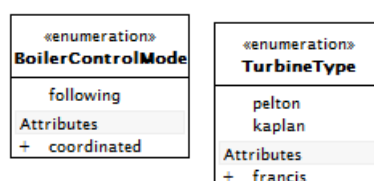


Figure 62 – Class diagram GenerationDynamics::Datatypes

This diagram shows the data types specific to the GenerationDynamics package.

6.9.2.2 BoilerControlMode enumeration

Boiler control mode.

Table 317 shows all literals of BoilerControlMode.

Table 317 – Literals of GenerationDynamics::BoilerControlMode

literal	description
following	Following.
coordinated	Coordinated.

6.9.2.3 BWRSteamSupply

Boiling water reactor used as a steam supply to a steam turbine.

Table 318 shows all attributes of BWRSteamSupply.

Table 318 – Attributes of GenerationDynamics::BWRSteamSupply

name	type	description
highPowerLimit	PU	High power limit.
inCoreThermalTC	Seconds	In-core thermal time constant.
integralGain	Float	Integral gain.
lowerLimit	PU	Initial lower limit.
lowPowerLimit	PU	Low power limit.
pressureLimit	PU	Pressure limit.
pressureSetpointGA	Float	Pressure setpoint gain adjuster.
pressureSetpointTC1	Seconds	Pressure setpoint time constant.
pressureSetpointTC2	Seconds	Pressure setpoint time constant.
proportionalGain	Float	Proportional gain.
rfAux1	PU	Coefficient for modeling the effect of off-nominal frequency and voltage on recirculation and core flow, which affects the BWR power output.
rfAux2	PU	Coefficient for modeling the effect of off-nominal frequency and voltage on recirculation and core flow, which affects the BWR power output.
rfAux3	PU	Coefficient for modeling the effect of off-nominal frequency and voltage on recirculation and core flow, which affects the BWR power output.
rfAux4	PU	Coefficient for modeling the effect of off-nominal frequency and voltage on recirculation and core flow, which affects the BWR power output.
rfAux5	PU	Coefficient for modeling the effect of off-nominal frequency and voltage on recirculation and core flow, which affects the BWR power output.

name	type	description
rfAux6	PU	Coefficient for modeling the effect of off-nominal frequency and voltage on recirculation and core flow, which affects the BWR power output.
rfAux7	PU	Coefficient for modeling the effect of off-nominal frequency and voltage on recirculation and core flow, which affects the BWR power output.
rfAux8	PU	Coefficient for modeling the effect of off-nominal frequency and voltage on recirculation and core flow, which affects the BWR power output.
rodPattern	PU	Rod pattern.
rodPatternConstant	Float	Constant associated with rod pattern.
upperLimit	PU	Initial upper limit.
steamSupplyRating	Float	inherited from: SteamSupply
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 319 shows all association ends of BWRSteamSupply with other classes.

Table 319 – Association ends of GenerationDynamics::BWRSteamSupply with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] SteamTurbines	SteamTurbine	inherited from: SteamSupply
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.2.4 CombustionTurbine

A prime mover that is typically fueled by gas or light oil.

Table 320 shows all attributes of CombustionTurbine.

Table 320 – Attributes of GenerationDynamics::CombustionTurbine

name	type	description
ambientTemp	Temperature	Default ambient temperature to be used in modeling applications.
auxPowerVersusFrequency	PU	Off-nominal frequency effect on turbine auxiliaries. Per unit reduction in auxiliary active power consumption versus per unit reduction in frequency (from rated frequency).

name	type	description
auxPowerVersusVoltage	PU	Off-nominal voltage effect on turbine auxiliaries. Per unit reduction in auxiliary active power consumption versus per unit reduction in auxiliary bus voltage (from a specified voltage level).
capabilityVersusFrequency	PU	Off-nominal frequency effect on turbine capability. Per unit reduction in unit active power capability versus per unit reduction in frequency (from rated frequency).
heatRecoveryFlag	Boolean	Flag that is set to true if the combustion turbine is associated with a heat recovery boiler.
powerVariationByTemp	PU	Per unit change in power per (versus) unit change in ambient temperature.
referenceTemp	Temperature	Reference temperature at which the output of the turbine was defined.
timeConstant	Seconds	The time constant for the turbine.
primeMoverRating	Float	inherited from: PrimeMover
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 321 shows all association ends of CombustionTurbine with other classes.

Table 321 – Association ends of GenerationDynamics::CombustionTurbine with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..1] AirCompressor	AirCompressor	A CAES air compressor is driven by combustion turbine.
[0..*]	[0..1] HeatRecoveryBoiler	HeatRecoveryBoiler	A combustion turbine may have a heat recovery boiler for making steam.
[1..1]	[0..1] CTTempActivePowerCurve	CTTempActivePowerCurve	A combustion turbine may have an active power versus ambient temperature relationship.
[0..*]	[0..*] SynchronousMachines	SynchronousMachine	inherited from: PrimeMover
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.2.5 CTTempActivePowerCurve

Relationship between the combustion turbine's power output rating in gross active power (X-axis) and the ambient air temperature (Y-axis).

Table 322 shows all attributes of CTTempActivePowerCurve.

Table 322 – Attributes of GenerationDynamics::CTTempActivePowerCurve

name	type	description
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 323 shows all association ends of CTTempActivePowerCurve with other classes.

Table 323 – Association ends of GenerationDynamics::CTTempActivePowerCurve with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] CombustionTurbine	CombustionTurbine	A combustion turbine may have an active power versus ambient temperature relationship.
[1..1]	[0..*] CurveData	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.2.6 DrumBoiler

Drum boiler.

Table 324 shows all attributes of DrumBoiler.

Table 324 – Attributes of GenerationDynamics::DrumBoiler

name	type	description
drumBoilerRating	Float	Rating of drum boiler in steam units.
auxPowerVersusFrequency	PU	inherited from: FossilSteamSupply
auxPowerVersusVoltage	PU	inherited from: FossilSteamSupply
boilerControlMode	BoilerControlMode	inherited from: FossilSteamSupply
controlErrorBiasP	Float	inherited from: FossilSteamSupply
controlIIC	Float	inherited from: FossilSteamSupply
controlIPC	Float	inherited from: FossilSteamSupply
controlPEB	Float	inherited from: FossilSteamSupply
controlPED	PU	inherited from: FossilSteamSupply

name	type	description
controlTC	Float	inherited from: FossilSteamSupply
feedWaterIG	Float	inherited from: FossilSteamSupply
feedWaterPG	Float	inherited from: FossilSteamSupply
feedWaterTC	Seconds	inherited from: FossilSteamSupply
fuelDemandLimit	PU	inherited from: FossilSteamSupply
fuelSupplyDelay	Seconds	inherited from: FossilSteamSupply
fuelSupplyTC	Seconds	inherited from: FossilSteamSupply
maxErrorRateP	Float	inherited from: FossilSteamSupply
mechPowerSensorLag	Seconds	inherited from: FossilSteamSupply
minErrorRateP	Float	inherited from: FossilSteamSupply
pressureCtrlDG	Float	inherited from: FossilSteamSupply
pressureCtrlIG	Float	inherited from: FossilSteamSupply
pressureCtrlPG	Float	inherited from: FossilSteamSupply
pressureFeedback	Integer	inherited from: FossilSteamSupply
superHeater1Capacity	Float	inherited from: FossilSteamSupply
superHeater2Capacity	Float	inherited from: FossilSteamSupply
superHeaterPipePD	Float	inherited from: FossilSteamSupply
throttlePressureSP	PU	inherited from: FossilSteamSupply
steamSupplyRating	Float	inherited from: SteamSupply
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 325 shows all association ends of DrumBoiler with other classes.

Table 325 – Association ends of GenerationDynamics::DrumBoiler with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] SteamTurbines	SteamTurbine	inherited from: SteamSupply
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.2.7 FossilSteamSupply

Fossil fueled boiler (e.g., coal, oil, gas).

Table 326 shows all attributes of FossilSteamSupply.

Table 326 – Attributes of GenerationDynamics::FossilSteamSupply

name	type	description
auxPowerVersusFrequency	PU	Off nominal frequency effect on auxiliary real power. Per unit active power variation versus per unit frequency variation.
auxPowerVersusVoltage	PU	Off nominal voltage effect on auxiliary real power. Per unit active power variation versus per unit voltage variation.
boilerControlMode	BoilerControlMode	The control mode of the boiler.
controlErrorBiasP	Float	Active power error bias ratio.
controlIC	Float	Integral constant.
controlPC	Float	Proportional constant.
controlPEB	Float	Pressure error bias ratio.
controlPED	PU	Pressure error deadband.
controlTC	Float	Time constant.
feedWaterIG	Float	Feedwater integral gain ratio.
feedWaterPG	Float	Feedwater proportional gain ratio.
feedWaterTC	Seconds	Feedwater time constant ratio.
fuelDemandLimit	PU	Fuel demand limit.
fuelSupplyDelay	Seconds	Fuel delay.
fuelSupplyTC	Seconds	Fuel supply time constant.
maxErrorRateP	Float	Active power maximum error rate limit.
mechPowerSensorLag	Seconds	Mechanical power sensor lag.
minErrorRateP	Float	Active power minimum error rate limit.
pressureCtrlDG	Float	Pressure control derivative gain ratio.
pressureCtrlIG	Float	Pressure control integral gain ratio.
pressureCtrlPG	Float	Pressure control proportional gain ratio.
pressureFeedback	Integer	Pressure feedback indicator.
superHeater1Capacity	Float	Drum/primary superheater capacity.
superHeater2Capacity	Float	Secondary superheater capacity.
superHeaterPipePD	Float	Superheater pipe pressure drop constant.
throttlePressureSP	PU	Throttle pressure setpoint.
steamSupplyRating	Float	inherited from: SteamSupply
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 327 shows all association ends of FossilSteamSupply with other classes.

Table 327 – Association ends of GenerationDynamics::FossilSteamSupply with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] SteamTurbines	SteamTurbine	inherited from: SteamSupply
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.2.8 HeatRecoveryBoiler

The heat recovery system associated with combustion turbines in order to produce steam for combined cycle plants.

Table 328 shows all attributes of HeatRecoveryBoiler.

Table 328 – Attributes of GenerationDynamics::HeatRecoveryBoiler

name	type	description
steamSupplyRating2	Float	The steam supply rating in kilopounds per hour, if dual pressure boiler.
auxPowerVersusFrequency	PU	inherited from: FossilSteamSupply
auxPowerVersusVoltage	PU	inherited from: FossilSteamSupply
boilerControlMode	BoilerControlMode	inherited from: FossilSteamSupply
controlErrorBiasP	Float	inherited from: FossilSteamSupply
controlIC	Float	inherited from: FossilSteamSupply
controlIPC	Float	inherited from: FossilSteamSupply
controlPEB	Float	inherited from: FossilSteamSupply
controlPED	PU	inherited from: FossilSteamSupply
controlTC	Float	inherited from: FossilSteamSupply
feedWaterIG	Float	inherited from: FossilSteamSupply
feedWaterPG	Float	inherited from: FossilSteamSupply
feedWaterTC	Seconds	inherited from: FossilSteamSupply
fuelDemandLimit	PU	inherited from: FossilSteamSupply
fuelSupplyDelay	Seconds	inherited from: FossilSteamSupply
fuelSupplyTC	Seconds	inherited from: FossilSteamSupply
maxErrorRateP	Float	inherited from: FossilSteamSupply
mechPowerSensorLag	Seconds	inherited from: FossilSteamSupply
minErrorRateP	Float	inherited from: FossilSteamSupply
pressureCtrlIDG	Float	inherited from: FossilSteamSupply
pressureCtrlIG	Float	inherited from: FossilSteamSupply
pressureCtrlIPG	Float	inherited from: FossilSteamSupply

name	type	description
pressureFeedback	Integer	inherited from: FossilSteamSupply
superHeater1Capacity	Float	inherited from: FossilSteamSupply
superHeater2Capacity	Float	inherited from: FossilSteamSupply
superHeaterPipePD	Float	inherited from: FossilSteamSupply
throttlePressureSP	PU	inherited from: FossilSteamSupply
steamSupplyRating	Float	inherited from: SteamSupply
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 329 shows all association ends of HeatRecoveryBoiler with other classes.

Table 329 – Association ends of GenerationDynamics::HeatRecoveryBoiler with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] CombustionTurbines	CombustionTurbine	A combustion turbine may have a heat recovery boiler for making steam.
[0..*]	[0..*] SteamTurbines	SteamTurbine	inherited from: SteamSupply
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.2.9 HydroTurbine

A water driven prime mover. Typical turbine types are: Francis, Kaplan, and Pelton.

Table 330 shows all attributes of HydroTurbine.

Table 330 – Attributes of GenerationDynamics::HydroTurbine

name	type	description
gateRateLimit	Float	Gate rate limit.
gateUpperLimit	PU	Gate upper limit.
maxHeadMaxP	ActivePower	Maximum efficiency active power at maximum head conditions.
minHeadMaxP	ActivePower	Maximum efficiency active power at minimum head conditions.
speedRating	RotationSpeed	Rated speed in number of revolutions.
speedRegulation	PU	Speed regulation.
transientDroopTime	Seconds	Transient droop time constant.
transientRegulation	PU	Transient regulation.
turbineRating	ActivePower	Rated turbine active power.
turbineType	TurbineType	Type of turbine.
waterStartingTime	Seconds	Water starting time.
primeMoverRating	Float	inherited from: PrimeMover
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 331 shows all association ends of HydroTurbine with other classes.

Table 331 – Association ends of GenerationDynamics::HydroTurbine with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] SynchronousMachines	SynchronousMachine	inherited from: PrimeMover
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.2.10 PrimeMover

The machine used to develop mechanical energy used to drive a generator.

Table 332 shows all attributes of PrimeMover.

Table 332 – Attributes of GenerationDynamics::PrimeMover

name	type	description
primeMoverRating	Float	Rating of prime mover.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 333 shows all association ends of PrimeMover with other classes.

Table 333 – Association ends of GenerationDynamics::PrimeMover with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] SynchronousMachines	SynchronousMachine	Synchronous machines this Prime mover drives.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.2.11 PWRSteamSupply

Pressurized water reactor used as a steam supply to a steam turbine.

Table 334 shows all attributes of PWRSteamSupply.

Table 334 – Attributes of GenerationDynamics::PWRSteamSupply

name	type	description
coldLegFBLagTC	PU	Cold leg feedback lag time constant.
coldLegFBLeadTC1	PU	Cold leg feedback lead time constant.
coldLegFBLeadTC2	PU	Cold leg feedback lead time constant.
coldLegFG1	PU	Cold leg feedback gain 1.
coldLegFG2	PU	Cold leg feedback gain 2.
coldLegLagTC	PU	Cold leg lag time constant.
coreHTLagTC1	PU	Core heat transfer lag time constant.
coreHTLagTC2	PU	Core heat transfer lag time constant.
coreNeutronicsEffTC	PU	Core neutronics effective time constant.
coreNeutronicsHT	PU	Core neutronics and heat transfer.
feedbackFactor	PU	Feedback factor.
hotLegLagTC	PU	Hot leg lag time constant.
hotLegSteamGain	PU	Hot leg steam gain.
hotLegToColdLegGain	PU	Hot leg to cold leg gain.

name	type	description
pressureCG	PU	Pressure control gain.
steamFlowFG	PU	Steam flow feedback gain.
steamPressureDropLagTC	PU	Steam pressure drop lag time constant.
steamPressureFG	PU	Steam pressure feedback gain.
throttlePressureFactor	PU	Throttle pressure factor.
throttlePressureSP	PU	Throttle pressure setpoint.
steamSupplyRating	Float	inherited from: SteamSupply
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 335 shows all association ends of PWRSteamSupply with other classes.

Table 335 – Association ends of GenerationDynamics::PWRSteamSupply with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] SteamTurbines	SteamTurbine	inherited from: SteamSupply
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.2.12 SteamSupply

Steam supply for steam turbine.

Table 336 shows all attributes of SteamSupply.

Table 336 – Attributes of GenerationDynamics::SteamSupply

name	type	description
steamSupplyRating	Float	Rating of steam supply.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 337 shows all association ends of SteamSupply with other classes.

Table 337 – Association ends of GenerationDynamics::SteamSupply with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] SteamTurbines	SteamTurbine	Steam turbines may have steam supplied by a steam supply.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.2.13 SteamTurbine

Steam turbine.

Table 338 shows all attributes of SteamTurbine.

Table 338 – Attributes of GenerationDynamics::SteamTurbine

name	type	description
crossoverTC	Seconds	Crossover time constant.
reheater1TC	Seconds	First reheater time constant.
reheater2TC	Seconds	Second reheater time constant.
shaft1PowerHP	Float	Fraction of power from shaft 1 high pressure turbine output.
shaft1PowerIP	Float	Fraction of power from shaft 1 intermediate pressure turbine output.
shaft1PowerLP1	Float	Fraction of power from shaft 1 first low pressure turbine output.
shaft1PowerLP2	Float	Fraction of power from shaft 1 second low pressure turbine output.
shaft2PowerHP	Float	Fraction of power from shaft 2 high pressure turbine output.
shaft2PowerIP	Float	Fraction of power from shaft 2 intermediate pressure turbine output.
shaft2PowerLP1	Float	Fraction of power from shaft 2 first low pressure turbine output.
shaft2PowerLP2	Float	Fraction of power from shaft 2 second low pressure turbine output.
steamChestTC	Seconds	Steam chest time constant.
primeMoverRating	Float	inherited from: PrimeMover
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 339 shows all association ends of SteamTurbine with other classes.

Table 339 – Association ends of GenerationDynamics::SteamTurbine with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] SteamSupplys	SteamSupply	Steam turbines may have steam supplied by a steam supply.
[0..*]	[0..*] SynchronousMachines	SynchronousMachine	inherited from: PrimeMover
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.2.14 Subcritical

Once-through subcritical boiler.

Table 340 shows all attributes of Subcritical.

Table 340 – Attributes of GenerationDynamics::Subcritical

name	type	description
auxPowerVersusFrequency	PU	inherited from: FossilSteamSupply
auxPowerVersusVoltage	PU	inherited from: FossilSteamSupply
boilerControlMode	BoilerControlMode	inherited from: FossilSteamSupply
controlErrorBiasP	Float	inherited from: FossilSteamSupply
controlIC	Float	inherited from: FossilSteamSupply
controlPC	Float	inherited from: FossilSteamSupply
controlPEB	Float	inherited from: FossilSteamSupply
controlPED	PU	inherited from: FossilSteamSupply
controlTC	Float	inherited from: FossilSteamSupply
feedWaterIG	Float	inherited from: FossilSteamSupply
feedWaterPG	Float	inherited from: FossilSteamSupply
feedWaterTC	Seconds	inherited from: FossilSteamSupply
fuelDemandLimit	PU	inherited from: FossilSteamSupply
fuelSupplyDelay	Seconds	inherited from: FossilSteamSupply
fuelSupplyTC	Seconds	inherited from: FossilSteamSupply
maxErrorRateP	Float	inherited from: FossilSteamSupply
mechPowerSensorLag	Seconds	inherited from: FossilSteamSupply
minErrorRateP	Float	inherited from: FossilSteamSupply
pressureCtrlDG	Float	inherited from: FossilSteamSupply
pressureCtrlIG	Float	inherited from: FossilSteamSupply
pressureCtrlPG	Float	inherited from: FossilSteamSupply
pressureFeedback	Integer	inherited from: FossilSteamSupply
superHeater1Capacity	Float	inherited from: FossilSteamSupply
superHeater2Capacity	Float	inherited from: FossilSteamSupply
superHeaterPipePD	Float	inherited from: FossilSteamSupply
throttlePressureSP	PU	inherited from: FossilSteamSupply
steamSupplyRating	Float	inherited from: SteamSupply
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 341 shows all association ends of Subcritical with other classes.

Table 341 – Association ends of GenerationDynamics::Subcritical with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] SteamTurbines	SteamTurbine	inherited from: SteamSupply
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.2.15 Supercritical

Once-through supercritical boiler.

Table 342 shows all attributes of Supercritical.

Table 342 – Attributes of GenerationDynamics::Supercritical

name	type	description
auxPowerVersusFrequency	PU	inherited from: FossilSteamSupply
auxPowerVersusVoltage	PU	inherited from: FossilSteamSupply
boilerControlMode	BoilerControlMode	inherited from: FossilSteamSupply
controlErrorBiasP	Float	inherited from: FossilSteamSupply
controlIC	Float	inherited from: FossilSteamSupply
controlPC	Float	inherited from: FossilSteamSupply
controlPEB	Float	inherited from: FossilSteamSupply
controlPED	PU	inherited from: FossilSteamSupply
controlTC	Float	inherited from: FossilSteamSupply
feedWaterIG	Float	inherited from: FossilSteamSupply
feedWaterPG	Float	inherited from: FossilSteamSupply
feedWaterTC	Seconds	inherited from: FossilSteamSupply
fuelDemandLimit	PU	inherited from: FossilSteamSupply
fuelSupplyDelay	Seconds	inherited from: FossilSteamSupply
fuelSupplyTC	Seconds	inherited from: FossilSteamSupply
maxErrorRateP	Float	inherited from: FossilSteamSupply
mechPowerSensorLag	Seconds	inherited from: FossilSteamSupply
minErrorRateP	Float	inherited from: FossilSteamSupply
pressureCtrlDG	Float	inherited from: FossilSteamSupply
pressureCtrlIG	Float	inherited from: FossilSteamSupply
pressureCtrlPG	Float	inherited from: FossilSteamSupply
pressureFeedback	Integer	inherited from: FossilSteamSupply

name	type	description
superHeater1Capacity	Float	inherited from: FossilSteamSupply
superHeater2Capacity	Float	inherited from: FossilSteamSupply
superHeaterPipePD	Float	inherited from: FossilSteamSupply
throttlePressureSP	PU	inherited from: FossilSteamSupply
steamSupplyRating	Float	inherited from: SteamSupply
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 343 shows all association ends of Supercritical with other classes.

Table 343 – Association ends of GenerationDynamics::Supercritical with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] SteamTurbines	SteamTurbine	inherited from: SteamSupply
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.2.16 TurbineType enumeration

Type of turbine.

Table 344 shows all literals of TurbineType.

Table 344 – Literals of GenerationDynamics::TurbineType

literal	description
francis	Francis.
pelton	Pelton.
kaplan	Kaplan.

6.9.3 Package Production

6.9.3.1 General

The production package is responsible for classes which describe various kinds of generators. These classes also provide production costing information which is used to economically allocate demand among committed units and calculate reserve quantities.

Figure 63 shows class diagram Nuclear.

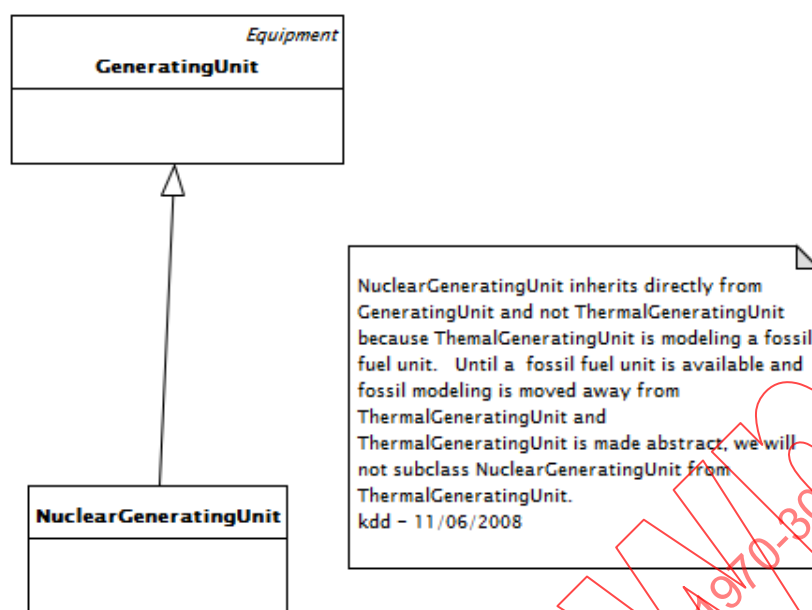


Figure 63 – Class diagram Production: Nuclear

This diagram shows nuclear generation inheritance.

Figure 64 shows class diagram Main.

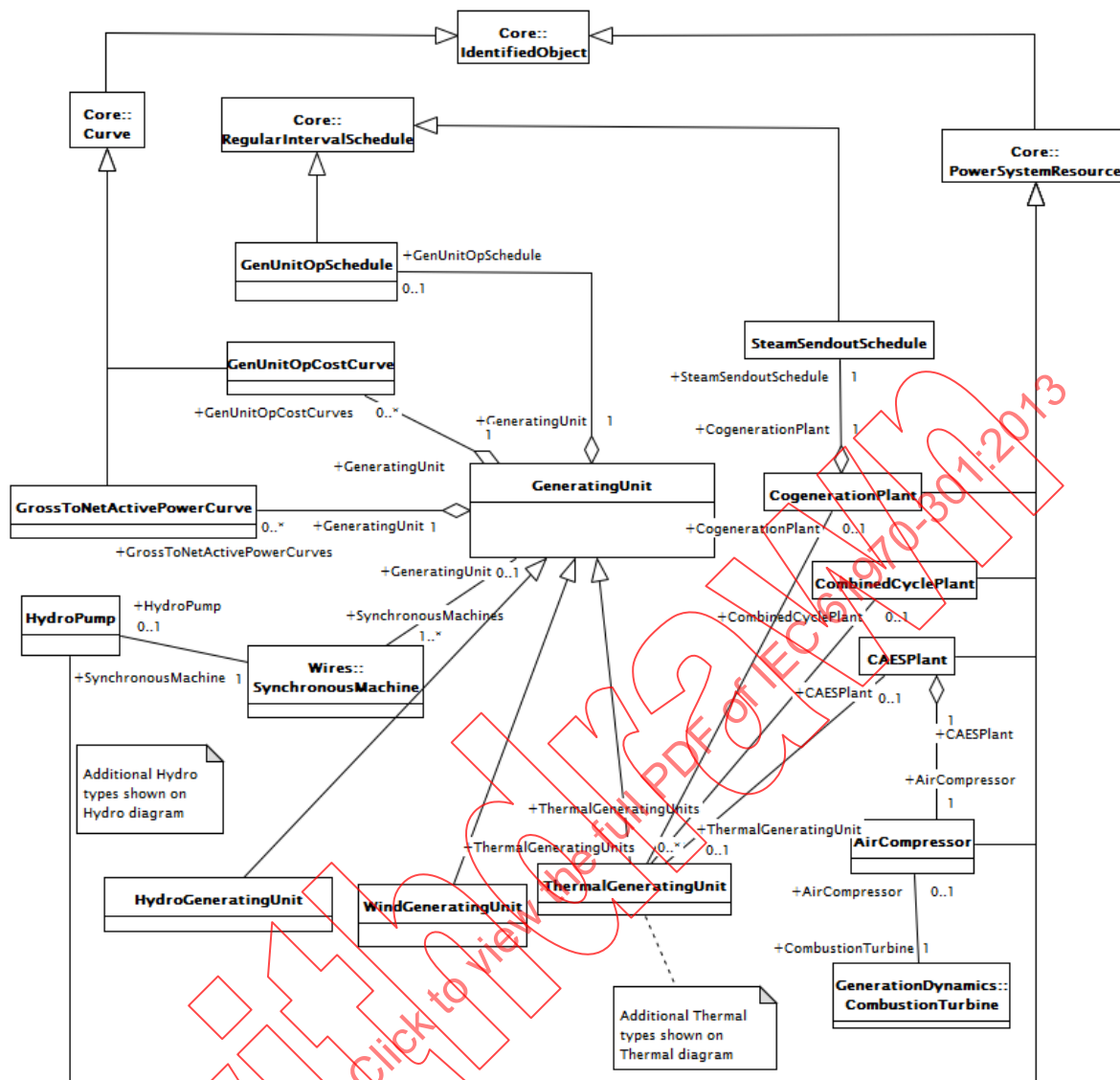


Figure 64 – Class diagram Production::Main

This diagram shows all classes included in the Production package that are needed by both Hydro and Thermal generation. It also shows key external classes that have associations with Production classes.

Figure 65 shows class diagram Datatypes.

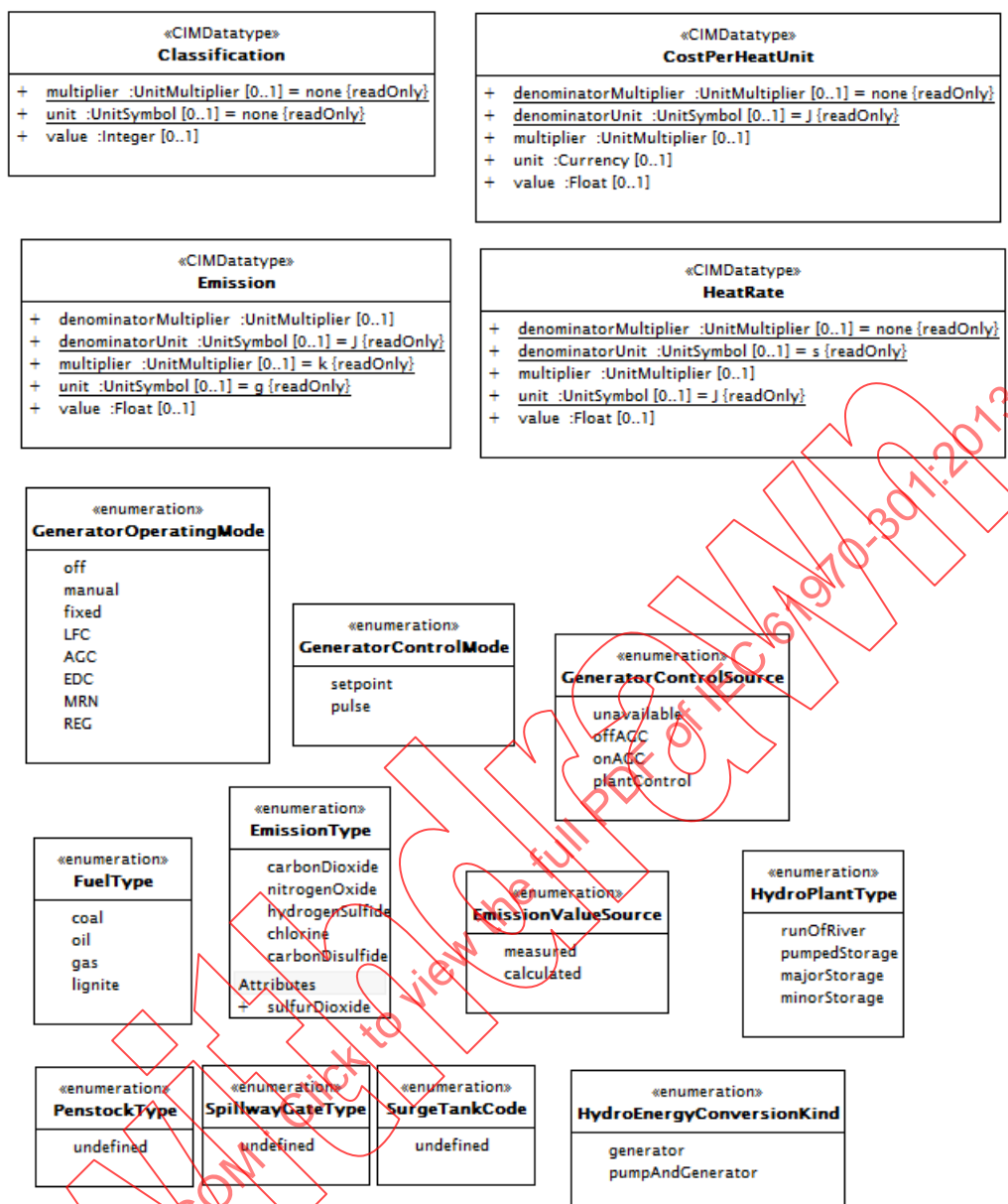


Figure 65 – Class diagram Production::Datatypes

This diagram show the data types specific to the Production package.

Figure 66 shows class diagram Hydro.

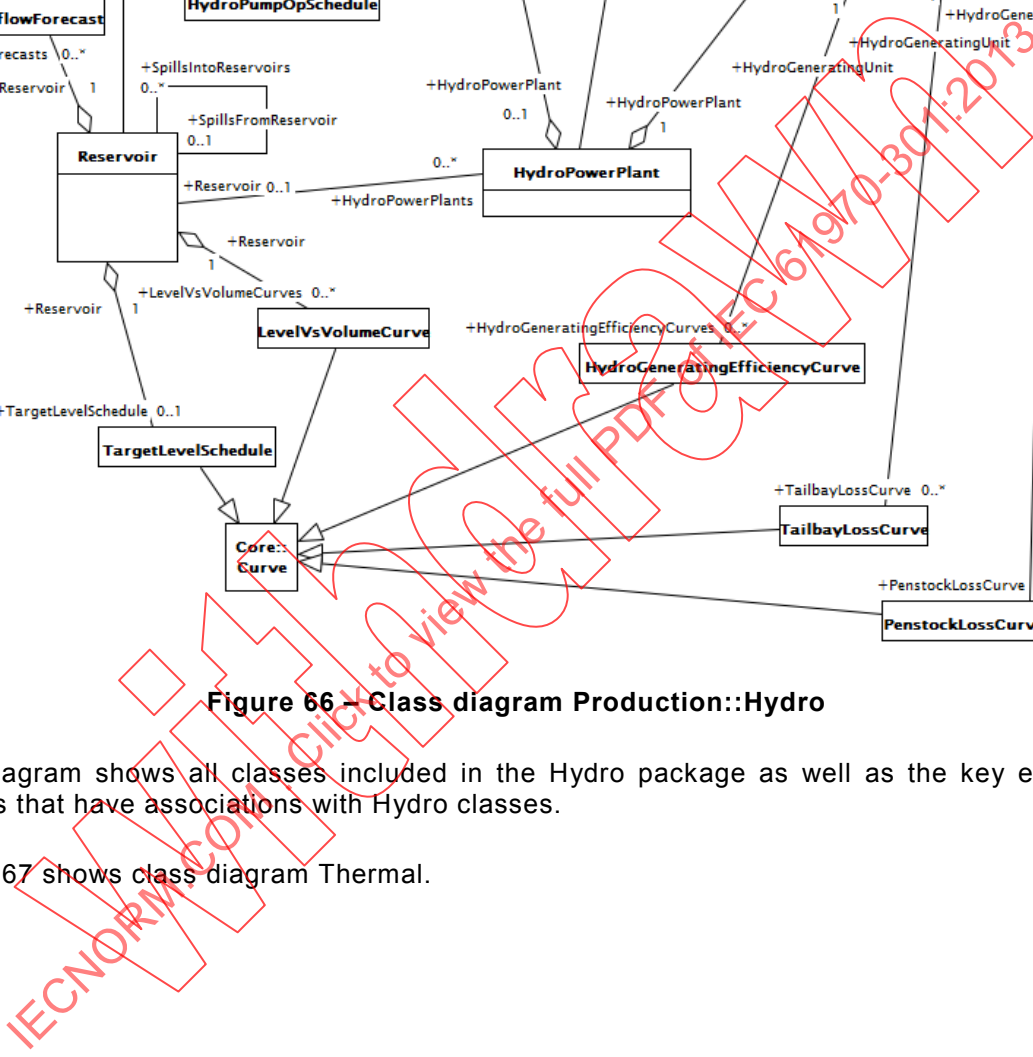


Figure 66 – Class diagram Production::Hydro

This diagram shows all classes included in the Hydro package as well as the key external classes that have associations with Hydro classes.

Figure 67 shows class diagram Thermal.

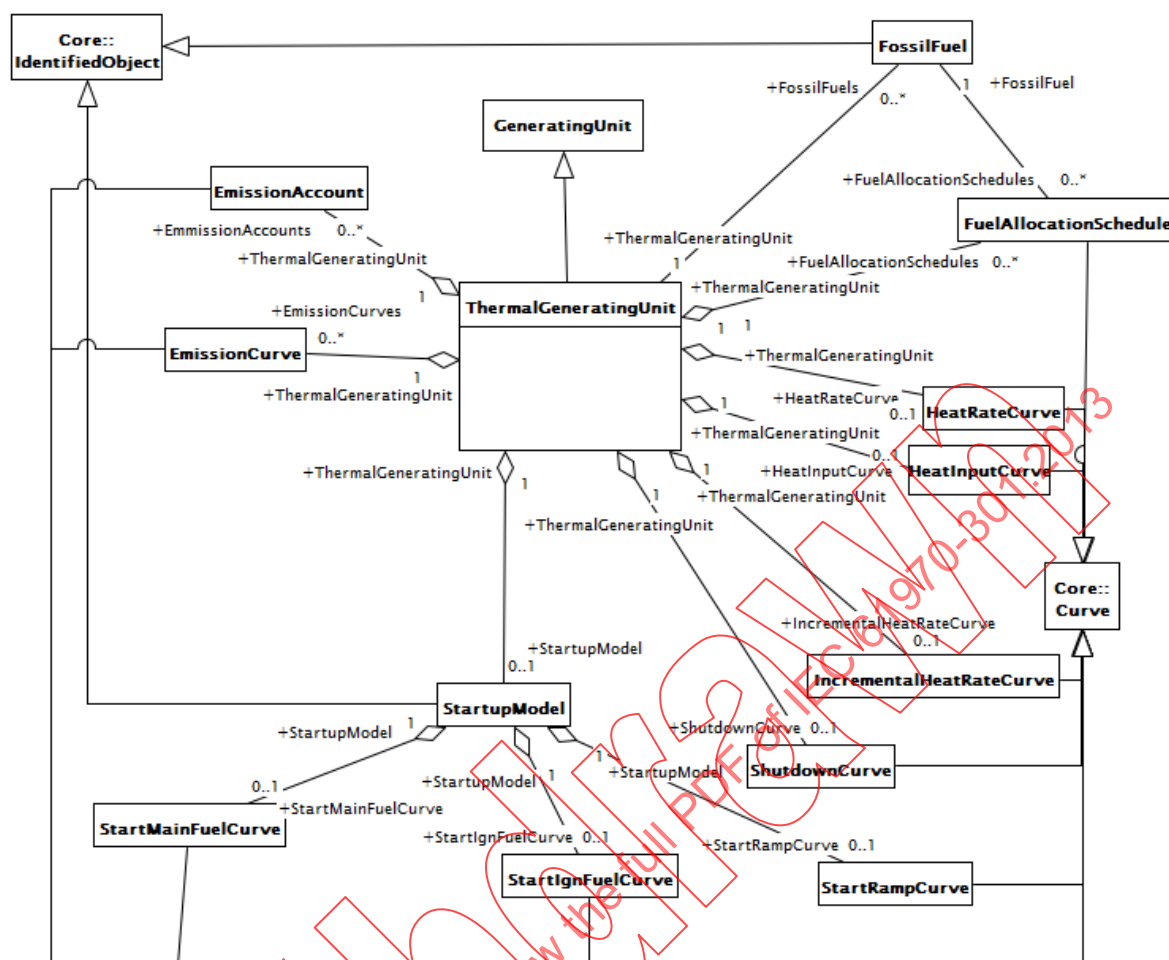


Figure 67 – Class diagram Production::Thermal

This diagram shows all classes included in the Thermal package as well as the key external classes that have associations with Thermal classes.

6.9.3.2 AirCompressor

Combustion turbine air compressor which is an integral part of a compressed air energy storage (CAES) plant.

Table 345 shows all attributes of AirCompressor.

Table 345 – Attributes of Production::AirCompressor

name	type	description
airCompressorRating	Float	Rating of the CAES air compressor.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 346 shows all association ends of AirCompressor with other classes.

Table 346 – Association ends of Production::AirCompressor with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] CombustionTurbine	CombustionTurbine	A CAES air compressor is driven by combustion turbine.
[1..1]	[1..1] CAESPlant	CAESPlant	An air compressor may be a member of a compressed air energy storage plant.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.3 CAESPlant

Compressed air energy storage plant.

Table 347 shows all attributes of CAESPlant.

Table 347 – Attributes of Production::CAESPlant

name	type	description
energyStorageCapacity	RealEnergy	The rated energy storage capacity.
ratedCapacityP	ActivePower	The CAES plant's gross rated generating capacity.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 348 shows all association ends of CAESPlant with other classes.

Table 348 – Association ends of Production::CAESPlant with other classes

[mult from]	[mult to] name	type	description
[1..1]	[1..1] AirCompressor	AirCompressor	An air compressor may be a member of a compressed air energy storage plant.
[0..1]	[0..1] ThermalGeneratingUnit	ThermalGeneratingUnit	A thermal generating unit may be a member of a compressed air energy storage plant.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.4 Classification datatype

Classification of level. Specify as 1..n, with 1 being the most detailed, highest priority, etc as described on the attribute using this data type.

Table 349 shows all attributes of Classification.

Table 349 – Attributes of Production::Classification

name	type	description
value	Integer	
unit=none (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	

6.9.3.5 CogenerationPlant

A set of thermal generating units for the production of electrical energy and process steam (usually from the output of the steam turbines). The steam sendout is typically used for industrial purposes or for municipal heating and cooling.

Table 350 shows all attributes of CogenerationPlant.

Table 350 – Attributes of Production::CogenerationPlant

name	type	description
cogenHPSendoutRating	Float	The high pressure steam sendout.
cogenHPSteamRating	Float	The high pressure steam rating.
cogenLPSendoutRating	Float	The low pressure steam sendout.
cogenLPSteamRating	Float	The low pressure steam rating.
ratedP	ActivePower	The rated output active power of the cogeneration plant.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 351 shows all association ends of CogenerationPlant with other classes.

Table 351 – Association ends of Production::CogenerationPlant with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] ThermalGeneratingUnits	ThermalGeneratingUnit	A thermal generating unit may be a member of a cogeneration plant.
[1..1]	[1..1] SteamSendoutSchedule	SteamSendoutSchedule	A cogeneration plant has a steam sendout schedule.
[0..*]	[0..1] PSRTYPE	PSRTYPE	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.6 CombinedCyclePlant

A set of combustion turbines and steam turbines where the exhaust heat from the combustion turbines is recovered to make steam for the steam turbines, resulting in greater overall plant efficiency.

Table 352 shows all attributes of CombinedCyclePlant.

Table 352 – Attributes of Production::CombinedCyclePlant

name	type	description
combCyclePlantRating	ActivePower	The combined cycle plant's active power output rating.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 353 shows all association ends of CombinedCyclePlant with other classes.

Table 353 – Association ends of Production::CombinedCyclePlant with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] ThermalGeneratingUnits	ThermalGeneratingUnit	A thermal generating unit may be a member of a combined cycle plant.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.7 CostPerHeatUnit datatype

Cost, in units of currency, per quantity of heat generated.

Table 354 shows all attributes of CostPerHeatUnit.

Table 354 – Attributes of Production::CostPerHeatUnit

name	type	description
value	Float	
unit	Currency	
multiplier	UnitMultiplier	
denominatorUnit=J (const)	UnitSymbol	
denominatorMultiplier=none (const)	UnitMultiplier	

6.9.3.8 Emission datatype

Quantity of emission per fuel heat content.

Table 355 shows all attributes of Emission.

Table 355 – Attributes of Production::Emission

name	type	description
value	Float	
unit=g (const)	UnitSymbol	
multiplier=k (const)	UnitMultiplier	
denominatorUnit=J (const)	UnitSymbol	
denominatorMultiplier	UnitMultiplier	

6.9.3.9 EmissionAccount

Accounts for tracking emissions usage and credits for thermal generating units. A unit may have zero or more emission accounts, and will typically have one for tracking usage and one for tracking credits.

Table 356 shows all attributes of EmissionAccount.

Table 356 – Attributes of Production::EmissionAccount

name	type	description
emissionType	EmissionType	The type of emission, for example sulfur dioxide (SO ₂). The y1AxisUnits of the curve contains the unit of measure (e.g. kg) and the emissionType is the type of emission (e.g. sulfur dioxide).
emissionValueSource	EmissionValueSource	The source of the emission value.
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 357 shows all association ends of EmissionAccount with other classes.

Table 357 – Association ends of Production::EmissionAccount with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] ThermalGeneratingUnit	ThermalGeneratingUnit	A thermal generating unit may have one or more emission allowance accounts.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.10 EmissionCurve

Relationship between the unit's emission rate in units of mass per hour (Y-axis) and output active power (X-axis) for a given type of emission. This curve applies when only one type of fuel is being burned.

Table 358 shows all attributes of EmissionCurve.

Table 358 – Attributes of Production::EmissionCurve

name	type	description
emissionContent	Emission	The emission content per quantity of fuel burned.
emissionType	EmissionType	The type of emission, which also gives the production rate measurement unit. The y1AxisUnits of the curve contains the unit of measure (e.g. kg) and the emissionType is the type of emission (e.g. sulfur dioxide).
isNetGrossP	Boolean	Flag is set to true when output is expressed in net active power.
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 359 shows all association ends of EmissionCurve with other classes.

Table 359 – Association ends of Production::EmissionCurve with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] ThermalGeneratingUnit	ThermalGeneratingUnit	A thermal generating unit may have one or more emission curves.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.11 EmissionType enumeration

The type of emission.

Table 360 shows all literals of EmissionType.

Table 360 – Literals of Production::EmissionType

literal	description
sulfurDioxide	Sulfur dioxide.
carbonDioxide	Carbon dioxide.
nitrogenOxide	Nitrogen oxide.
hydrogenSulfide	Hydrogen sulfide.
chlorine	Chlorine.
carbonDisulfide	Carbon disulfide.

6.9.3.12 EmissionValueSource enumeration

The source of the emission value.

Table 361 shows all literals of EmissionValueSource.

Table 361 – Literals of Production::EmissionValueSource

literal	description
measured	Measured.
calculated	Calculated.

6.9.3.13 FossilFuel

The fossil fuel consumed by the non-nuclear thermal generating unit. For example, coal, oil, gas, etc. This is the specific fuels that the generating unit can consume.

Table 362 shows all attributes of FossilFuel.

Table 362 – Attributes of Production::FossilFuel

name	type	description
fossilFuelType	FuelType	The type of fossil fuel, such as coal, oil, or gas.
fuelCost	CostPerHeatUnit	The cost in terms of heat value for the given type of fuel.
fuelDispatchCost	CostPerHeatUnit	The cost of fuel used for economic dispatching which includes: fuel cost, transportation cost, and incremental maintenance cost.
fuelEffFactor	PU	The efficiency factor for the fuel (per unit) in terms of the effective energy absorbed.
fuelHandlingCost	CostPerHeatUnit	Handling and processing cost associated with this fuel.
fuelHeatContent	Float	The amount of heat per weight (or volume) of the given type of fuel.
fuelMixture	PerCent	Relative amount of the given type of fuel, when multiple fuels are being consumed.
fuelSulfur	PU	The fuel's fraction of pollution credit per unit of heat content.
highBreakpointP	ActivePower	The active power output level of the unit at which the given type of fuel is switched on. This fuel (e.g., oil) is sometimes used to supplement the base fuel (e.g., coal) at high active power output levels.
lowBreakpointP	ActivePower	The active power output level of the unit at which the given type of fuel is switched off. This fuel (e.g., oil) is sometimes used to stabilize the base fuel (e.g., coal) at low active power output levels.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 363 shows all association ends of FossilFuel with other classes.

Table 363 – Association ends of Production::FossilFuel with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] FuelAllocationSchedules	FuelAllocationSchedule	A fuel allocation schedule must have a fossil fuel.
[0..*]	[1..1] ThermalGeneratingUnit	ThermalGeneratingUnit	A thermal generating unit may have one or more fossil fuels.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.14 FuelAllocationSchedule

The amount of fuel of a given type which is allocated for consumption over a specified period of time.

Table 364 shows all attributes of FuelAllocationSchedule.

Table 364 – Attributes of Production::FuelAllocationSchedule

name	type	description
fuelAllocationEndDate	DateTime	The end time and date of the fuel allocation schedule.
fuelAllocationStartDate	DateTime	The start time and date of the fuel allocation schedule.
fuelType	FuelType	The type of fuel, which also indicates the corresponding measurement unit.
maxFuelAllocation	Float	The maximum amount fuel that is allocated for consumption for the scheduled time period.
minFuelAllocation	Float	The minimum amount fuel that is allocated for consumption for the scheduled time period, e.g., based on a "take-or-pay" contract.
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 365 shows all association ends of FuelAllocationSchedule with other classes.

Table 365 – Association ends of Production::FuelAllocationSchedule with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] ThermalGeneratingUnit	ThermalGeneratingUnit	A thermal generating unit may have one or more fuel allocation schedules.
[0..*]	[1..1] FossilFuel	FossilFuel	A fuel allocation schedule must have a fossil fuel.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.15 FuelType enumeration

Type of fuel.

Table 366 shows all literals of FuelType.

Table 366 – Literals of Production::FuelType

literal	description
coal	Generic coal, not including lignite type.
oil	Oil.
gas	Gas.
lignite	The fuel is lignite coal. Note that this is a special type of coal, so the other enum of coal is reserved for hard coal types or if the exact type of coal is not known.

6.9.3.16 GeneratingUnit

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

Table 367 shows all attributes of GeneratingUnit.

Table 367 – Attributes of Production::GeneratingUnit

name	type	description
allocSpinResP	ActivePower	The planned unused capacity (spinning reserve) which can be used to support emergency load.
autoCntrlMarginP	ActivePower	The planned unused capacity which can be used to support automatic control overruns.
baseP	ActivePower	For dispatchable units, this value represents the economic active power basepoint, for units that are not dispatchable, this value represents the fixed generation value. The value must be between the operating low and high limits.
controlDeadband	ActivePower	Unit control error deadband. When a unit's desired active power change is less than this deadband, then no control pulses will be sent to the unit.
controlPulseHigh	Seconds	Pulse high limit which is the largest control pulse that the unit can respond to.

name	type	description
controlPulseLow	Seconds	Pulse low limit which is the smallest control pulse that the unit can respond to.
controlResponseRate	ActivePowerChangeRate	Unit response rate which specifies the active power change for a control pulse of one second in the most responsive loading level of the unit.
dispReserveFlag	Boolean	Recommended to be deleted from standard.
efficiency	PU	The efficiency of the unit in converting mechanical energy, from the prime mover, into electrical energy.
energyMinP	HeatRate	Recommended to be deleted from standard.
fastStartFlag	Boolean	Indicates the generating unit has a fast start capability and implies the generating unit can potentially contribute to spinning reserve even while not spinning.
fuelPriority	Integer	Recommended to be deleted from standard.
genControlMode	GeneratorControlMode	The unit control mode.
genControlSource	GeneratorControlSource	The source of controls for a generating unit.
genOperatingMode	GeneratorOperatingMode	Operating mode for secondary control.
governorMPL	PU	Governor motor position limit.
governorSCD	PerCent	Governor Speed Changer Droop. This is the change in generator power output divided by the change in frequency normalized by the nominal power of the generator and the nominal frequency and expressed in percent and negated. A positive value of speed change droop provides additional generator output upon a drop in frequency.
highControlLimit	ActivePower	High limit for secondary (AGC) control.
initialIP	ActivePower	Default initial active power which is used to store a powerflow result for the initial active power for this unit in this network configuration.
longPF	Float	Generating unit long term economic participation factor.
lowControlLimit	ActivePower	Low limit for secondary (AGC) control.
lowerRampRate	ActivePowerChangeRate	The normal maximum rate the generating unit active power output can be lowered by control actions.
maxEconomicP	ActivePower	Maximum high economic active power limit, that should not exceed the maximum operating active power limit.
maximumAllowableSpinningReserve	ActivePower	Maximum allowable spinning reserve. Spinning reserve will never be considered greater than this value regardless of the current operating point.
maxOperatingP	ActivePower	This is the maximum operating active power limit the dispatcher can enter for this unit.
minEconomicP	ActivePower	Low economic active power limit that must be greater than or equal to the minimum operating active power limit.
minimumOffTime	Seconds	Minimum time interval between unit shutdown and startup.
minOperatingP	ActivePower	This is the minimum operating active power limit the dispatcher can enter for this unit.
modelDetail	Classification	Detail level of the generator model data.
nominalP	ActivePower	The nominal power of the generating unit. Used to give precise meaning to percentage based attributes such as the governor speed change droop (governorSCD attribute).

name	type	description
normalPF	Float	Generating unit economic participation factor.
penaltyFactor	Float	Defined as: $1 / (1 - \text{Incremental Transmission Loss})$; with the Incremental Transmission Loss expressed as a plus or minus value. The typical range of penalty factors is (0.9 to 1.1).
raiseRampRate	ActivePowerChangeRate	The normal maximum rate the generating unit active power output can be raised by control actions.
ratedGrossMaxP	ActivePower	The unit's gross rated maximum capacity (book value).
ratedGrossMinP	ActivePower	The gross rated minimum generation level which the unit can safely operate at while delivering power to the transmission grid.
ratedNetMaxP	ActivePower	The net rated maximum capacity determined by subtracting the auxiliary power used to operate the internal plant machinery from the rated gross maximum capacity.
shortPF	Float	Generating unit short term economic participation factor.
spinReserveRamp	ActivePowerChangeRate	Recommended to be deleted from standard.
startupCost	Money	The initial startup cost incurred for each start of the GeneratingUnit.
startupTime	Seconds	Time it takes to get the unit on-line, from the time that the prime mover mechanical power is applied.
stepChange	ActivePower	Recommended to be deleted from standard.
tieLinePF	Float	Generating unit economic participation factor.
variableCost	Money	The variable cost component of production per unit of ActivePower.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 368 shows all association ends of GeneratingUnit with other classes.

Table 368 – Association ends of Production::GeneratingUnit with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..*] SynchronousMachines	SynchronousMachine	A synchronous machine may operate as a generator and as such becomes a member of a generating unit.
[1..1]	[0..1] GenUnitOpSchedule	GenUnitOpSchedule	A generating unit may have an operating schedule, indicating the planned operation of the unit.
[1..1]	[0..*] GrossToNetActivePowerCurves	GrossToNetActivePowerCurve	A generating unit may have a gross active power to net active power curve, describing the losses and auxiliary power requirements of the unit.
[1..1]	[0..*] GenUnitOpCostCurves	GenUnitOpCostCurve	A generating unit may have one or more cost curves, depending upon fuel mixture and fuel cost.
[1..1]	[0..*] ControlAreaGeneratingUnit	ControlAreaGeneratingUnit	ControlArea specifications for this generating unit.
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.17 GeneratorControlMode enumeration

Unit control modes.

Table 369 shows all literals of GeneratorControlMode.

Table 369 – Literals of Production::GeneratorControlMode

literal	description
setpoint	Setpoint control mode.
pulse	Pulse control mode.

6.9.3.18 GeneratorControlSource enumeration

The source of controls for a generating unit.

Table 370 shows all literals of GeneratorControlSource.

Table 370 – Literals of Production::GeneratorControlSource

literal	description
unavailable	Not available.
offAGC	Off of automatic generation control (AGC).
onAGC	On automatic generation control (AGC).
plantControl	Plant is controlling.

6.9.3.19 GeneratorOperatingMode enumeration

Operating mode for secondary generator control.

Table 371 shows all literals of GeneratorOperatingMode.

Table 371 – Literals of Production::GeneratorOperatingMode

literal	description
off	Off.
manual	Manual.
fixed	Fixed.
LFC	Load frequency control.
AGC	Automatic generation control (AGC).
EDC	Economic dispatch control.
MRN	MRN.
REG	REG.

6.9.3.20 GenUnitOpCostCurve

Relationship between unit operating cost (Y-axis) and unit output active power (X-axis). The operating cost curve for thermal units is derived from heat input and fuel costs. The operating cost curve for hydro units is derived from water flow rates and equivalent water costs.

Table 372 shows all attributes of GenUnitOpCostCurve.

Table 372 – Attributes of Production::GenUnitOpCostCurve

name	type	description
isNetGrossP	Boolean	Flag is set to true when output is expressed in net active power.
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 373 shows all association ends of GenUnitOpCostCurve with other classes.

Table 373 – Association ends of Production::GenUnitOpCostCurve with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] GeneratingUnit	GeneratingUnit	A generating unit may have one or more cost curves, depending upon fuel mixture and fuel cost.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.21 GenUnitOpSchedule

The generating unit's Operator-approved current operating schedule (or plan), typically produced with the aid of unit commitment type analyses. The X-axis represents absolute time. The Y1-axis represents the status (0=off-line and unavailable: 1=available: 2=must run: 3=must run at fixed power value: etc.). The Y2-axis represents the must run fixed power value where required.

Table 374 shows all attributes of GenUnitOpSchedule.

Table 374 – Attributes of Production::GenUnitOpSchedule

name	type	description
endTime	DateTime	inherited from: RegularIntervalSchedule
timeStep	Seconds	inherited from: RegularIntervalSchedule
startTime	DateTime	inherited from: BasicIntervalSchedule
value1Unit	UnitSymbol	inherited from: BasicIntervalSchedule
value1Multiplier	UnitMultiplier	inherited from: BasicIntervalSchedule
value2Unit	UnitSymbol	inherited from: BasicIntervalSchedule
value2Multiplier	UnitMultiplier	inherited from: BasicIntervalSchedule
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 375 shows all association ends of GenUnitOpSchedule with other classes.

Table 375 – Association ends of Production::GenUnitOpSchedule with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] GeneratingUnit	GeneratingUnit	A generating unit may have an operating schedule, indicating the planned operation of the unit.
[1..1]	[1..*] TimePoints	RegularTimePoint	inherited from: RegularIntervalSchedule
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.22 GrossToNetActivePowerCurve

Relationship between the generating unit's gross active power output on the X-axis (measured at the terminals of the machine(s)) and the generating unit's net active power output on the Y-axis (based on utility-defined measurements at the power station). Station service loads, when modeled, should be treated as non-conforming bus loads. There may be more than one curve, depending on the auxiliary equipment that is in service.

Table 376 shows all attributes of GrossToNetActivePowerCurve.

Table 376 – Attributes of Production::GrossToNetActivePowerCurve

name	type	description
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 377 shows all association ends of GrossToNetActivePowerCurve with other classes.

Table 377 – Association ends of Production::GrossToNetActivePowerCurve with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] GeneratingUnit	GeneratingUnit	A generating unit may have a gross active power to net active power curve, describing the losses and auxiliary power requirements of the unit.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.23 HeatInputCurve

Relationship between unit heat input in energy per time for main fuel (Y1-axis) and supplemental fuel (Y2-axis) versus unit output in active power (X-axis). The quantity of main fuel used to sustain generation at this output level is prorated for throttling between definition points. The quantity of supplemental fuel used at this output level is fixed and not prorated.

Table 378 shows all attributes of HeatInputCurve.

Table 378 – Attributes of Production::HeatInputCurve

name	type	description
auxPowerMult	PU	Power output – auxiliary power multiplier adjustment factor.
auxPowerOffset	ActivePower	Power output – auxiliary power offset adjustment factor.
heatInputEff	PU	Heat input – efficiency multiplier adjustment factor.
heatInputOffset	HeatRate	Heat input – offset adjustment factor.
isNetGrossP	Boolean	Flag is set to true when output is expressed in net active power.
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 379 shows all association ends of HeatInputCurve with other classes.

Table 379 – Association ends of Production::HeatInputCurve with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] ThermalGeneratingUnit	ThermalGeneratingUnit	A thermal generating unit may have a heat input curve.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.24 HeatRate datatype

Heat generated, in energy pertime unit of elapsed time.

Table 380 shows all attributes of HeatRate.

Table 380 – Attributes of Production::HeatRate

name	type	description
unit=J (const)	UnitSymbol	
value	Float	
multiplier	UnitMultiplier	
denominatorUnit=s (const)	UnitSymbol	
denominatorMultiplier=none (const)	UnitMultiplier	

6.9.3.25 HeatRateCurve

Relationship between unit heat rate per active power (Y-axis) and unit output (X-axis). The heat input is from all fuels.

Table 381 shows all attributes of HeatRateCurve.

Table 381 – Attributes of Production::HeatRateCurve

name	type	description
isNetGrossP	Boolean	Flag is set to true when output is expressed in net active power.
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 382 shows all association ends of HeatRateCurve with other classes.

Table 382 – Association ends of Production::HeatRateCurve with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] ThermalGeneratingUnit	ThermalGeneratingUnit	A thermal generating unit may have a heat rate curve.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.26 HydroEnergyConversionKind enumeration

Specifies the capability of the hydro generating unit to convert energy as a generator or pump.

Table 383 shows all literals of HydroEnergyConversionKind.

Table 383 – Literals of Production::HydroEnergyConversionKind

literal	description
generator	Able to generate power, but not able to pump water for energy storage.
pumpAndGenerator	Able to both generate power and pump water for energy storage.

6.9.3.27 HydroGeneratingEfficiencyCurve

Relationship between unit efficiency in percent and unit output active power for a given net head in metres. The relationship between efficiency, discharge, head, and power output is expressed as follows: $E = KP/HQ$

Where: (E=percentage) (P=active power) (H=height) (Q=volume/time unit) (K=constant)

For example, a curve instance for a given net head could relate efficiency (Y-axis) versus active power output (X-axis) or versus discharge on the X-axis.

Table 384 shows all attributes of HydroGeneratingEfficiencyCurve.

Table 384 – Attributes of Production::HydroGeneratingEfficiencyCurve

name	type	description
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 385 shows all association ends of HydroGeneratingEfficiencyCurve with other classes.

Table 385 – Association ends of Production::HydroGeneratingEfficiencyCurve with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] HydroGeneratingUnit	HydroGeneratingUnit	A hydro generating unit has an efficiency curve.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.28 HydroGeneratingUnit

A generating unit whose prime mover is a hydraulic turbine (e.g., Francis, Pelton, Kaplan).

Table 386 shows all attributes of HydroGeneratingUnit.

Table 386 – Attributes of Production::HydroGeneratingUnit

name	type	description
energyConversionCapability	HydroEnergyConversionKind	Energy conversion capability for generating.
hydroUnitWaterCost	CostPerVolume	The equivalent cost of water that drives the hydro turbine.
allocSpinResP	ActivePower	inherited from: GeneratingUnit
autoCntrlMarginP	ActivePower	inherited from: GeneratingUnit
baseP	ActivePower	inherited from: GeneratingUnit
controlDeadband	ActivePower	inherited from: GeneratingUnit
controlPulseHigh	Seconds	inherited from: GeneratingUnit
controlPulseLow	Seconds	inherited from: GeneratingUnit
controlResponseRate	ActivePowerChangeRate	inherited from: GeneratingUnit
dispReserveFlag	Boolean	inherited from: GeneratingUnit
efficiency	PU	inherited from: GeneratingUnit
energyMinP	HeatRate	inherited from: GeneratingUnit
fastStartFlag	Boolean	inherited from: GeneratingUnit
fuelPriority	Integer	inherited from: GeneratingUnit
genControlMode	GeneratorControlMode	inherited from: GeneratingUnit
genControlSource	GeneratorControlSource	inherited from: GeneratingUnit
genOperatingMode	GeneratorOperatingMode	inherited from: GeneratingUnit
governorMPL	PU	inherited from: GeneratingUnit
governorSCD	PerCent	inherited from: GeneratingUnit
highControlLimit	ActivePower	inherited from: GeneratingUnit
initialP	ActivePower	inherited from: GeneratingUnit
longPF	Float	inherited from: GeneratingUnit
lowControlLimit	ActivePower	inherited from: GeneratingUnit
lowerRampRate	ActivePowerChangeRate	inherited from: GeneratingUnit
maxEconomicP	ActivePower	inherited from: GeneratingUnit
maximumAllowableSpinningReserve	ActivePower	inherited from: GeneratingUnit
maxOperatingP	ActivePower	inherited from: GeneratingUnit
minEconomicP	ActivePower	inherited from: GeneratingUnit
minimumOffTime	Seconds	inherited from: GeneratingUnit
minOperatingP	ActivePower	inherited from: GeneratingUnit
modelDetail	Classification	inherited from: GeneratingUnit
nominalP	ActivePower	inherited from: GeneratingUnit
normalPF	Float	inherited from: GeneratingUnit
penaltyFactor	Float	inherited from: GeneratingUnit
raiseRampRate	ActivePowerChangeRate	inherited from: GeneratingUnit
ratedGrossMaxP	ActivePower	inherited from: GeneratingUnit
ratedGrossMinP	ActivePower	inherited from: GeneratingUnit

name	type	description
ratedNetMaxP	ActivePower	inherited from: GeneratingUnit
shortPF	Float	inherited from: GeneratingUnit
spinReserveRamp	ActivePowerChangeRate	inherited from: GeneratingUnit
startupCost	Money	inherited from: GeneratingUnit
startupTime	Seconds	inherited from: GeneratingUnit
stepChange	ActivePower	inherited from: GeneratingUnit
tieLinePF	Float	inherited from: GeneratingUnit
variableCost	Money	inherited from: GeneratingUnit
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 387 shows all association ends of HydroGeneratingUnit with other classes.

Table 387 – Association ends of Production::HydroGeneratingUnit with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] HydroGeneratingEfficiencyCurves	HydroGeneratingEfficiencyCurve	A hydro generating unit has an efficiency curve.
[1..1]	[0..*] TailbayLossCurve	TailbayLossCurve	A hydro generating unit has a tailbay loss curve.
[1..*]	[1..1] HydroPowerPlant	HydroPowerPlant	The hydro generating unit belongs to a hydro power plant.
[1..1]	[0..1] PenstockLossCurve	PenstockLossCurve	A hydro generating unit has a penstock loss curve.
[0..1]	[1..*] SynchronousMachines	SynchronousMachine	inherited from: GeneratingUnit
[1..1]	[0..1] GenUnitOpSchedule	GenUnitOpSchedule	inherited from: GeneratingUnit
[1..1]	[0..*] GrossToNetActivePowerCurves	GrossToNetActivePowerCurve	inherited from: GeneratingUnit
[1..1]	[0..*] GenUnitOpCostCurves	GenUnitOpCostCurve	inherited from: GeneratingUnit
[1..1]	[0..*] ControlAreaGeneratingUnit	ControlAreaGeneratingUnit	inherited from: GeneratingUnit
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.29 HydroPlantType enumeration

The type of hydro power plant.

Table 388 shows all literals of HydroPlantType.

Table 388 – Literals of Production::HydroPlantType

literal	description
runOfRiver	Run of river.
pumpedStorage	Pumped storage.
majorStorage	Major storage.
minorStorage	Minor storage.

6.9.3.30 HydroPowerPlant

A hydro power station which can generate or pump. When generating, the generator turbines receive there water from an upper reservoir. When pumping, the pumps receive their water from a lower reservoir.

Table 389 shows all attributes of HydroPowerPlant.

Table 389 – Attributes of Production::HydroPowerPlant

name	type	description
dischargeTravelDelay	Seconds	Water travel delay from tailbay to next downstream hydro power station.
genRatedP	ActivePower	The hydro plant's generating rating active power for rated head conditions.
hydroPlantType	HydroPlantType	The type of hydro power plant.
penstockType	PenstockType	Type and configuration of hydro plant penstock(s).
plantDischargeCapacity	VolumeFlowRate	Total plant discharge capacity.
plantRatedHead	Length	The plant's rated gross head.
pumpRatedP	ActivePower	The hydro plant's pumping rating active power for rated head conditions.
surgeTankCode	SurgeTankCode	A code describing the type (or absence) of surge tank that is associated with the hydro power plant.
surgeTankCrestLevel	WaterLevel	The level at which the surge tank spills.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 390 shows all association ends of HydroPowerPlant with other classes.

Table 390 – Association ends of Production::HydroPowerPlant with other classes

[mult from]	[mult to] name	type	description
[1..1]	[1..*] HydroGeneratingUnits	HydroGeneratingUnit	The hydro generating unit belongs to a hydro power plant.
[0..1]	[1..*] HydroPumps	HydroPump	The hydro pump may be a member of a pumped storage plant or a pump for distributing water.
[0..*]	[0..1] Reservoir	Reservoir	Generators discharge water to or pumps are supplied water from a downstream reservoir.
[0..*]	[1..1] GenSourcePumpDischargeReservoir	Reservoir	Generators are supplied water from or pumps discharge water to an upstream reservoir.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.31 HydroPump

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

Table 391 shows all attributes of HydroPump.

Table 391 – Attributes of Production::HydroPump

name	type	description
pumpDischAtMaxHead	VolumeFlowRate	The pumping discharge under maximum head conditions, usually at full gate.
pumpDischAtMinHead	VolumeFlowRate	The pumping discharge under minimum head conditions, usually at full gate.
pumpPowerAtMaxHead	ActivePower	The pumping power under maximum head conditions, usually at full gate.
pumpPowerAtMinHead	ActivePower	The pumping power under minimum head conditions, usually at full gate.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 392 shows all association ends of HydroPump with other classes.

Table 392 – Association ends of Production::HydroPump with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] SynchronousMachine	SynchronousMachine	The synchronous machine drives the turbine which moves the water from a low elevation to a higher elevation. The direction of machine rotation for pumping may or may not be the same as for generating.
[1..*]	[0..1] HydroPowerPlant	HydroPowerPlant	The hydro pump may be a member of a pumped storage plant or a pump for distributing water.
[1..1]	[0..1] HydroPumpOpSchedule	HydroPumpOpSchedule	The hydro pump has a pumping schedule over time, indicating when pumping is to occur.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.32 HydroPumpOpSchedule

The hydro pump's Operator-approved current operating schedule (or plan), typically produced with the aid of unit commitment type analyses. The unit's operating schedule status is typically given as: (0=unavailable) (1=available to startup or shutdown) (2=must pump).

Table 393 shows all attributes of HydroPumpOpSchedule.

Table 393 – Attributes of Production::HydroPumpOpSchedule

name	type	description
endTime	DateTime	inherited from: RegularIntervalSchedule
timeStep	Seconds	inherited from: RegularIntervalSchedule
startTime	DateTime	inherited from: BasicIntervalSchedule
value1Unit	UnitSymbol	inherited from: BasicIntervalSchedule
value1Multiplier	UnitMultiplier	inherited from: BasicIntervalSchedule
value2Unit	UnitSymbol	inherited from: BasicIntervalSchedule
value2Multiplier	UnitMultiplier	inherited from: BasicIntervalSchedule
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 394 shows all association ends of HydroPumpOpSchedule with other classes.

Table 394 – Association ends of Production::HydroPumpOpSchedule with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] HydroPump	HydroPump	The hydro pump has a pumping schedule over time, indicating when pumping is to occur.
[1..1]	[1..*] TimePoints	RegularTimePoint	inherited from: RegularIntervalSchedule
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.33 IncrementalHeatRateCurve

Relationship between unit incremental heat rate in (delta energy/time) per (delta active power) and unit output in active power. The IHR curve represents the slope of the HeatInputCurve. Note that the "incremental heat rate" and the "heat rate" have the same engineering units.

Table 395 shows all attributes of IncrementalHeatRateCurve.

Table 395 – Attributes of Production::IncrementalHeatRateCurve

name	type	description
isNetGrossP	Boolean	Flag is set to true when output is expressed in net active power.
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 396 shows all association ends of IncrementalHeatRateCurve with other classes.

Table 396 – Association ends of Production::IncrementalHeatRateCurve with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] ThermalGeneratingUnit	ThermalGeneratingUnit	A thermal generating unit may have an incremental heat rate curve.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.34 InflowForecast

Natural water inflow to a reservoir, usually forecasted from predicted rain and snowmelt. Typically in one hour increments for up to 10 days. The forecast is given in average cubic metres per second over the time increment.

Table 397 shows all attributes of InflowForecast.

Table 397 – Attributes of Production::InflowForecast

name	type	description
endTime	DateTime	inherited from: RegularIntervalSchedule
timeStep	Seconds	inherited from: RegularIntervalSchedule
startTime	DateTime	inherited from: BasicIntervalSchedule
value1Unit	UnitSymbol	inherited from: BasicIntervalSchedule
value1Multiplier	UnitMultiplier	inherited from: BasicIntervalSchedule
value2Unit	UnitSymbol	inherited from: BasicIntervalSchedule
value2Multiplier	UnitMultiplier	inherited from: BasicIntervalSchedule
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 398 shows all association ends of InflowForecast with other classes.

Table 398 – Association ends of Production::InflowForecast with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Reservoir	Reservoir	A reservoir may have a "natural" inflow forecast.
[1..1]	[1..*] TimePoints	RegularTimePoint	inherited from: RegularIntervalSchedule
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.35 LevelVsVolumeCurve

Relationship between reservoir volume and reservoir level. The volume is at the Y-axis and the reservoir level at the X-axis.

Table 399 shows all attributes of LevelVsVolumeCurve.

Table 399 – Attributes of Production::LevelVsVolumeCurve

name	type	description
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 400 shows all association ends of LevelVsVolumeCurve with other classes.

Table 400 – Association ends of Production::LevelVsVolumeCurve with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Reservoir	Reservoir	A reservoir may have a level versus volume relationship.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.36 NuclearGeneratingUnit

A nuclear generating unit.

Table 401 shows all attributes of NuclearGeneratingUnit.

Table 401 – Attributes of Production::NuclearGeneratingUnit

name	type	description
allocSpinResP	ActivePower	inherited from: GeneratingUnit
autoCntrlMarginP	ActivePower	inherited from: GeneratingUnit
baseP	ActivePower	inherited from: GeneratingUnit
controlDeadband	ActivePower	inherited from: GeneratingUnit
controlPulseHigh	Seconds	inherited from: GeneratingUnit
controlPulseLow	Seconds	inherited from: GeneratingUnit
controlResponseRate	ActivePowerChangeRate	inherited from: GeneratingUnit
dispReserveFlag	Boolean	inherited from: GeneratingUnit
efficiency	PU	inherited from: GeneratingUnit
energyMinP	HeatRate	inherited from: GeneratingUnit
fastStartFlag	Boolean	inherited from: GeneratingUnit
fuelPriority	Integer	inherited from: GeneratingUnit

name	type	description
genControlMode	GeneratorControlMode	inherited from: GeneratingUnit
genControlSource	GeneratorControlSource	inherited from: GeneratingUnit
genOperatingMode	GeneratorOperatingMode	inherited from: GeneratingUnit
governorMPL	PU	inherited from: GeneratingUnit
governorSCD	PerCent	inherited from: GeneratingUnit
highControlLimit	ActivePower	inherited from: GeneratingUnit
initialP	ActivePower	inherited from: GeneratingUnit
longPF	Float	inherited from: GeneratingUnit
lowControlLimit	ActivePower	inherited from: GeneratingUnit
lowerRampRate	ActivePowerChangeRate	inherited from: GeneratingUnit
maxEconomicP	ActivePower	inherited from: GeneratingUnit
maximumAllowableSpinningReserve	ActivePower	inherited from: GeneratingUnit
maxOperatingP	ActivePower	inherited from: GeneratingUnit
minEconomicP	ActivePower	inherited from: GeneratingUnit
minimumOffTime	Seconds	inherited from: GeneratingUnit
minOperatingP	ActivePower	inherited from: GeneratingUnit
modelDetail	Classification	inherited from: GeneratingUnit
nominalP	ActivePower	inherited from: GeneratingUnit
normalPF	Float	inherited from: GeneratingUnit
penaltyFactor	Float	inherited from: GeneratingUnit
raiseRampRate	ActivePowerChangeRate	inherited from: GeneratingUnit
ratedGrossMaxP	ActivePower	inherited from: GeneratingUnit
ratedGrossMinP	ActivePower	inherited from: GeneratingUnit
ratedNetMaxP	ActivePower	inherited from: GeneratingUnit
shortPF	Float	inherited from: GeneratingUnit
spinReserveRamp	ActivePowerChangeRate	inherited from: GeneratingUnit
startupCost	Money	inherited from: GeneratingUnit
startupTime	Seconds	inherited from: GeneratingUnit
stepChange	ActivePower	inherited from: GeneratingUnit
tieLinePF	Float	inherited from: GeneratingUnit
variableCost	Money	inherited from: GeneratingUnit
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 402 shows all association ends of NuclearGeneratingUnit with other classes.

Table 402 – Association ends of Production::NuclearGeneratingUnit with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..*] SynchronousMachines	SynchronousMachine	inherited from: GeneratingUnit
[1..1]	[0..1] GenUnitOpSchedule	GenUnitOpSchedule	inherited from: GeneratingUnit
[1..1]	[0..*] GrossToNetActivePowerCurves	GrossToNetActivePowerCurve	inherited from: GeneratingUnit
[1..1]	[0..*] GenUnitOpCostCurves	GenUnitOpCostCurve	inherited from: GeneratingUnit
[1..1]	[0..*] ControlAreaGeneratingUnit	ControlAreaGeneratingUnit	inherited from: GeneratingUnit
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.37 PenstockLossCurve

Relationship between penstock head loss (in metres) and total discharge through the penstock (in cubic metres per second). One or more turbines may be connected to the same penstock.

Table 403 shows all attributes of PenstockLossCurve.

Table 403 – Attributes of Production::PenstockLossCurve

name	type	description
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 404 shows all association ends of PenstockLossCurve with other classes.

Table 404 – Association ends of Production::PenstockLossCurve with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] HydroGeneratingUnit	HydroGeneratingUnit	A hydro generating unit has a penstock loss curve.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.38 PenstockType enumeration

Type of hydro plant penstock.

Table 405 shows all literals of PenstockType.

Table 405 – Literals of Production::PenstockType

literal	description
undefined	The penstock type is undefined.

6.9.3.39 Reservoir

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

Table 406 shows all attributes of Reservoir.

Table 406 – Attributes of Production::Reservoir

name	type	description
activeStorageCapacity	Volume	Storage volume between the full supply level and the normal minimum operating level.
energyStorageRating	Float	The reservoir's energy storage rating in energy for given head conditions.
fullSupplyLevel	WaterLevel	Full supply level, above which water will spill. This can be the spillway crest level or the top of closed gates.
grossCapacity	Volume	Total capacity of reservoir.
normalMinOperateLevel	WaterLevel	Normal minimum operating level below which the penstocks will draw air.
riverOutletWorks	String	River outlet works for riparian right releases or other purposes.
spillTravelDelay	Seconds	The spillway water travel delay to the next downstream reservoir.
spillwayCapacity	Float	The flow capacity of the spillway in cubic metres per second.
spillwayCrestLength	Length	The length of the spillway crest.
spillwayCrestLevel	WaterLevel	Spillway crest level above which water will spill.
spillWayGateType	SpillwayGateType	Type of spillway gate, including parameters.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 407 shows all association ends of Reservoir with other classes.

Table 407 – Association ends of Production::Reservoir with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] SpillsIntoReservoirs	Reservoir	A reservoir may spill into a downstream reservoir.
[0..1]	[0..*] HydroPowerPlants	HydroPowerPlant	Generators discharge water to or pumps are supplied water from a downstream reservoir.
[1..1]	[0..*] UpstreamFromHydroPowerPlants	HydroPowerPlant	Generators are supplied water from or pumps discharge water to an upstream reservoir.
[1..1]	[0..*] InflowForecasts	InflowForecast	A reservoir may have a "natural" inflow forecast.
[1..1]	[0..*] LevelVsVolumeCurves	LevelVsVolumeCurve	A reservoir may have a level versus volume relationship.
[1..1]	[0..1] TargetLevelSchedule	TargetLevelSchedule	A reservoir may have a water level target schedule.
[0..*]	[0..1] SpillsFromReservoir	Reservoir	A reservoir may spill into a downstream reservoir.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.40 ShutdownCurve

Relationship between the rate in gross active power/minute (Y-axis) at which a unit should be shutdown and its present gross MW output (X-axis).

Table 408 shows all attributes of ShutdownCurve.

Table 408 – Attributes of Production::ShutdownCurve

name	type	description
shutdownCost	Money	Fixed shutdown cost.
shutdownDate	DateTime	The date and time of the most recent generating unit shutdown.
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 409 shows all association ends of ShutdownCurve with other classes.

Table 409 – Association ends of Production::ShutdownCurve with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] ThermalGeneratingUnit	ThermalGeneratingUnit	A thermal generating unit may have a shutdown curve.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.41 SpillwayGateType enumeration

Type of spillway gate.

Table 410 shows all literals of SpillwayGateType.

Table 410 – Literals of Production::SpillwayGateType

literal	description
undefined	The type is spillway gate type is undefined.

6.9.3.42 StartIgnFuelCurve

The quantity of ignition fuel (Y-axis) used to restart and repay the auxiliary power consumed versus the number of hours (X-axis) the unit was off line.

Table 411 shows all attributes of StartIgnFuelCurve.

Table 411 – Attributes of Production::StartIgnFuelCurve

name	type	description
ignitionFuelType	FuelType	Type of ignition fuel.
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 412 shows all association ends of StartIgnFuelCurve with other classes.

Table 412 – Association ends of Production::StartIgnFuelCurve with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] StartupModel	StartupModel	The unit's startup model may have a startup ignition fuel curve.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.43 StartMainFuelCurve

The quantity of main fuel (Y-axis) used to restart and repay the auxiliary power consumed versus the number of hours (X-axis) the unit was off line.

Table 413 shows all attributes of StartMainFuelCurve.

Table 413 – Attributes of Production::StartMainFuelCurve

name	type	description
mainFuelType	FuelType	Type of main fuel.
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 414 shows all association ends of StartMainFuelCurve with other classes.

Table 414 – Association ends of Production::StartMainFuelCurve with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] StartupModel	StartupModel	The unit's startup model may have a startup main fuel curve.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.44 StartRampCurve

Rate in gross active power/minute (Y-axis) at which a unit can be loaded versus the number of hours (X-axis) the unit was off line.

Table 415 shows all attributes of StartRampCurve.

Table 415 – Attributes of Production::StartRampCurve

name	type	description
hotStandbyRamp	ActivePowerChangeRate	The startup ramp rate in gross for a unit that is on hot standby.
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 416 shows all association ends of StartRampCurve with other classes.

Table 416 – Association ends of Production::StartRampCurve with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] StartupModel	StartupModel	The unit's startup model may have a startup ramp curve.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.45 StartupModel

Unit start up characteristics depending on how long the unit has been off line.

Table 417 shows all attributes of StartupModel.

Table 417 – Attributes of Production::StartupModel

name	type	description
fixedMaintCost	CostRate	Fixed maintenance cost.
hotStandbyHeat	HeatRate	The amount of heat input per time unit required for hot standby operation.
incrementalMaintCost	CostPerEnergyUnit	Incremental maintenance cost.
minimumDownTime	Hours	The minimum number of hours the unit must be down before restart.
minimumRunTime	Hours	The minimum number of hours the unit must be operating before being allowed to shut down.
riskFactorCost	Money	The opportunity cost associated with the return in monetary unit. This represents the restart's "share" of the unit depreciation and risk of an event which would damage the unit.
startupCost	Money	Total miscellaneous start up costs.

name	type	description
startupDate	DateTime	The date and time of the most recent generating unit startup.
startupPriority	Integer	Startup priority within control area where lower numbers indicate higher priorities. More than one unit in an area may be assigned the same priority.
stbyAuxP	ActivePower	The unit's auxiliary active power consumption to maintain standby mode.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 418 shows all association ends of StartupModel with other classes.

Table 418 – Association ends of Production::StartupModel with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] ThermalGeneratingUnit	ThermalGeneratingUnit	A thermal generating unit may have a startup model.
[1..1]	[0..1] StartIgnFuelCurve	StartIgnFuelCurve	The unit's startup model may have a startup ignition fuel curve.
[1..1]	[0..1] StartMainFuelCurve	StartMainFuelCurve	The unit's startup model may have a startup main fuel curve.
[1..1]	[0..1] StartRampCurve	StartRampCurve	The unit's startup model may have a startup ramp curve.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.46 SteamSendoutSchedule

The cogeneration plant's steam sendout schedule in volume per time unit.

Table 419 shows all attributes of SteamSendoutSchedule.

Table 419 – Attributes of Production::SteamSendoutSchedule

name	type	description
endTime	DateTime	inherited from: RegularIntervalSchedule
timeStep	Seconds	inherited from: RegularIntervalSchedule
startTime	DateTime	inherited from: BasicIntervalSchedule
value1Unit	UnitSymbol	inherited from: BasicIntervalSchedule
value1Multiplier	UnitMultiplier	inherited from: BasicIntervalSchedule
value2Unit	UnitSymbol	inherited from: BasicIntervalSchedule
value2Multiplier	UnitMultiplier	inherited from: BasicIntervalSchedule
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 420 shows all association ends of SteamSendoutSchedule with other classes.

Table 420 – Association ends of Production::SteamSendoutSchedule with other classes

[mult from]	[mult to] name	type	description
[1..1]	[1..1] CogenerationPlant	CogenerationPlant	A cogeneration plant has a steam sendout schedule.
[1..1]	[1..*] TimePoints	RegularTimePoint	inherited from: RegularIntervalSchedule
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.47 SurgeTankCode enumeration

Type (or absence) of surge tank that is associated with the hydro power plant.

Table 421 shows all literals of SurgeTankCode.

Table 421 – Literals of Production::SurgeTankCode

literal	description
undefined	The surge tank code is undefined.

6.9.3.48 TailbayLossCurve

Relationship between tailbay head loss hight (Y-axis) and the total discharge into the power station's tailbay volume per time unit (X-axis) . There could be more than one curve depending on the level of the tailbay reservoir or river level.

Table 422 shows all attributes of TailbayLossCurve.

Table 422 – Attributes of Production::TailbayLossCurve

name	type	description
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 423 shows all association ends of TailbayLossCurve with other classes.

Table 423 – Association ends of Production::TailbayLossCurve with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] HydroGeneratingUnit	HydroGeneratingUnit	A hydro generating unit has a tailbay loss curve.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.49 TargetLevelSchedule

Reservoir water level targets from advanced studies or "rule curves". Typically in one hour increments for up to 10 days.

Table 424 shows all attributes of TargetLevelSchedule.

Table 424 – Attributes of Production::TargetLevelSchedule

name	type	description
highLevelLimit	WaterLevel	High target level limit, above which the reservoir operation will be penalized.
lowLevelLimit	WaterLevel	Low target level limit, below which the reservoir operation will be penalized.
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 425 shows all association ends of TargetLevelSchedule with other classes.

Table 425 – Association ends of Production::TargetLevelSchedule with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] Reservoir	Reservoir	A reservoir may have a water level target schedule.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.50 ThermalGeneratingUnit

A generating unit whose prime mover could be a steam turbine, combustion turbine, or diesel engine.

Table 426 shows all attributes of ThermalGeneratingUnit.

Table 426 – Attributes of Production::ThermalGeneratingUnit

name	type	description
oMCost	CostPerHeatUnit	Operating and maintenance cost for the thermal unit.
allocSpinResP	ActivePower	inherited from: GeneratingUnit
autoCntrlMarginP	ActivePower	inherited from: GeneratingUnit
baseP	ActivePower	inherited from: GeneratingUnit
controlDeadband	ActivePower	inherited from: GeneratingUnit
controlPulseHigh	Seconds	inherited from: GeneratingUnit
controlPulseLow	Seconds	inherited from: GeneratingUnit
controlResponseRate	ActivePowerChangeRate	inherited from: GeneratingUnit
dispReserveFlag	Boolean	inherited from: GeneratingUnit
efficiency	PU	inherited from: GeneratingUnit
energyMinP	HeatRate	inherited from: GeneratingUnit
fastStartFlag	Boolean	inherited from: GeneratingUnit
fuelPriority	Integer	inherited from: GeneratingUnit
genControlMode	GeneratorControlMode	inherited from: GeneratingUnit
genControlSource	GeneratorControlSource	inherited from: GeneratingUnit
genOperatingMode	GeneratorOperatingMode	inherited from: GeneratingUnit
governorMPL	PU	inherited from: GeneratingUnit
governorSCD	PerCent	inherited from: GeneratingUnit
highControlLimit	ActivePower	inherited from: GeneratingUnit
initialIP	ActivePower	inherited from: GeneratingUnit
longPF	Float	inherited from: GeneratingUnit
lowControlLimit	ActivePower	inherited from: GeneratingUnit
lowerRampRate	ActivePowerChangeRate	inherited from: GeneratingUnit
maxEconomicP	ActivePower	inherited from: GeneratingUnit
maximumAllowableSpinningReserve	ActivePower	inherited from: GeneratingUnit
maxOperatingP	ActivePower	inherited from: GeneratingUnit
minEconomicP	ActivePower	inherited from: GeneratingUnit
minimumOffTime	Seconds	inherited from: GeneratingUnit
minOperatingP	ActivePower	inherited from: GeneratingUnit
modelDetail	Classification	inherited from: GeneratingUnit
nominalP	ActivePower	inherited from: GeneratingUnit
normalPF	Float	inherited from: GeneratingUnit
penaltyFactor	Float	inherited from: GeneratingUnit
raiseRampRate	ActivePowerChangeRate	inherited from: GeneratingUnit
ratedGrossMaxP	ActivePower	inherited from: GeneratingUnit
ratedGrossMinP	ActivePower	inherited from: GeneratingUnit

name	type	description
ratedNetMaxP	ActivePower	inherited from: GeneratingUnit
shortPF	Float	inherited from: GeneratingUnit
spinReserveRamp	ActivePowerChangeRate	inherited from: GeneratingUnit
startupCost	Money	inherited from: GeneratingUnit
startupTime	Seconds	inherited from: GeneratingUnit
stepChange	ActivePower	inherited from: GeneratingUnit
tieLinePF	Float	inherited from: GeneratingUnit
variableCost	Money	inherited from: GeneratingUnit
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 427 shows all association ends of ThermalGeneratingUnit with other classes.

Table 427 – Association ends of Production::ThermalGeneratingUnit with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..1] CAESPlant	CAESPlant	A thermal generating unit may be a member of a compressed air energy storage plant.
[0..*]	[0..1] CogenerationPlant	CogenerationPlant	A thermal generating unit may be a member of a cogeneration plant.
[0..*]	[0..1] CombinedCyclePlant	CombinedCyclePlant	A thermal generating unit may be a member of a combined cycle plant.
[1..1]	[0..*] EmissionAccounts	EmissionAccount	A thermal generating unit may have one or more emission allowance accounts.
[1..1]	[0..*] EmissionCurves	EmissionCurve	A thermal generating unit may have one or more emission curves.
[1..1]	[0..*] FossilFuels	FossilFuel	A thermal generating unit may have one or more fossil fuels.
[1..1]	[0..*] FuelAllocationSchedules	FuelAllocationSchedule	A thermal generating unit may have one or more fuel allocation schedules.
[1..1]	[0..1] HeatInputCurve	HeatInputCurve	A thermal generating unit may have a heat input curve.
[1..1]	[0..1] HeatRateCurve	HeatRateCurve	A thermal generating unit may have a heat rate curve.
[1..1]	[0..1] IncrementalHeatRateCurve	IncrementalHeatRateCurve	A thermal generating unit may have an incremental heat rate curve.
[1..1]	[0..1] ShutdownCurve	ShutdownCurve	A thermal generating unit may have a shutdown curve.
[1..1]	[0..1] StartupModel	StartupModel	A thermal generating unit may have a startup model.
[0..1]	[1..*] SynchronousMachines	SynchronousMachine	inherited from: GeneratingUnit
[1..1]	[0..1] GenUnitOpSchedule	GenUnitOpSchedule	inherited from: GeneratingUnit
[1..1]	[0..*] GrossToNetActivePowerCurves	GrossToNetActivePowerCurve	inherited from: GeneratingUnit

[mult from]	[mult to] name	type	description
[1..1]	[0..*] GenUnitOpCostCurves	GenUnitOpCostCurve	inherited from: GeneratingUnit
[1..1]	[0..*] ControlAreaGeneratingUnit	ControlAreaGeneratingUnit	inherited from: GeneratingUnit
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.51 WindGeneratingUnit

A wind driven generating unit.

Table 428 shows all attributes of WindGeneratingUnit.

Table 428 – Attributes of Production::WindGeneratingUnit

name	type	description
allocSpinResP	ActivePower	inherited from: GeneratingUnit
autoCntrlMarginP	ActivePower	inherited from: GeneratingUnit
baseP	ActivePower	inherited from: GeneratingUnit
controlDeadband	ActivePower	inherited from: GeneratingUnit
controlPulseHigh	Seconds	inherited from: GeneratingUnit
controlPulseLow	Seconds	inherited from: GeneratingUnit
controlResponseRate	ActivePowerChangeRate	inherited from: GeneratingUnit
dispReserveFlag	Boolean	inherited from: GeneratingUnit
efficiency	PU	inherited from: GeneratingUnit
energyMinP	HeatRate	inherited from: GeneratingUnit
fastStartFlag	Boolean	inherited from: GeneratingUnit
fuelPriority	Integer	inherited from: GeneratingUnit
genControlMode	GeneratorControlMode	inherited from: GeneratingUnit
genControlSource	GeneratorControlSource	inherited from: GeneratingUnit
genOperatingMode	GeneratorOperatingMode	inherited from: GeneratingUnit
governorMPL	PU	inherited from: GeneratingUnit
governorSCD	PerCent	inherited from: GeneratingUnit
highControlLimit	ActivePower	inherited from: GeneratingUnit
initialP	ActivePower	inherited from: GeneratingUnit
longPF	Float	inherited from: GeneratingUnit
lowControlLimit	ActivePower	inherited from: GeneratingUnit

name	type	description
lowerRampRate	ActivePowerChangeRate	inherited from: GeneratingUnit
maxEconomicP	ActivePower	inherited from: GeneratingUnit
maximumAllowableSpinningReserve	ActivePower	inherited from: GeneratingUnit
maxOperatingP	ActivePower	inherited from: GeneratingUnit
minEconomicP	ActivePower	inherited from: GeneratingUnit
minimumOffTime	Seconds	inherited from: GeneratingUnit
minOperatingP	ActivePower	inherited from: GeneratingUnit
modelDetail	Classification	inherited from: GeneratingUnit
nominalP	ActivePower	inherited from: GeneratingUnit
normalPF	Float	inherited from: GeneratingUnit
penaltyFactor	Float	inherited from: GeneratingUnit
raiseRampRate	ActivePowerChangeRate	inherited from: GeneratingUnit
ratedGrossMaxP	ActivePower	inherited from: GeneratingUnit
ratedGrossMinP	ActivePower	inherited from: GeneratingUnit
ratedNetMaxP	ActivePower	inherited from: GeneratingUnit
shortPF	Float	inherited from: GeneratingUnit
spinReserveRamp	ActivePowerChangeRate	inherited from: GeneratingUnit
startupCost	Money	inherited from: GeneratingUnit
startupTime	Seconds	inherited from: GeneratingUnit
stepChange	ActivePower	inherited from: GeneratingUnit
tieLinePF	Float	inherited from: GeneratingUnit
variableCost	Money	inherited from: GeneratingUnit
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 429 shows all association ends of WindGeneratingUnit with other classes.

Table 429 – Association ends of Production::WindGeneratingUnit with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..*] SynchronousMachines	SynchronousMachine	inherited from: GeneratingUnit
[1..1]	[0..1] GenUnitOpSchedule	GenUnitOpSchedule	inherited from: GeneratingUnit
[1..1]	[0..*] GrossToNetActivePowerCurves	GrossToNetActivePowerCurve	inherited from: GeneratingUnit
[1..1]	[0..*] GenUnitOpCostCurves	GenUnitOpCostCurve	inherited from: GeneratingUnit
[1..1]	[0..*] ControlAreaGeneratingUnit	ControlAreaGeneratingUnit	inherited from: GeneratingUnit
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject

This diagram shows all classes included in the LoadModel package as well as the key external classes that have associations with LoadModel classes.

Figure 69 shows class diagram Datatypes.



Figure 69 – Class diagram LoadModel::Datatypes

Shows datatypes for load model.

6.10.2 ConformLoad

ConformLoad represent loads that follow a daily load change pattern where the pattern can be used to scale the load with a system load.

Table 430 shows all attributes of ConformLoad.

Table 430 – Attributes of LoadModel::ConformLoad

name	type	description
customerCount	Integer	inherited from: EnergyConsumer
grounded	WindingConnection	inherited from: EnergyConsumer
pfixed	ActivePower	inherited from: EnergyConsumer
pfixedPct	PerCent	inherited from: EnergyConsumer
phaseConnection	PhaseShuntConnectionKind	inherited from: EnergyConsumer
qfixed	ReactivePower	inherited from: EnergyConsumer
qfixedPct	PerCent	inherited from: EnergyConsumer
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 431 shows all association ends of ConformLoad with other classes.

Table 431 – Association ends of LoadModel::ConformLoad with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] LoadGroup	ConformLoadGroup	Group of this ConformLoad.
[0..*]	[0..1] LoadResponse	LoadResponseCharacteristic	inherited from: EnergyConsumer
[1..1]	[0..*] EnergyConsumerPhase	EnergyConsumerPhase	inherited from: EnergyConsumer
[1..*]	[0..1] PowerCutZone	PowerCutZone	inherited from: EnergyConsumer
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.10.3 ConformLoadGroup

A group of loads conforming to an allocation pattern.

Table 432 shows all attributes of ConformLoadGroup.

Table 432 – Attributes of LoadModel::ConformLoadGroup

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 433 shows all association ends of ConformLoadGroup with other classes.

Table 433 – Association ends of LoadModel::ConformLoadGroup with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] EnergyConsumers	ConformLoad	Conform loads assigned to this ConformLoadGroup.
[1..1]	[1..*] ConformLoadSchedules	ConformLoadSchedule	The ConformLoadSchedules in the ConformLoadGroup.
[1..*]	[1..1] SubLoadArea	SubLoadArea	inherited from: LoadGroup
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.10.4 ConformLoadSchedule

A curve of load versus time (X-axis) showing the active power values (Y1-axis) and reactive power (Y2-axis) for each unit of the period covered. This curve represents a typical pattern of load over the time period for a given day type and season.

Table 434 shows all attributes of ConformLoadSchedule.

Table 434 – Attributes of LoadModel::ConformLoadSchedule

name	type	description
endTime	DateTime	inherited from: RegularIntervalSchedule
timeStep	Seconds	inherited from: RegularIntervalSchedule
startTime	DateTime	inherited from: BasicIntervalSchedule
value1Unit	UnitSymbol	inherited from: BasicIntervalSchedule
value1Multiplier	UnitMultiplier	inherited from: BasicIntervalSchedule
value2Unit	UnitSymbol	inherited from: BasicIntervalSchedule
value2Multiplier	UnitMultiplier	inherited from: BasicIntervalSchedule
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 435 shows all association ends of ConformLoadSchedule with other classes.

Table 435 – Association ends of LoadModel::ConformLoadSchedule with other classes

[mult from]	[mult to] name	type	description
[1..*]	[1..1] ConformLoadGroup	ConformLoadGroup	The ConformLoadGroup where the ConformLoadSchedule belongs.
[0..*]	[0..1] DayType	DayType	inherited from: SeasonDayTypeSchedule
[0..*]	[0..1] Season	Season	inherited from: SeasonDayTypeSchedule
[1..1]	[1..*] TimePoints	RegularTimePoint	inherited from: RegularIntervalSchedule
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.10.5 DayType

Group of similar days. For example it could be used to represent weekdays, weekend, or holidays.

Table 436 shows all attributes of DayType.

Table 436 – Attributes of LoadModel::DayType

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 437 shows all association ends of DayType with other classes.

Table 437 – Association ends of LoadModel::DayType with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] SeasonDayTypeSchedules	SeasonDayTypeSchedule	Schedules that use this DayType.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.10.6 EnergyArea

The class describes an area having energy production or consumption.

Table 438 shows all attributes of EnergyArea.

Table 438 – Attributes of LoadModel::EnergyArea

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 439 shows all association ends of EnergyArea with other classes.

Table 439 – Association ends of LoadModel::EnergyArea with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..1] ControlArea	ControlArea	The control area specification that is used for the load forecast.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.10.7 LoadArea

The class is the root or first level in a hierarchical structure for grouping of loads for the purpose of load flow load scaling.

Table 440 shows all attributes of LoadArea.

Table 440 – Attributes of LoadModel::LoadArea

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 441 shows all association ends of LoadArea with other classes.

Table 441 – Association ends of LoadModel::LoadArea with other classes

[mult from]	[mult to] name	type	description
[1..1]	[1..*] SubLoadAreas	SubLoadArea	The SubLoadAreas in the LoadArea.
[0..1]	[0..1] ControlArea	ControlArea	inherited from: EnergyArea
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.10.8 LoadGroup

The class is the third level in a hierarchical structure for grouping of loads for the purpose of load flow load scaling.

Table 442 shows all attributes of LoadGroup.

Table 442 – Attributes of LoadModel::LoadGroup

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 443 shows all association ends of LoadGroup with other classes.

Table 443 – Association ends of LoadModel::LoadGroup with other classes

[mult from]	[mult to] name	type	description
[1..*]	[1..1] SubLoadArea	SubLoadArea	The SubLoadArea where the Loadgroup belongs.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.10.9 LoadResponseCharacteristic

Models the characteristic response of the load demand due to changes in system conditions such as voltage and frequency. This is not related to demand response.

If LoadResponseCharacteristic.exponentModel is True, the voltage exponents are specified and used as to calculate:

Active power component = $P_{\text{nominal}} \times (\text{Voltage}/\text{cim:BaseVoltage.nominalVoltage})^{**\text{cim:LoadResponseCharacteristic.pVoltageExponent}}$

Reactive power component = $Q_{\text{nominal}} \times (\text{Voltage}/\text{cim:BaseVoltage.nominalVoltage})^{**\text{cim:LoadResponseCharacteristic.qVoltageExponent}}$

Where $\times$ means "multiply" and $**$ is "raised to power of".

Table 444 shows all attributes of LoadResponseCharacteristic.

Table 444 – Attributes of LoadModel::LoadResponseCharacteristic

name	type	description
exponentModel	Boolean	Indicates the exponential voltage dependency model (pVoltageExponent and qVoltageExponent) is to be used. If false, the coefficient model (consisting of pConstantImpedance, pConstantCurrent, pConstantPower, qConstantImpedance, qConstantCurrent, and qConstantPower) is to be used.
pConstantCurrent	Float	Portion of active power load modeled as constant current. Used only if the useExponentModel is false. This value is normalized against the sum of pZ, pI, and pP.
pConstantImpedance	Float	Portion of active power load modeled as constant impedance. Used only if the useExponentModel is false. This value is normalized against the sum of pZ, pI, and pP.
pConstantPower	Float	Portion of active power load modeled as constant power. Used only if the useExponentModel is false. This value is normalized against the sum of pZ, pI, and pP.
pFrequencyExponent	Float	Exponent of per unit frequency effecting active power.
pVoltageExponent	Float	Exponent of per unit voltage effecting real power. This model used only when "useExponentModel" is true.
qConstantCurrent	Float	Portion of reactive power load modeled as constant current. Used only if the useExponentModel is false. This value is normalized against the sum of qZ, qI, and qP.
qConstantImpedance	Float	Portion of reactive power load modeled as constant impedance. Used only if the useExponentModel is false. This value is normalized against the sum of qZ, qI, and qP.
qConstantPower	Float	Portion of reactive power load modeled as constant power. Used only if the useExponentModel is false. This value is normalized against the sum of qZ, qI, and qP.
qFrequencyExponent	Float	Exponent of per unit frequency effecting reactive power.
qVoltageExponent	Float	Exponent of per unit voltage effecting reactive power. This model used only when "useExponentModel" is true.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 445 shows all association ends of LoadResponseCharacteristic with other classes.

Table 445 – Association ends of LoadModel::LoadResponseCharacteristic with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] EnergyConsumer	EnergyConsumer	The set of loads that have the response characteristics.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.10.10 NonConformLoad

NonConformLoad represent loads that do not follow a daily load change pattern and changes are not correlated with the daily load change pattern.

Table 446 shows all attributes of NonConformLoad.

Table 446 – Attributes of LoadModel::NonConformLoad

name	type	description
customerCount	Integer	inherited from: EnergyConsumer
grounded	WindingConnection	inherited from: EnergyConsumer
pfixed	ActivePower	inherited from: EnergyConsumer
pfixedPct	PerCent	inherited from: EnergyConsumer
phaseConnection	PhaseShuntConnectionKind	inherited from: EnergyConsumer
qfixed	ReactivePower	inherited from: EnergyConsumer
qfixedPct	PerCent	inherited from: EnergyConsumer
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 447 shows all association ends of NonConformLoad with other classes.

Table 447 – Association ends of LoadModel::NonConformLoad with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] LoadGroup	NonConformLoadGroup	Group of this ConformLoad.
[0..*]	[0..1] LoadResponse	LoadResponseCharacteristic	inherited from: EnergyConsumer
[1..1]	[0..*] EnergyConsumerPhase	EnergyConsumerPhase	inherited from: EnergyConsumer
[1..*]	[0..1] PowerCutZone	PowerCutZone	inherited from: EnergyConsumer
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.10.11 NonConformLoadGroup

Loads that do not follow a daily and seasonal load variation pattern.

Table 448 shows all attributes of NonConformLoadGroup.

Table 448 – Attributes of LoadModel::NonConformLoadGroup

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 449 shows all association ends of NonConformLoadGroup with other classes.

Table 449 – Association ends of LoadModel::NonConformLoadGroup with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] EnergyConsumers	NonConformLoad	Conform loads assigned to this ConformLoadGroup.
[1..1]	[1..*] NonConformLoadSchedules	NonConformLoadSchedule	The NonConformLoadSchedules in the NonConformLoadGroup.
[1..*]	[1..1] SubLoadArea	SubLoadArea	inherited from: LoadGroup
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.10.12 NonConformLoadSchedule

An active power (Y1-axis) and reactive power (Y2-axis) schedule (curves) versus time (X-axis) for non-conforming loads, e.g., large industrial load or power station service (where modeled).

Table 450 shows all attributes of NonConformLoadSchedule.

Table 450 – Attributes of LoadModel::NonConformLoadSchedule

name	type	description
endTime	DateTime	inherited from: RegularIntervalSchedule
timeStep	Seconds	inherited from: RegularIntervalSchedule
startTime	DateTime	inherited from: BasicIntervalSchedule
value1Unit	UnitSymbol	inherited from: BasicIntervalSchedule
value1Multiplier	UnitMultiplier	inherited from: BasicIntervalSchedule
value2Unit	UnitSymbol	inherited from: BasicIntervalSchedule
value2Multiplier	UnitMultiplier	inherited from: BasicIntervalSchedule
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 451 shows all association ends of NonConformLoadSchedule with other classes.

Table 451 – Association ends of LoadModel::NonConformLoadSchedule with other classes

[mult from]	[mult to] name	type	description
[1..*]	[1..1] NonConformLoadGroup	NonConformLoadGroup	The NonConformLoadGroup where the NonConformLoadSchedule belongs.
[0..*]	[0..1] DayType	DayType	inherited from: SeasonDayTypeSchedule
[0..*]	[0..1] Season	Season	inherited from: SeasonDayTypeSchedule
[1..1]	[1..*] TimePoints	RegularTimePoint	inherited from: RegularIntervalSchedule
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.10.13 PowerCutZone

An area or zone of the power system which is used for load shedding purposes.

Table 452 shows all attributes of PowerCutZone.

Table 452 – Attributes of LoadModel::PowerCutZone

name	type	description
cutLevel1	PerCent	First level (amount) of load to cut as a percentage of total zone load.
cutLevel2	PerCent	Second level (amount) of load to cut as a percentage of total zone load.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 453 shows all association ends of PowerCutZone with other classes.

Table 453 – Association ends of LoadModel::PowerCutZone with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..*] EnergyConsumers	EnergyConsumer	Energy consumer is assigned to the power cut zone.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.10.14 Season root class

A specified time period of the year.

Table 454 shows all attributes of Season.

Table 454 – Attributes of LoadModel::Season

name	type	description
endDate	DateTime	Date season ends.
name	SeasonName	Name of the season.
startDate	DateTime	Date season starts.

Table 455 shows all association ends of Season with other classes.

Table 455 – Association ends of LoadModel::Season with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] SeasonDayTypeSchedules	SeasonDayTypeSchedule	Schedules that use this Season.

6.10.15 SeasonDayTypeSchedule

A time schedule covering a 24 hour period, with curve data for a specific type of season and day.

Table 456 shows all attributes of SeasonDayTypeSchedule.

Table 456 – Attributes of LoadModel::SeasonDayTypeSchedule

name	type	description
endTime	DateTime	inherited from: RegularIntervalSchedule
timeStep	Seconds	inherited from: RegularIntervalSchedule
startTime	DateTime	inherited from: BasicIntervalSchedule
value1Unit	UnitSymbol	inherited from: BasicIntervalSchedule
value1Multiplier	UnitMultiplier	inherited from: BasicIntervalSchedule
value2Unit	UnitSymbol	inherited from: BasicIntervalSchedule
value2Multiplier	UnitMultiplier	inherited from: BasicIntervalSchedule
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 457 shows all association ends of SeasonDayTypeSchedule with other classes.

Table 457 – Association ends of LoadModel::SeasonDayTypeSchedule with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] DayType	DayType	DayType for the Schedule.
[0..*]	[0..1] Season	Season	Season for the Schedule.
[1..1]	[1..*] TimePoints	RegularTimePoint	inherited from: RegularIntervalSchedule
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.10.16 SeasonName enumeration

Name of season.

Table 458 shows all literals of SeasonName.

Table 458 – Literals of LoadModel::SeasonName

literal	description
winter	Winter.
spring	Spring.
summer	Summer.
fall	Fall.

6.10.17 StationSupply

Station supply with load derived from the station output.

Table 459 shows all attributes of StationSupply.

Table 459 – Attributes of LoadModel::StationSupply

name	type	description
customerCount	Integer	inherited from: EnergyConsumer
grounded	WindingConnection	inherited from: EnergyConsumer
pfixed	ActivePower	inherited from: EnergyConsumer
pfixedPct	PerCent	inherited from: EnergyConsumer
phaseConnection	PhaseShuntConnectionKind	inherited from: EnergyConsumer
qfixed	ReactivePower	inherited from: EnergyConsumer
qfixedPct	PerCent	inherited from: EnergyConsumer
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 460 shows all association ends of StationSupply with other classes.

Table 460 – Association ends of LoadModel::StationSupply with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] LoadResponse	LoadResponseCharacteristic	inherited from: EnergyConsumer
[1..1]	[0..*] EnergyConsumerPhase	EnergyConsumerPhase	inherited from: EnergyConsumer
[1..*]	[0..1] PowerCutZone	PowerCutZone	inherited from: EnergyConsumer
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.10.18 SubLoadArea

The class is the second level in a hierarchical structure for grouping of loads for the purpose of load flow load scaling.

Table 461 shows all attributes of SubLoadArea.

Table 461 – Attributes of LoadModel::SubLoadArea

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 462 shows all association ends of SubLoadArea with other classes.

Table 462 – Association ends of LoadModel::SubLoadArea with other classes

[mult from]	[mult to] name	type	description
[1..1]	[1..*] LoadGroups	LoadGroup	The Loadgroups in the SubLoadArea.
[1..*]	[1..1] LoadArea	LoadArea	The LoadArea where the SubLoadArea belongs.
[0..1]	[0..1] ControlArea	ControlArea	inherited from: EnergyArea
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.11 Package Outage

6.11.1 General

An extension to the Core and Wires packages that models information on the current and planned network configuration. These entities are optional within typical network applications.

Figure 70 shows class diagram Datatypes.

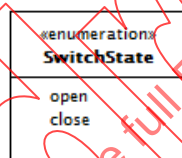


Figure 70 – Class diagram Outage::Datatypes

This diagram shows the data types specific to the Outage package.

Figure 71 shows class diagram Main.

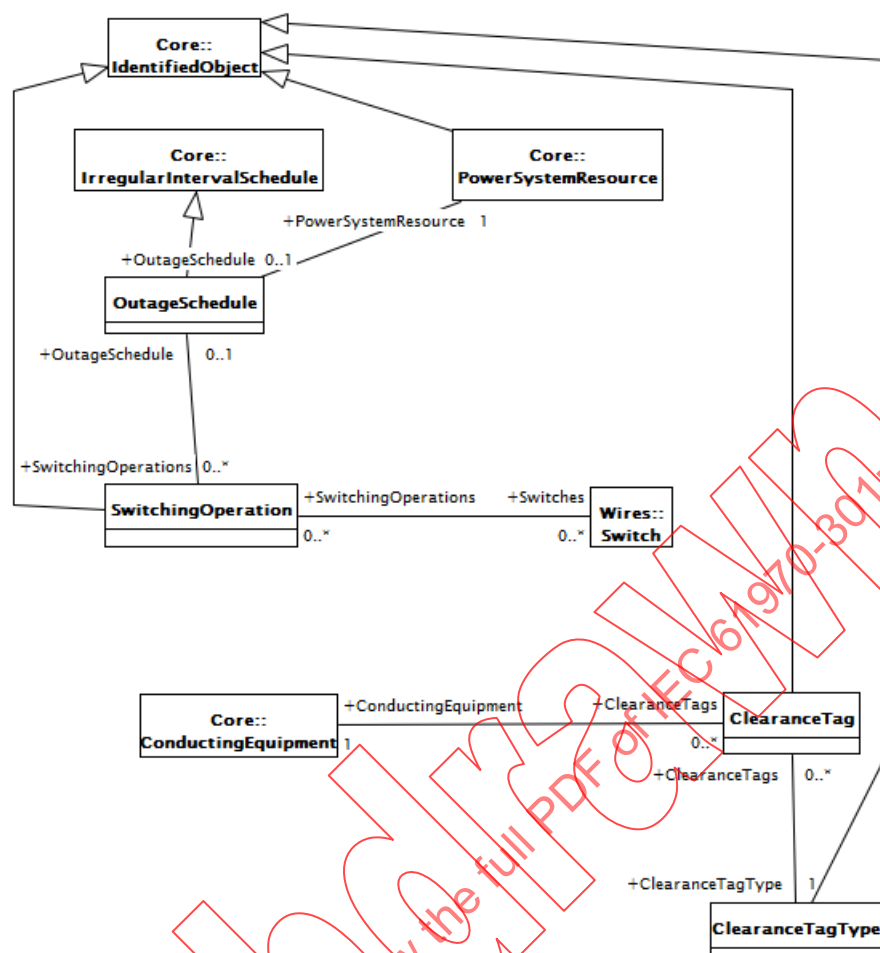


Figure 71 – Class diagram Outage::Main

This diagram shows all classes included in the Outage package as well as the key external classes that have associations with Outage classes.

6.11.2 ClearanceTag

A clearance tag that is used to authorize and schedule work on conducting equipment in the field. Tagged equipment is not available for commercial service.

Table 463 shows all attributes of ClearanceTag.

Table 463 – Attributes of Outage::ClearanceTag

name	type	description
authorityName	String	The name of the person who is authorized to issue the tag.
deenergizeReqFlag	Boolean	Set true if equipment must be deenergized.
groundReqFlag	Boolean	Set true if equipment must be grounded.
phaseCheckReqFlag	Boolean	Set true if equipment phasing must be checked.
tagIssueTime	DateTime	The time at which the clearance tag was issued.
workDescription	String	Description of the work to be performed.
workEndTime	DateTime	The time at which the clearance tag is scheduled to be removed.
workStartTime	DateTime	The time at which the clearance tag is scheduled to be set.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 464 shows all association ends of ClearanceTag with other classes.

Table 464 – Association ends of Outage::ClearanceTag with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] ConductingEquipment	ConductingEquipment	Conducting equipment of this clearance tags for authorized field work.
[0..*]	[1..1] ClearanceTagType	ClearanceTagType	The type of tag, depending on the purpose of the work to be performed and/or the type of supervisory control allowed.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.11.3 ClearanceTagType

Type of ClearanceTag. Could indicate the type of work to be performed and/or the type of supervisory control.

Table 465 shows all attributes of ClearanceTagType.

Table 465 – Attributes of Outage::ClearanceTagType

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 466 shows all association ends of ClearanceTagType with other classes.

Table 466 – Association ends of Outage::ClearanceTagType with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] ClearanceTags	ClearanceTag	The ClearanceTags currently being defined for this type.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.11.4 OutageSchedule

The period of time that a piece of equipment is out of service, for example, for maintenance or testing; including the equipment's active power rating while under maintenance. The X-axis represents absolute time and the Y-axis represents the equipment's available rating while out of service.

Table 467 shows all attributes of OutageSchedule.

Table 467 – Attributes of Outage::OutageSchedule

name	type	description
startTime	DateTime	inherited from: BasicIntervalSchedule
value1Unit	UnitSymbol	inherited from: BasicIntervalSchedule
value1Multiplier	UnitMultiplier	inherited from: BasicIntervalSchedule
value2Unit	UnitSymbol	inherited from: BasicIntervalSchedule
value2Multiplier	UnitMultiplier	inherited from: BasicIntervalSchedule
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 468 shows all association ends of OutageSchedule with other classes.

Table 468 – Association ends of Outage::OutageSchedule with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] PowerSystemResource	PowerSystemResource	A power system resource may have an outage schedule.
[0..1]	[0..*] SwitchingOperations	SwitchingOperation	An OutageSchedule may operate many switches.
[1..1]	[1..*] TimePoints	IrregularTimePoint	inherited from: IrregularIntervalSchedule
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.11.5 SwitchingOperation

A SwitchingOperation is used to define individual switch operations for an OutageSchedule. This OutageSchedule may be associated with another item of Substation such as a Transformer, Line, or Generator; or with the Switch itself as a PowerSystemResource. A Switch may be referenced by many OutageSchedules.

Table 469 shows all attributes of SwitchingOperation.

Table 469 – Attributes of Outage::SwitchingOperation

name	type	description
newState	SwitchState	The switch position that shall result from this SwitchingOperation.
operationTime	DateTime	Time of operation in same units as OutageSchedule.xAxisUnits.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 470 shows all association ends of SwitchingOperation with other classes.

Table 470 – Association ends of Outage::SwitchingOperation with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] Switches	Switch	A switch may be operated by many schedules.
[0..*]	[0..1] OutageSchedule	OutageSchedule	An OutageSchedule may operate many switches.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.11.6 SwitchState enumeration

Possible states for a switch.

Table 471 shows all literals of SwitchState.

Table 471 – Literals of Outage::SwitchState

literal	description
open	Switch is open.
close	Switch is closed.

6.12 Package AuxiliaryEquipment

6.12.1 General

Contains equipment which is not normal conducting equipment such as sensors, fault locators, and surge protectors. These devices do not define power carrying topological connections as conducting equipment, but are associated to terminals of other conducting equipment.

Figure 72 shows class diagram AuxiliaryEquipment.

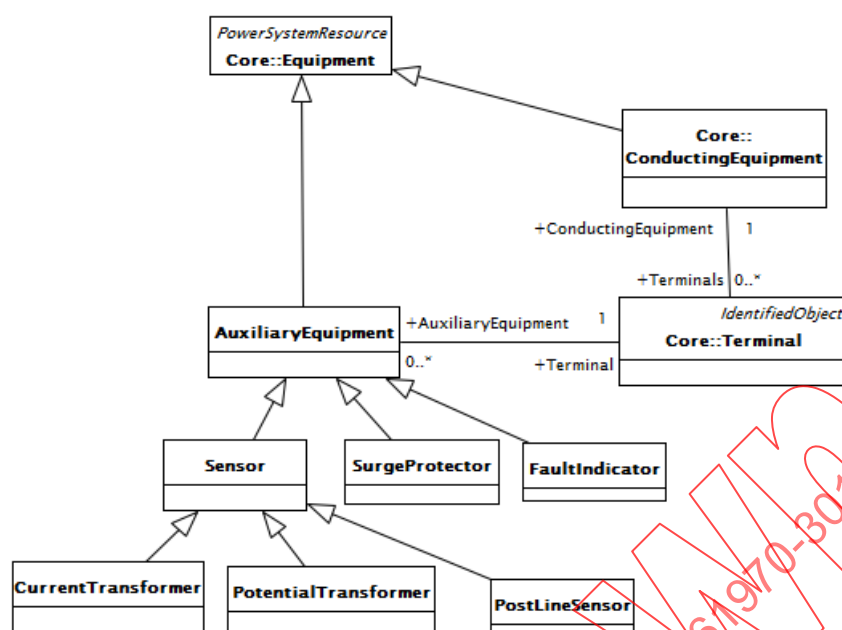


Figure 72 – Class diagram AuxiliaryEquipment::AuxiliaryEquipment

Inheritance and associations for auxiliary equipment.

6.12.2 AuxiliaryEquipment

AuxiliaryEquipment describe equipment that is not performing any primary functions but support for the equipment performing the primary function.

AuxiliaryEquipment is attached to primary equipment via an association with Terminal.

Table 472 shows all attributes of AuxiliaryEquipment.

Table 472 – Attributes of AuxiliaryEquipment::AuxiliaryEquipment

name	type	description
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 473 shows all association ends of AuxiliaryEquipment with other classes.

Table 473 – Association ends of AuxiliaryEquipment::AuxiliaryEquipment with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Terminal	Terminal	The Terminal at the equipment where the AuxiliaryEquipment is attached.
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.12.3 CurrentTransformer

Instrument transformer used to measure electrical qualities of the circuit that is being protected and/or monitored. Typically used as current transducer for the purpose of metering or protection. A typical secondary current rating would be 5 A.

Table 474 shows all attributes of CurrentTransformer.

Table 474 – Attributes of AuxiliaryEquipment::CurrentTransformer

name	type	description
accuracyClass	String	CT accuracy classification.
accuracyLimit	PerCent	Percent of rated current for which the CT remains accurate within specified limits.
coreCount	Integer	Number of cores.
ctClass	String	CT classification; i.e. class 10P.
maxRatio	Float	For multi-ratio CT's, the maximum permissible ratio attainable.
usage	String	Intended usage of the CT; i.e. metering, protection.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 475 shows all association ends of CurrentTransformer with other classes.

Table 475 – Association ends of AuxiliaryEquipment::CurrentTransformer with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Terminal	Terminal	inherited from: AuxiliaryEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.12.4 FaultIndicator

A FaultIndicator is typically only an indicator (which may or may not be remotely monitored), and not a piece of equipment that actually initiates a protection event. It is used for FLISR (Fault Location, Isolation and Restoration) purposes, assisting with the dispatch of crews to "most likely" part of the network (i.e. assists with determining circuit section where the fault most likely happened).

Table 476 shows all attributes of FaultIndicator.

Table 476 – Attributes of AuxiliaryEquipment::FaultIndicator

name	type	description
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 477 shows all association ends of FaultIndicator with other classes.

Table 477 – Association ends of AuxiliaryEquipment::FaultIndicator with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Terminal	Terminal	inherited from: AuxiliaryEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.12.5 PostLineSensor

A sensor used mainly in overhead distribution networks as the source of both current and voltage measurements.

Table 478 shows all attributes of PostLineSensor.

Table 478 – Attributes of AuxiliaryEquipment::PostLineSensor

name	type	description
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 479 shows all association ends of PostLineSensor with other classes.

Table 479 – Association ends of AuxiliaryEquipment::PostLineSensor with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Terminal	Terminal	inherited from: AuxiliaryEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.12.6 PotentialTransformer

Instrument transformer (also known as Voltage Transformer) used to measure electrical qualities of the circuit that is being protected and/or monitored. Typically used as voltage transducer for the purpose of metering, protection, or sometimes auxiliary substation supply. A typical secondary voltage rating would be 120 V.

Table 480 shows all attributes of PotentialTransformer.

Table 480 – Attributes of AuxiliaryEquipment::PotentialTransformer

name	type	description
accuracyClass	String	PT accuracy classification.
nominalRatio	Float	Nominal ratio between the primary and secondary voltage.
ptClass	String	Potential transformer (PT) classification.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 481 shows all association ends of PotentialTransformer with other classes.

Table 481 – Association ends of AuxiliaryEquipment::PotentialTransformer with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Terminal	Terminal	inherited from: AuxiliaryEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.12.7 Sensor

This class describe devices that transform a measured quantity into signals that can be presented at displays, used in control or be recorded.

Table 482 shows all attributes of Sensor.

Table 482 – Attributes of AuxiliaryEquipment::Sensor

name	type	description
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 483 shows all association ends of Sensor with other classes.

Table 483 – Association ends of AuxiliaryEquipment::Sensor with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Terminal	Terminal	inherited from: AuxiliaryEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.12.8 SurgeProtector

Shunt device, installed on the network, usually in the proximity of electrical equipment in order to protect the said equipment against transient voltage spikes caused by lightning or switching activity.

Table 484 shows all attributes of SurgeProtector.

Table 484 – Attributes of AuxiliaryEquipment::SurgeProtector

name	type	description
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 485 shows all association ends of SurgeProtector with other classes.

Table 485 – Association ends of AuxiliaryEquipment::SurgeProtector with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Terminal	Terminal	inherited from: AuxiliaryEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.13 Package Protection

6.13.1 General

An extension to the Core and Wires packages that models information for protection equipment such as relays. These entities are used within training simulators and distribution network fault location applications.

Figure 73 shows class diagram Main.

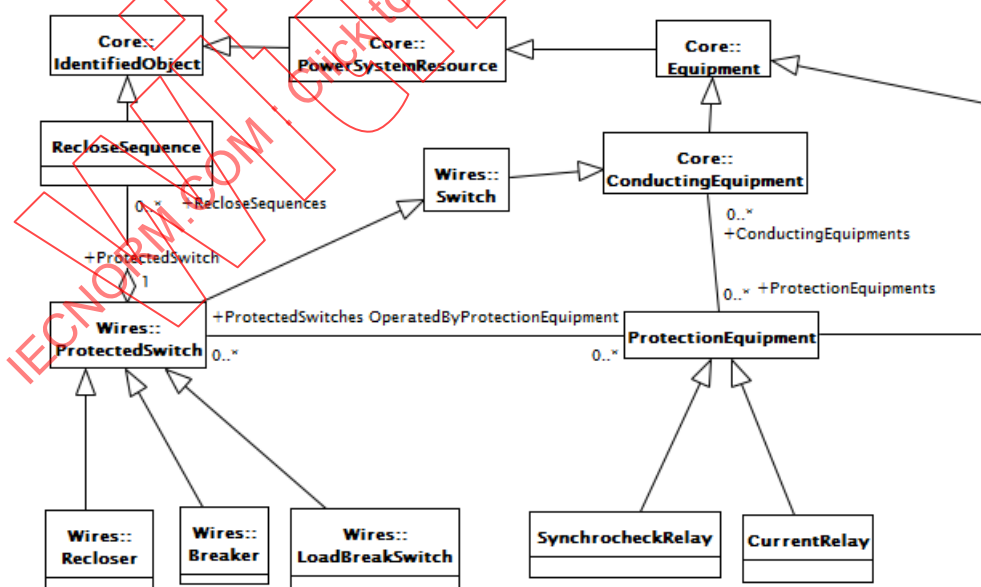


Figure 73 – Class diagram Protection::Main

This diagram shows all classes included in the Protection package as well as the key external classes that have associations with Protection classes.

6.13.2 CurrentRelay

A device that checks current flow values in any direction or designated direction.

Table 486 shows all attributes of CurrentRelay.

Table 486 – Attributes of Protection::CurrentRelay

name	type	description
currentLimit1	CurrentFlow	Current limit number 1 for inverse time pickup.
currentLimit2	CurrentFlow	Current limit number 2 for inverse time pickup.
currentLimit3	CurrentFlow	Current limit number 3 for inverse time pickup.
inverseTimeFlag	Boolean	Set true if the current relay has inverse time characteristic.
timeDelay1	Seconds	Inverse time delay number 1 for current limit number 1.
timeDelay2	Seconds	Inverse time delay number 2 for current limit number 2.
timeDelay3	Seconds	Inverse time delay number 3 for current limit number 3.
relayDelayTime	Seconds	inherited from: ProtectionEquipment
highLimit	Float	inherited from: ProtectionEquipment
lowLimit	Float	inherited from: ProtectionEquipment
powerDirectionFlag	Boolean	inherited from: ProtectionEquipment
unitSymbol	UnitSymbol	inherited from: ProtectionEquipment
unitMultiplier	UnitMultiplier	inherited from: ProtectionEquipment
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 487 shows all association ends of CurrentRelay with other classes.

Table 487 – Association ends of Protection::CurrentRelay with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] ConductingEquipments	ConductingEquipment	inherited from: ProtectionEquipment
[0..*]	[0..*] ProtectedSwitches	ProtectedSwitch	inherited from: ProtectionEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.13.3 ProtectionEquipment

An electrical device designed to respond to input conditions in a prescribed manner and after specified conditions are met to cause contact operation or similar abrupt change in associated electric control circuits, or simply to display the detected condition. Protection equipment are associated with conducting equipment and usually operate circuit breakers.

Table 488 shows all attributes of ProtectionEquipment.

Table 488 – Attributes of Protection::ProtectionEquipment

name	type	description
relayDelayTime	Seconds	The time delay from detection of abnormal conditions to relay operation.
highLimit	Float	The maximum allowable value.
lowLimit	Float	The minimum allowable value.
powerDirectionFlag	Boolean	Direction same as positive active power flow value.
unitSymbol	UnitSymbol	The unit of measure of the value.
unitMultiplier	UnitMultiplier	The unit multiplier of the value.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 489 shows all association ends of ProtectionEquipment with other classes.

Table 489 – Association ends of Protection::ProtectionEquipment with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] ConductingEquipments	ConductingEquipment	Protection equipment may be used to protect specific conducting equipment.
[0..*]	[0..*] ProtectedSwitches	ProtectedSwitch	Protected switches operated by this ProtectionEquipment.
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.13.4 RecloseSequence

A reclose sequence (open and close) is defined for each possible reclosure of a breaker.

Table 490 shows all attributes of RecloseSequence.

Table 490 – Attributes of Protection::RecloseSequence

name	type	description
recloseDelay	Seconds	Indicates the time lapse before the reclose step will execute a reclose.
recloseStep	Integer	Indicates the ordinal position of the reclose step relative to other steps in the sequence.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 491 shows all association ends of RecloseSequence with other classes.

Table 491 – Association ends of Protection::RecloseSequence with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] ProtectedSwitch	ProtectedSwitch	A breaker may have zero or more automatic reclosures after a trip occurs.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.13.5 SynchrocheckRelay

A device that operates when two AC circuits are within the desired limits of frequency, phase angle, and voltage, to permit or to cause the paralleling of these two circuits. Used to prevent the paralleling of non-synchronous topological islands.

Table 492 shows all attributes of SynchrocheckRelay.

Table 492 – Attributes of Protection::SynchrocheckRelay

name	type	description
maxAngleDiff	AngleRadians	The maximum allowable voltage vector phase angle difference across the open device.
maxFreqDiff	Frequency	The maximum allowable frequency difference across the open device.
maxVoltDiff	Voltage	The maximum allowable difference voltage across the open device.
relayDelayTime	Seconds	inherited from: ProtectionEquipment
highLimit	Float	inherited from: ProtectionEquipment
lowLimit	Float	inherited from: ProtectionEquipment
powerDirectionFlag	Boolean	inherited from: ProtectionEquipment
unitSymbol	UnitSymbol	inherited from: ProtectionEquipment
unitMultiplier	UnitMultiplier	inherited from: ProtectionEquipment
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 493 shows all association ends of SynchrocheckRelay with other classes.

Table 493 – Association ends of Protection::SynchrocheckRelay with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] ConductingEquipments	ConductingEquipment	inherited from: ProtectionEquipment
[0..*]	[0..*] ProtectedSwitches	ProtectedSwitch	inherited from: ProtectionEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.14 Package Equivalents

6.14.1 General

The equivalents package models equivalent networks.

Figure 74 shows class diagram Main.

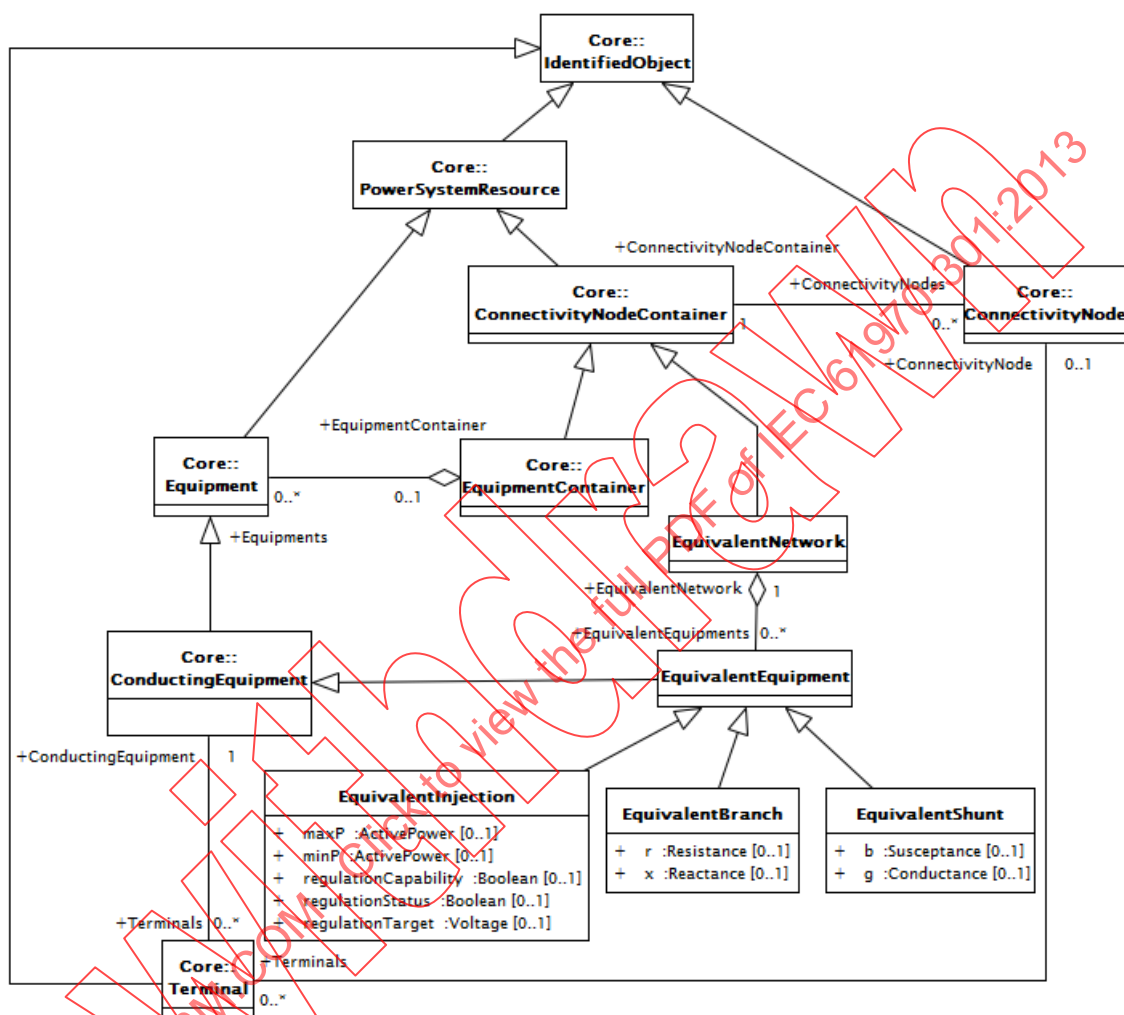


Figure 74 – Class diagram Equivalents::Main

Shows model for equivalent networks.

6.14.2 EquivalentBranch

The class represents equivalent branches.

Table 494 shows all attributes of EquivalentBranch.

Table 494 – Attributes of Equivalents::EquivalentBranch

name	type	description
r	Resistance	Positive sequence series resistance of the reduced branch.
x	Reactance	Positive sequence series reactance of the reduced branch.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 495 shows all association ends of EquivalentBranch with other classes.

Table 495 – Association ends of Equivalents::EquivalentBranch with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] EquivalentNetwork	EquivalentNetwork	inherited from: EquivalentEquipment
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.14.3 EquivalentEquipment

The class represents equivalent objects that are the result of a network reduction. The class is the base for equivalent objects of different types.

Table 496 shows all attributes of EquivalentEquipment.

Table 496 – Attributes of Equivalents::EquivalentEquipment

name	type	description
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 497 shows all association ends of EquivalentEquipment with other classes.

Table 497 – Association ends of Equivalents::EquivalentEquipment with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] EquivalentNetwork	EquivalentNetwork	The equivalent where the reduced model belongs.
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.14.4 EquivalentInjection

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

Table 498 shows all attributes of EquivalentInjection.

Table 498 – Attributes of Equivalents::EquivalentInjection

name	type	description
maxP	ActivePower	Minimum active power of the injection.
minP	ActivePower	Maximum active power of the injection.
regulationCapability	Boolean	Specifies whether or not the EquivalentInjection has the capability to regulate the local voltage.
regulationStatus	Boolean	Specifies the default regulation status of the EquivalentInjection. True is regulating. False is not regulating.
regulationTarget	Voltage	The target voltage for voltage regulation.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 499 shows all association ends of EquivalentInjection with other classes.

Table 499 – Association ends of Equivalents::EquivalentInjection with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] EquivalentNetwork	EquivalentNetwork	inherited from: EquivalentEquipment
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.14.5 EquivalentNetwork

A class that represents an external meshed network that has been reduced to an electrically equivalent model. The ConnectivityNodes contained in the equivalent are intended to reflect internal nodes of the equivalent. The boundary Connectivity nodes where the equivalent connects outside itself are NOT contained by the equivalent.

Table 500 shows all attributes of EquivalentNetwork.

Table 500 – Attributes of Equivalents::EquivalentNetwork

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 501 shows all association ends of EquivalentNetwork with other classes.

Table 501 – Association ends of Equivalents::EquivalentNetwork with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] EquivalentEquipments	EquivalentEquipment	The associated reduced equivalents.
[1..1]	[0..*] ConnectivityNodes	ConnectivityNode	inherited from: ConnectivityNodeContainer
[0..1]	[0..*] TopologicalNode	TopologicalNode	inherited from: ConnectivityNodeContainer
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.14.6 EquivalentShunt

The class represents equivalent shunts.

Table 502 shows all attributes of EquivalentShunt.

Table 502 – Attributes of Equivalents::EquivalentShunt

name	type	description
b	Susceptance	Positive sequence shunt susceptance.
g	Conductance	Positive sequence shunt conductance.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 503 shows all association ends of EquivalentShunt with other classes.

Table 503 – Association ends of EquivalentShunt with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] EquivalentNetwork	EquivalentNetwork	inherited from: EquivalentEquipment
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15 Package Meas

6.15.1 General

Contains entities that describe dynamic measurement data exchanged between applications.

Figure 75 shows class diagram Datatypes.

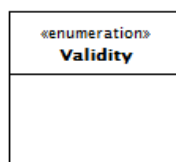


Figure 75 – Class diagram Meas::Datatypes

Shows datatypes for measurements.

Figure 76 shows class diagram Control.

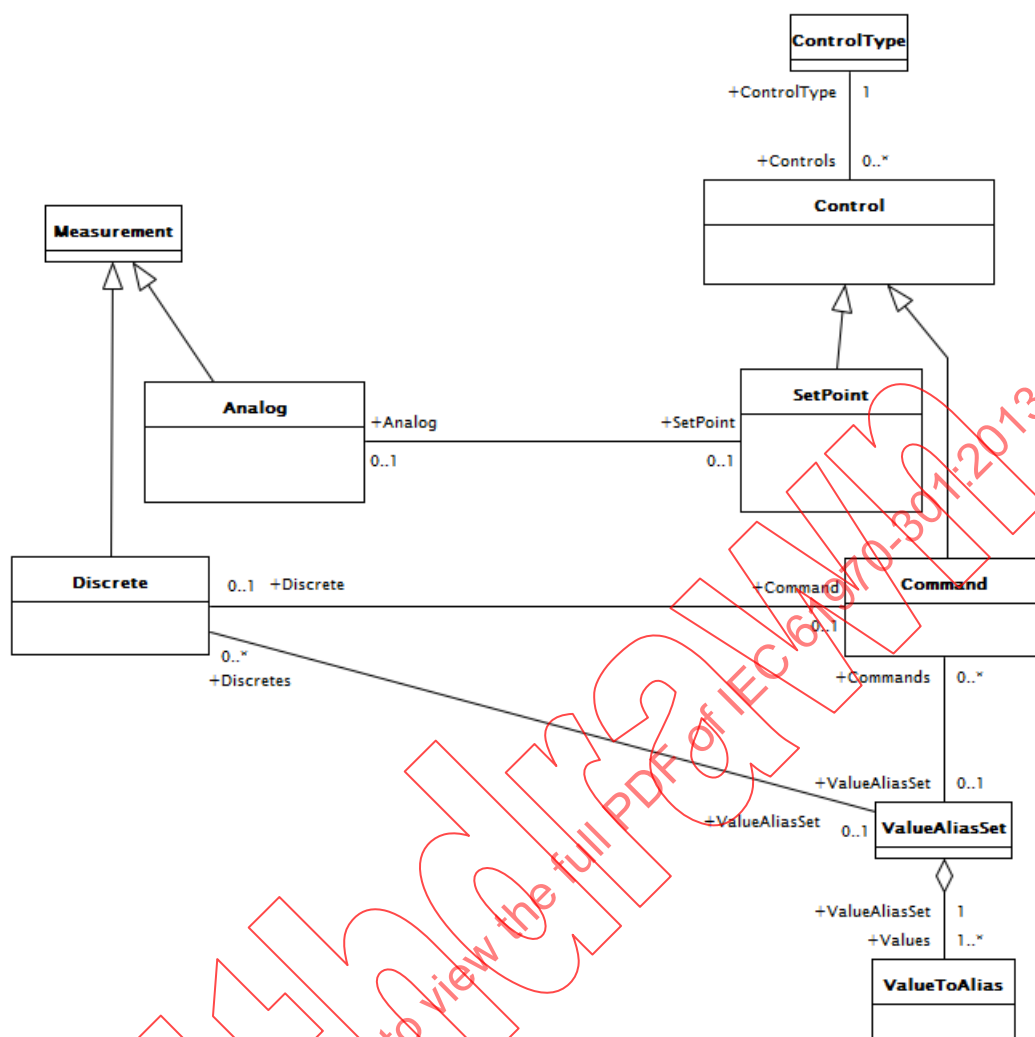


Figure 76 – Class diagram Meas::Control

This diagram shows all classes included in the Meas package.

Figure 77 shows class diagram Measurement.

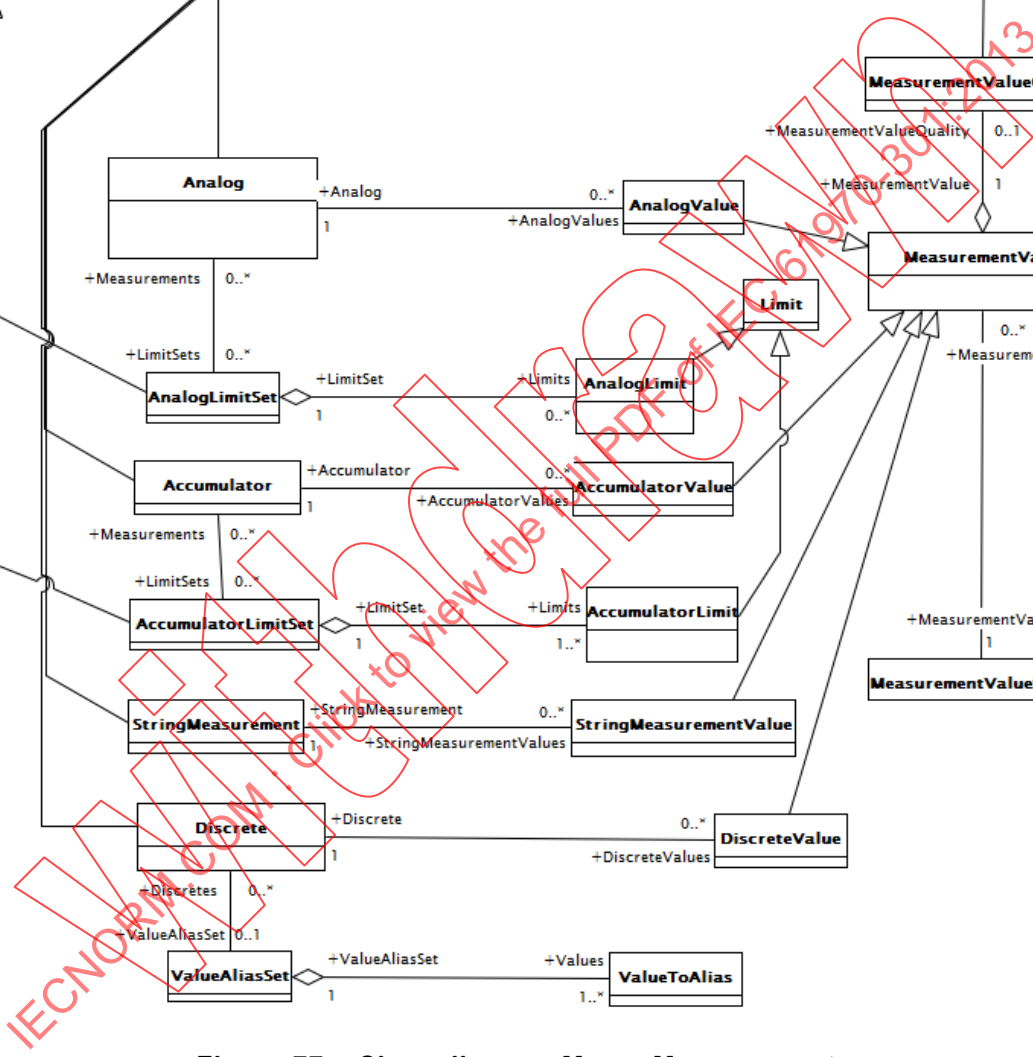


Figure 77 – Class diagram Meas::Measurement

This diagram shows classes central to the Measurement package and connections to some external classes.

Figure 78 shows class diagram MeasurementInheritance.

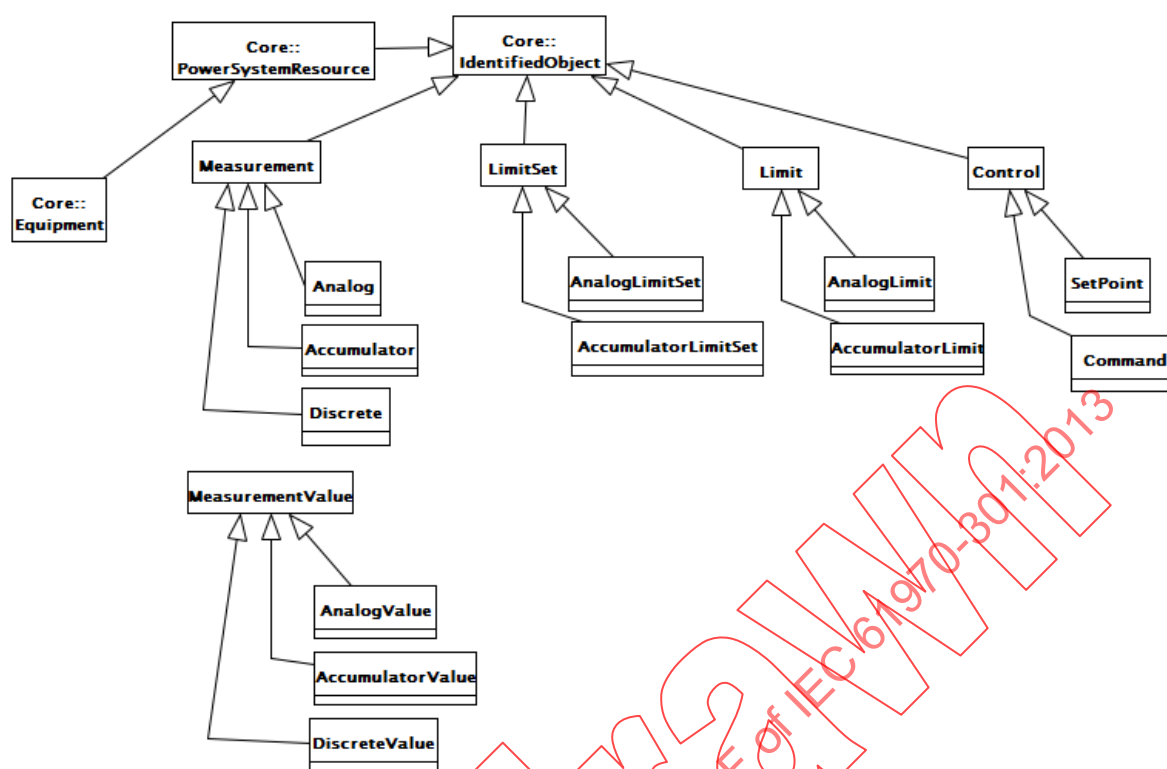


Figure 78 – Class diagram Meas::MeasurementInheritance

This diagram shows the measurement classes inheritance.

Figure 79 shows class diagram Quality.

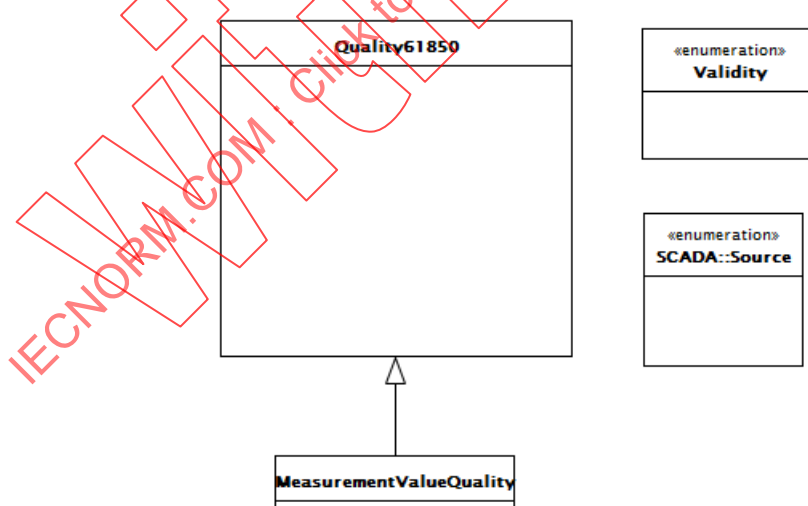


Figure 79 – Class diagram Meas::Quality

This diagram shows the details of the quality codes. The quality flags can be used also in other packages not only specific to MeasurementValues. A generalized Quality class is added and inherited into MeasurementValueQuality.

6.15.2 Accumulator

Accumulator represents a accumulated (counted) Measurement, e.g. an energy value.

Table 504 shows all attributes of Accumulator.

Table 504 – Attributes of Meas::Accumulator

name	type	description
maxValue	Integer	Normal value range maximum for any of the MeasurementValue.values. Used for scaling, e.g. in bar graphs or of telemetered raw values.
measurementType	String	inherited from: Measurement
phases	PhaseCode	inherited from: Measurement
unitMultiplier	UnitMultiplier	inherited from: Measurement
unitSymbol	UnitSymbol	inherited from: Measurement
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 505 shows all association ends of Accumulator with other classes.

Table 505 – Association ends of Meas::Accumulator with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] AccumulatorValues	AccumulatorValue	The values connected to this measurement.
[0..*]	[0..*] LimitSets	AccumulatorLimitSet	A measurement may have zero or more limit ranges defined for it.
[0..*]	[0..1] Terminal	Terminal	inherited from: Measurement
[0..*]	[0..1] PowerSystemResource	PowerSystemResource	inherited from: Measurement
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.3 AccumulatorLimit

Limit values for Accumulator measurements.

Table 506 shows all attributes of AccumulatorLimit.

Table 506 – Attributes of Meas::AccumulatorLimit

name	type	description
value	Integer	The value to supervise against. The value is positive.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 507 shows all association ends of AccumulatorLimit with other classes.

Table 507 – Association ends of Meas::AccumulatorLimit with other classes

[mult from]	[mult to] name	type	description
[1..*]	[1..1] LimitSet	AccumulatorLimitSet	The set of limits.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.4 AccumulatorLimitSet

An AccumulatorLimitSet specifies a set of Limits that are associated with an Accumulator measurement.

Table 508 shows all attributes of AccumulatorLimitSet.

Table 508 – Attributes of Meas::AccumulatorLimitSet

name	type	description
isPercentageLimits	Boolean	inherited from: LimitSet
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 509 shows all association ends of AccumulatorLimitSet with other classes.

Table 509 – Association ends of Meas::AccumulatorLimitSet with other classes

[mult from]	[mult to] name	type	description
[1..1]	[1..*] Limits	AccumulatorLimit	The limit values used for supervision of Measurements.
[0..*]	[0..*] Measurements	Accumulator	The Measurements using the LimitSet.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.5 AccumulatorValue

AccumulatorValue represents an accumulated (counted) MeasurementValue.

Table 510 shows all attributes of AccumulatorValue.

Table 510 – Attributes of Meas::AccumulatorValue

name	type	description
value	Integer	The value to supervise. The value is positive.
timeStamp	DateTime	inherited from: MeasurementValue
sensorAccuracy	PerCent	inherited from: MeasurementValue
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 511 shows all association ends of AccumulatorValue with other classes.

Table 511 – Association ends of Meas::AccumulatorValue with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Accumulator	Accumulator	Measurement to which this value is connected.
[1..1]	[0..1] RemoteSource	RemoteSource	inherited from: MeasurementValue
[1..1]	[0..1] MeasurementValueQuality	MeasurementValueQuality	inherited from: MeasurementValue
[0..*]	[1..1] MeasurementValueSource	MeasurementValueSource	inherited from: MeasurementValue
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.6 Analog

Analog represents an analog Measurement.

Table 512 shows all attributes of Analog.

Table 512 – Attributes of Meas::Analog

name	type	description
maxValue	Float	Normal value range maximum for any of the MeasurementValue.values. Used for scaling, e.g. in bar graphs or of telemetered raw values.
minValue	Float	Normal value range minimum for any of the MeasurementValue.values. Used for scaling, e.g. in bar graphs or of telemetered raw values.
normalValue	Float	Normal measurement value, e.g., used for percentage calculations.
positiveFlowIn	Boolean	If true then this measurement is an active power, reactive power or current with the convention that a positive value measured at the Terminal means power is flowing into the related PowerSystemResource.
measurementType	String	inherited from: Measurement
phases	PhaseCode	inherited from: Measurement
unitMultiplier	UnitMultiplier	inherited from: Measurement
unitSymbol	UnitSymbol	inherited from: Measurement
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 513 shows all association ends of Analog with other classes.

Table 513 – Association ends of Meas::Analog with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] AnalogValues	AnalogValue	The values connected to this measurement.
[0..*]	[0..*] LimitSets	AnalogLimitSet	A measurement may have zero or more limit ranges defined for it.
[0..1]	[0..1] SetPoint	SetPoint	The Control variable associated with the Measurement.
[0..*]	[0..1] Terminal	Terminal	inherited from: Measurement
[0..*]	[0..1] PowerSystemResource	PowerSystemResource	inherited from: Measurement
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.7 AnalogLimit

Limit values for Analog measurements.

Table 514 shows all attributes of AnalogLimit.

Table 514 – Attributes of Meas::AnalogLimit

name	type	description
value	Float	The value to supervise against.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 515 shows all association ends of AnalogLimit with other classes.

Table 515 – Association ends of Meas::AnalogLimit with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] LimitSet	AnalogLimitSet	The set of limits.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.8 AnalogLimitSet

An AnalogLimitSet specifies a set of Limits that are associated with an Analog measurement.

Table 516 shows all attributes of AnalogLimitSet.

Table 516 – Attributes of Meas::AnalogLimitSet

name	type	description
isPercentageLimits	Boolean	inherited from: LimitSet
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 517 shows all association ends of AnalogLimitSet with other classes.

Table 517 – Association ends of Meas::AnalogLimitSet with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] Limits	AnalogLimit	The limit values used for supervision of Measurements.
[0..*]	[0..*] Measurements	Analog	The Measurements using the LimitSet.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.9 AnalogValue

AnalogValue represents an analog MeasurementValue.

Table 518 shows all attributes of AnalogValue.

Table 518 – Attributes of Meas::AnalogValue

name	type	description
value	Float	The value to supervise.
timeStamp	DateTime	inherited from: MeasurementValue
sensorAccuracy	PerCent	inherited from: MeasurementValue
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 519 shows all association ends of AnalogValue with other classes.

Table 519 – Association ends of Meas::AnalogValue with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Analog	Analog	Measurement to which this value is connected.
[1..1]	[0..*] AltGeneratingUnit	AltGeneratingUnitMeas	The alternate generating unit for which this measurement value applies.
[1..1]	[0..*] AltTieMeas	AltTieMeas	The usage of the measurement within the control area specification.
[1..1]	[0..1] RemoteSource	RemoteSource	inherited from: MeasurementValue
[1..1]	[0..1] MeasurementValueQuality	MeasurementValueQuality	inherited from: MeasurementValue
[0..*]	[1..1] MeasurementValueSource	MeasurementValueSource	inherited from: MeasurementValue
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.10 Command

A Command is a discrete control used for supervisory control.

Table 520 shows all attributes of Command.

Table 520 – Attributes of Meas::Command

name	type	description
normalValue	Integer	Normal value for Control.value e.g. used for percentage scaling.
value	Integer	The value representing the actuator output.
operationInProgress	Boolean	inherited from: Control
timeStamp	DateTime	inherited from: Control
unitMultiplier	UnitMultiplier	inherited from: Control
unitSymbol	UnitSymbol	inherited from: Control
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 521 shows all association ends of Command with other classes.

Table 521 – Association ends of Meas::Command with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] ValueAliasSet	ValueAliasSet	The Commands using the set for translation.
[0..1]	[0..1] Discrete	Discrete	The Measurement variable used for control.
[1..*]	[0..1] RegulatingCondEq	RegulatingCondEq	inherited from: Control
[1..1]	[0..1] RemoteControl	RemoteControl	inherited from: Control
[0..*]	[1..1] ControlType	ControlType	inherited from: Control
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.11 Control

Control is used for supervisory/device control. It represents control outputs that are used to change the state in a process, e.g. close or open breaker, a set point value or a raise lower command.

Table 522 shows all attributes of Control.

Table 522 – Attributes of Meas::Control

name	type	description
operationInProgress	Boolean	Indicates that a client is currently sending control commands that has not completed.
timeStamp	DateTime	The last time a control output was sent.
unitMultiplier	UnitMultiplier	The unit multiplier of the controlled quantity.
unitSymbol	UnitSymbol	The unit of measure of the controlled quantity.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 523 shows all association ends of Control with other classes.

Table 523 – Association ends of Meas::Control with other classes

[mult from]	[mult to] name	type	description
[1..*]	[0..1] RegulatingCondEq	RegulatingCondEq	Regulating device governed by this control output.
[1..1]	[0..1] RemoteControl	RemoteControl	The remote point controlling the physical actuator.
[0..*]	[1..1] ControlType	ControlType	The type of control.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.12 ControlType

Specifies the type of Control, e.g. BreakerOn/Off, GeneratorVoltageSetPoint, TieLineFlow, etc. The ControlType.name shall be unique among all specified types and describe the type. The ControlType.aliasName is meant to be used for localization.

Table 524 shows all attributes of ControlType.

Table 524 – Attributes of Meas::ControlType

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 525 shows all association ends of ControlType with other classes.

Table 525 – Association ends of Meas::ControlType with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] Controls	Control	The controls having the control type.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.13 Discrete

Discrete represents a discrete Measurement, i.e. a Measurement representing discrete values, e.g. a Breaker position.

Table 526 shows all attributes of Discrete.

Table 526 – Attributes of Meas::Discrete

name	type	description
maxValue	Integer	Normal value range maximum for any of the MeasurementValue.values. Used for scaling, e.g. in bar graphs or of telemetered raw values.
minValue	Integer	Normal value range minimum for any of the MeasurementValue.values. Used for scaling, e.g. in bar graphs or of telemetered raw values.
normalValue	Integer	Normal measurement value, e.g., used for percentage calculations.
measurementType	String	inherited from: Measurement
phases	PhaseCode	inherited from: Measurement
unitMultiplier	UnitMultiplier	inherited from: Measurement
unitSymbol	UnitSymbol	inherited from: Measurement
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 527 shows all association ends of Discrete with other classes.

Table 527 – Association ends of Meas::Discrete with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..1] Command	Command	The Control variable associated with the Measurement.
[1..1]	[0..*] DiscreteValues	DiscreteValue	The values connected to this measurement.
[0..*]	[0..1] ValueAliasSet	ValueAliasSet	The ValueAliasSet used for translation of a MeasurementValue.value to a name.
[0..*]	[0..1] Terminal	Terminal	inherited from: Measurement
[0..*]	[0..1] PowerSystemResource	PowerSystemResource	inherited from: Measurement
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.14 DiscreteValue

DiscreteValue represents a discrete MeasurementValue.

Table 528 shows all attributes of DiscreteValue.

Table 528 – Attributes of Meas::DiscreteValue

name	type	description
value	Integer	The value to supervise.
timeStamp	DateTime	inherited from: MeasurementValue
sensorAccuracy	PerCent	inherited from: MeasurementValue
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 529 shows all association ends of DiscreteValue with other classes.

Table 529 – Association ends of Meas::DiscreteValue with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Discrete	Discrete	Measurement to which this value is connected.
[1..1]	[0..1] RemoteSource	RemoteSource	inherited from: MeasurementValue
[1..1]	[0..1] MeasurementValueQuality	MeasurementValueQuality	inherited from: MeasurementValue
[0..*]	[1..1] MeasurementValueSource	MeasurementValueSource	inherited from: MeasurementValue
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.15 Limit

Specifies one limit value for a Measurement. A Measurement typically has several limits that are kept together by the LimitSet class. The actual meaning and use of a Limit instance (i.e., if it is an alarm or warning limit or if it is a high or low limit) is not captured in the Limit class. However the name of a Limit instance may indicate both meaning and use.

Table 530 shows all attributes of Limit.

Table 530 – Attributes of Meas::Limit

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 531 shows all association ends of Limit with other classes.

Table 531 – Association ends of Meas::Limit with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.16 LimitSet

Specifies a set of Limits that are associated with a Measurement. A Measurement may have several LimitSets corresponding to seasonal or other changing conditions. The condition is captured in the name and description attributes. The same LimitSet may be used for several Measurements. In particular percentage limits are used this way.

Table 532 shows all attributes of LimitSet.

Table 532 – Attributes of Meas::LimitSet

name	type	description
isPercentageLimits	Boolean	Tells if the limit values are in percentage of normalValue or the specified Unit for Measurements and Controls.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 533 shows all association ends of LimitSet with other classes.

Table 533 – Association ends of Meas::LimitSet with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.17 Measurement

A Measurement represents any measured, calculated or non-measured non-calculated quantity. Any piece of equipment may contain Measurements, e.g. a substation may have temperature measurements and door open indications, a transformer may have oil temperature and tank pressure measurements, a bay may contain a number of power flow measurements and a Breaker may contain a switch status measurement.

The PSR – Measurement association is intended to capture this use of Measurement and is included in the naming hierarchy based on EquipmentContainer. The naming hierarchy typically has Measurements as leafs, e.g. Substation-VoltageLevel-Bay-Switch-Measurement.

Some Measurements represent quantities related to a particular sensor location in the network, e.g. a voltage transformer (PT) at a busbar or a current transformer (CT) at the bar between a breaker and an isolator. The sensing position is not captured in the PSR – Measurement association. Instead it is captured by the Measurement – Terminal association that is used to define the sensing location in the network topology. The location is defined by the connection of the Terminal to ConductingEquipment.

If both a Terminal and PSR are associated, and the PSR is of type ConductingEquipment, the associated Terminal should belong to that ConductingEquipment instance.

When the sensor location is needed both Measurement-PSR and Measurement-Terminal are used. The Measurement-Terminal association is never used alone.

Table 534 shows all attributes of Measurement.

Table 534 – Attributes of Meas::Measurement

name	type	description
measurementType	String	Specifies the type of measurement. For example, this specifies if the measurement represents an indoor temperature, outdoor temperature, bus voltage, line flow, etc.
phases	PhaseCode	Indicates to which phases the measurement applies and avoids the need to use 'measurementType' to also encode phase information (which would explode the types). Since Controls have associations with Measurements, they will have the capability to handle each phase. The phase information in Measurement, along with 'measurementType' and 'phases' uniquely defines a Measurement for a device, based on normal network phase. Their meaning will not change when the computed energizing phasing is changed due to jumpers or other reasons.
unitMultiplier	UnitMultiplier	The unit multiplier of the measured quantity.
unitSymbol	UnitSymbol	The unit of measure of the measured quantity.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 535 shows all association ends of Measurement with other classes.

Table 535 – Association ends of Meas::Measurement with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] Terminal	Terminal	One or more measurements may be associated with a terminal in the network.
[0..*]	[0..1] PowerSystemResource	PowerSystemResource	The power system resource that contains the measurement.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.18 MeasurementValue

The current state for a measurement. A state value is an instance of a measurement from a specific source. Measurements can be associated with many state values, each representing a different source for the measurement.

Table 536 shows all attributes of MeasurementValue.

Table 536 – Attributes of Meas::MeasurementValue

name	type	description
timeStamp	DateTime	The time when the value was last updated
sensorAccuracy	PerCent	The limit, expressed as a percentage of the sensor maximum, that errors will not exceed when the sensor is used under reference conditions.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 537 shows all association ends of MeasurementValue with other classes.

Table 537 – Association ends of Meas::MeasurementValue with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..1] RemoteSource	RemoteSource	Link to the physical telemetered point associated with this measurement.
[1..1]	[0..1] MeasurementValueQuality	MeasurementValueQuality	A MeasurementValue has a MeasurementValueQuality associated with it.
[0..*]	[1..1] MeasurementValueSource	MeasurementValueSource	A reference to the type of source that updates the MeasurementValue, e.g. SCADA, CCLink, manual, etc. User conventions for the names of sources are contained in the introduction to IEC 61970-301.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.19 MeasurementValueQuality

Measurement quality flags. Bits 0-10 are defined for substation automation in IEC 61850-7-3. Bits 11-15 are reserved for future expansion by that document. Bits 16-31 are reserved for EMS applications.

Table 538 shows all attributes of MeasurementValueQuality.

Table 538 – Attributes of Meas::MeasurementValueQuality

name	type	description
badReference	Boolean	inherited from: Quality61850
estimatorReplaced	Boolean	inherited from: Quality61850
failure	Boolean	inherited from: Quality61850
oldData	Boolean	inherited from: Quality61850
operatorBlocked	Boolean	inherited from: Quality61850
oscillatory	Boolean	inherited from: Quality61850
outOfRange	Boolean	inherited from: Quality61850
overFlow	Boolean	inherited from: Quality61850
source	Source	inherited from: Quality61850
suspect	Boolean	inherited from: Quality61850
test	Boolean	inherited from: Quality61850
validity	Validity	inherited from: Quality61850

Table 539 shows all association ends of MeasurementValueQuality with other classes.

Table 539 – Association ends of Meas::MeasurementValueQuality with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] MeasurementValue	MeasurementValue	A MeasurementValue has a MeasurementValueQuality associated with it.

6.15.20 MeasurementValueSource

MeasurementValueSource describes the alternative sources updating a MeasurementValue. User conventions for how to use the MeasurementValueSource attributes are described in the introduction to this standard.

Table 540 shows all attributes of MeasurementValueSource.

Table 540 – Attributes of Meas::MeasurementValueSource

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 541 shows all association ends of MeasurementValueSource with other classes.

Table 541 – Association ends of Meas::MeasurementValueSource with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] MeasurementValues	MeasurementValue	The MeasurementValues updated by the source.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.21 Quality61850 root class

Quality flags in this class are as defined in IEC 61850-7-3, except for estimatorReplaced, which has been included in this class for convenience.

Table 542 shows all attributes of Quality61850.

Table 542 – Attributes of Meas::Quality61850

name	type	description
badReference	Boolean	Measurement value may be incorrect due to a reference being out of calibration.
estimatorReplaced	Boolean	Value has been replaced by State Estimator. estimatorReplaced is not an IEC61850 quality bit but has been put in this class for convenience.
failure	Boolean	This identifier indicates that a supervision function has detected an internal or external failure, e.g. communication failure.
oldData	Boolean	Measurement value is old and possibly invalid, as it has not been successfully updated during a specified time interval.
operatorBlocked	Boolean	Measurement value is blocked and hence unavailable for transmission.
oscillatory	Boolean	To prevent some overload of the communication it is sensible to detect and suppress oscillating (fast changing) binary inputs. If a signal changes in a defined time (tosc) twice in the same direction (from 0 to 1 or from 1 to 0) then oscillation is detected and the detail quality identifier "oscillatory" is set. If it is detected a configured number of transient changes could be passed by. In this time the validity status "questionable" is set. If after this defined numbers of changes the signal is still in the oscillating state the value shall be set either to the opposite state of the previous stable value or to a defined default value. In this case the validity status "questionable" is reset and "invalid" is set as long as the signal is oscillating. If it is configured such that no transient changes should be passed by then the validity status "invalid" is set immediately in addition to the detail quality identifier "oscillatory" (used for status information only).
outOfRange	Boolean	Measurement value is beyond a predefined range of value.
overflow	Boolean	Measurement value is beyond the capability of being represented properly. For example, a counter value overflows from maximum count back to a value of zero.
source	Source	Source gives information related to the origin of a value. The value may be acquired from the process, defaulted or substituted.
suspect	Boolean	A correlation function has detected that the value is not consistent with other values. Typically set by a network State Estimator.
test	Boolean	Measurement value is transmitted for test purposes.
validity	Validity	Validity of the measurement value.

6.15.22 SetPoint

A SetPoint is an analog control used for supervisory control.

Table 543 shows all attributes of SetPoint.

Table 543 – Attributes of Meas::SetPoint

name	type	description
maxValue	Float	Normal value range maximum for any of the Control.value. Used for scaling, e.g. in bar graphs.
minValue	Float	Normal value range minimum for any of the Control.value. Used for scaling, e.g. in bar graphs.
normalValue	Float	Normal value for Control.value e.g. used for percentage scaling.
value	Float	The value representing the actuator output.
operationInProgress	Boolean	inherited from: Control
timeStamp	DateTime	inherited from: Control
unitMultiplier	UnitMultiplier	inherited from: Control
unitSymbol	UnitSymbol	inherited from: Control
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 544 shows all association ends of SetPoint with other classes.

Table 544 – Association ends of Meas::SetPoint with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..1] Analog	Analog	The Measurement variable used for control.
[1..*]	[0..1] RegulatingCondEq	RegulatingCondEq	inherited from: Control
[1..1]	[0..1] RemoteControl	RemoteControl	inherited from: Control
[0..*]	[1..1] ControlType	ControlType	inherited from: Control
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.23 StringMeasurement

StringMeasurement represents a measurement with values of type string.

Table 545 shows all attributes of StringMeasurement.

Table 545 – Attributes of Meas::StringMeasurement

name	type	description
measurementType	String	inherited from: Measurement
phases	PhaseCode	inherited from: Measurement
unitMultiplier	UnitMultiplier	inherited from: Measurement
unitSymbol	UnitSymbol	inherited from: Measurement
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 546 shows all association ends of StringMeasurement with other classes.

Table 546 – Association ends of Meas::StringMeasurement with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] StringMeasurementValues	StringMeasurementValue	The values connected to this measurement.
[0..*]	[0..1] Terminal	Terminal	inherited from: Measurement
[0..*]	[0..1] PowerSystemResource	PowerSystemResource	inherited from: Measurement
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.24 StringMeasurementValue

StringMeasurementValue represents a measurement value of type string.

Table 547 shows all attributes of StringMeasurementValue.

Table 547 – Attributes of Meas::StringMeasurementValue

name	type	description
value	String	The value to supervise.
timeStamp	DateTime	inherited from: MeasurementValue
sensorAccuracy	PerCent	inherited from: MeasurementValue
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 548 shows all association ends of StringMeasurementValue with other classes.

Table 548 – Association ends of Meas::StringMeasurementValue with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] StringMeasurement	StringMeasurement	Measurement to which this value is connected.
[1..1]	[0..1] RemoteSource	RemoteSource	inherited from: MeasurementValue
[1..1]	[0..1] MeasurementValueQuality	MeasurementValueQuality	inherited from: MeasurementValue
[0..*]	[1..1] MeasurementValueSource	MeasurementValueSource	inherited from: MeasurementValue
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.25 Validity enumeration

Validity for MeasurementValue.

Table 549 shows all literals of Validity.

Table 549 – Literals of Meas::Validity

literal	description
GOOD	The value is marked good if no abnormal condition of the acquisition function or the information source is detected.
QUESTIONABLE	The value is marked questionable if a supervision function detects an abnormal behaviour, however the value could still be valid. The client is responsible for determining whether or not values marked "questionable" should be used.
INVALID	The value is marked invalid when a supervision function recognises abnormal conditions of the acquisition function or the information source (missing or non-operating updating devices). The value is not defined under this condition. The mark invalid is used to indicate to the client that the value may be incorrect and shall not be used.

6.15.26 ValueAliasSet

Describes the translation of a set of values into a name and is intended to facilitate custom translations. Each ValueAliasSet has a name, description, etc. A specific Measurement may represent a discrete state like Open, Closed, Intermediate, etc. This requires a translation from the MeasurementValue.value number to a string, e.g. 0 = "Invalid", 1 = "Open", 2 = "Closed", 3 = "Intermediate". Each ValueToAlias member in ValueAliasSet.Value describe a mapping for one particular value to a name.

Table 550 shows all attributes of ValueAliasSet.

Table 550 – Attributes of Meas::ValueAliasSet

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 551 shows all association ends of ValueAliasSet with other classes.

Table 551 – Association ends of Meas::ValueAliasSet with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] Discretes	Discrete	The Measurements using the set for translation.
[1..1]	[1..*] Values	ValueToAlias	The ValueToAlias mappings included in the set.
[0..1]	[0..*] Commands	Command	The ValueAliasSet used for translation of a Control value to a name.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.27 ValueToAlias

Describes the translation of one particular value into a name, e.g. 1 as "Open".

Table 552 shows all attributes of ValueToAlias.

Table 552 – Attributes of Meas::ValueToAlias

name	type	description
value	Integer	The value that is mapped.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 553 shows all association ends of ValueToAlias with other classes.

Table 553 – Association ends of Meas::ValueToAlias with other classes

[mult from]	[mult to] name	type	description
[1..*]	[1..1] ValueAliasSet	ValueAliasSet	The ValueAliasSet having the ValueToAlias mappings.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.16 Package SCADA

6.16.1 General

Contains entities to model information used by Supervisory Control and Data Acquisition (SCADA) applications. Supervisory control supports operator control of equipment, such as opening or closing a breaker. Data acquisition gathers telemetered data from various sources. The subtypes of the Telemetry entity deliberately match the UCA and IEC 61850 definitions.

This package also supports alarm presentation but it is not expected to be used by other applications.

Figure 80 shows class diagram Datatypes.



Figure 80 – Class diagram SCADA::Datatypes

Shows datatypes for SCADA model.

Figure 81 shows class diagram Main.

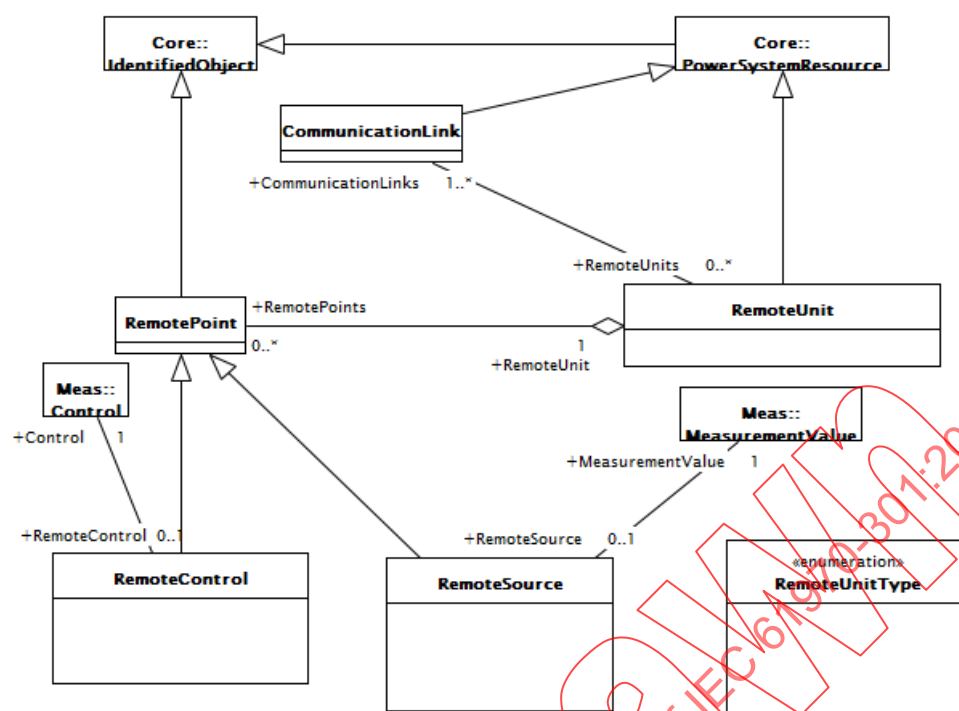


Figure 81 – Class diagram SCADA::Main

This diagram shows all classes included in the SCADA package as well as the key external classes that have associations with SCADA classes.

6.16.2 CommunicationLink

The connection to remote units is through one or more communication links. Redundant links may exist. The CommunicationLink class inherits PowerSystemResource. The intention is to allow CommunicationLinks to have Measurements. These Measurements can be used to model link status as operational, out of service, unit failure, etc.

Table 554 shows all attributes of CommunicationLink.

Table 554 – Attributes of SCADA::CommunicationLink

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 555 shows all association ends of CommunicationLink with other classes.

Table 555 – Association ends of SCADA::CommunicationLink with other classes

[mult from]	[mult to] name	type	description
[1..*]	[0..*] RemoteUnits	RemoteUnit	RTUs may be attached to communication links.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.16.3 RemoteControl

Remote controls are outputs that are sent by the remote unit to actuators in the process.

Table 556 shows all attributes of RemoteControl.

Table 556 – Attributes of SCADA::RemoteControl

name	type	description
remoteControlled	Boolean	Set to true if the actuator is remotely controlled.
actuatorMaximum	Float	The maximum set point value accepted by the remote control point.
actuatorMinimum	Float	The minimum set point value accepted by the remote control point.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 557 shows all association ends of RemoteControl with other classes.

Table 557 – Association ends of SCADA::RemoteControl with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] Control	Control	The Control for the RemoteControl point.
[0..*]	[1..1] RemoteUnit	RemoteUnit	inherited from: RemotePoint
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.16.4 RemotePoint

For a RTU remote points correspond to telemetered values or control outputs. Other units (e.g. control centers) usually also contain calculated values.

Table 558 shows all attributes of RemotePoint.

Table 558 – Attributes of SCADA::RemotePoint

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 559 shows all association ends of RemotePoint with other classes.

Table 559 – Association ends of SCADA::RemotePoint with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] RemoteUnit	RemoteUnit	Remote unit this point belongs to.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.16.5 RemoteSource

Remote sources are state variables that are telemetered or calculated within the remote unit.

Table 560 shows all attributes of RemoteSource.

Table 560 – Attributes of SCADA::RemoteSource

name	type	description
sensorMaximum	Float	The maximum value the telemetry item can return.
sensorMinimum	Float	The minimum value the telemetry item can return.
scanInterval	Seconds	The time interval between scans.
deadband	Float	The smallest change in value to be reported.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 561 shows all association ends of RemoteSource with other classes.

Table 561 – Association ends of SCADA::RemoteSource with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] MeasurementValue	MeasurementValue	Link to the physical telemetered point associated with this measurement.
[0..*]	[1..1] RemoteUnit	RemoteUnit	inherited from: RemotePoint
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.16.6 RemoteUnit

A remote unit can be a RTU, IED, substation control system, control center, etc. The communication with the remote unit can be through various standard protocols (e.g. IEC 61870, IEC 61850) or non standard protocols (e.g. DNP, RP570, etc.). A remote unit contains remote data points that might be telemetered, collected or calculated. The RemoteUnit class inherits from PowerSystemResource. The intention is to allow RemoteUnits to have Measurements. These Measurements can be used to model unit status as operational, out of service, unit failure, etc.

Table 562 shows all attributes of RemoteUnit.

Table 562 – Attributes of SCADA::RemoteUnit

name	type	description
remoteUnitType	RemoteUnitType	Type of remote unit.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 563 shows all association ends of RemoteUnit with other classes.

Table 563 – Association ends of SCADA::RemoteUnit with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..*] CommunicationLinks	CommunicationLink	RTUs may be attached to communication links.
[1..1]	[0..*] RemotePoints	RemotePoint	Remote points this Remote unit contains.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.16.7 RemoteUnitType enumeration

Type of remote unit.

Table 564 shows all literals of RemoteUnitType.

Table 564 – Literals of SCADA::RemoteUnitType

literal	description
RTU	Remote terminal unit.
SubstationControlSystem	Substation control system.
ControlCenter	Control center.
IED	Intelligent electronic device (IED).

6.16.8 Source enumeration

Source gives information related to the origin of a value.

Table 565 shows all literals of Source.

Table 565 – Literals of SCADA::Source

literal	description
PROCESS	The value is provided by input from the process I/O or calculated from some function.
DEFAULTED	The value contains a default value.
SUBSTITUTED	The value is provided by input of an operator or by an automatic source.

6.17 Package ControlArea

6.17.1 General

The ControlArea package models area specifications which can be used for a variety of purposes. The package as a whole models potentially overlapping control area specifications for the purpose of actual generation control, load forecast area load capture, or powerflow based analysis.

Figure 82 shows class diagram ControlArea.

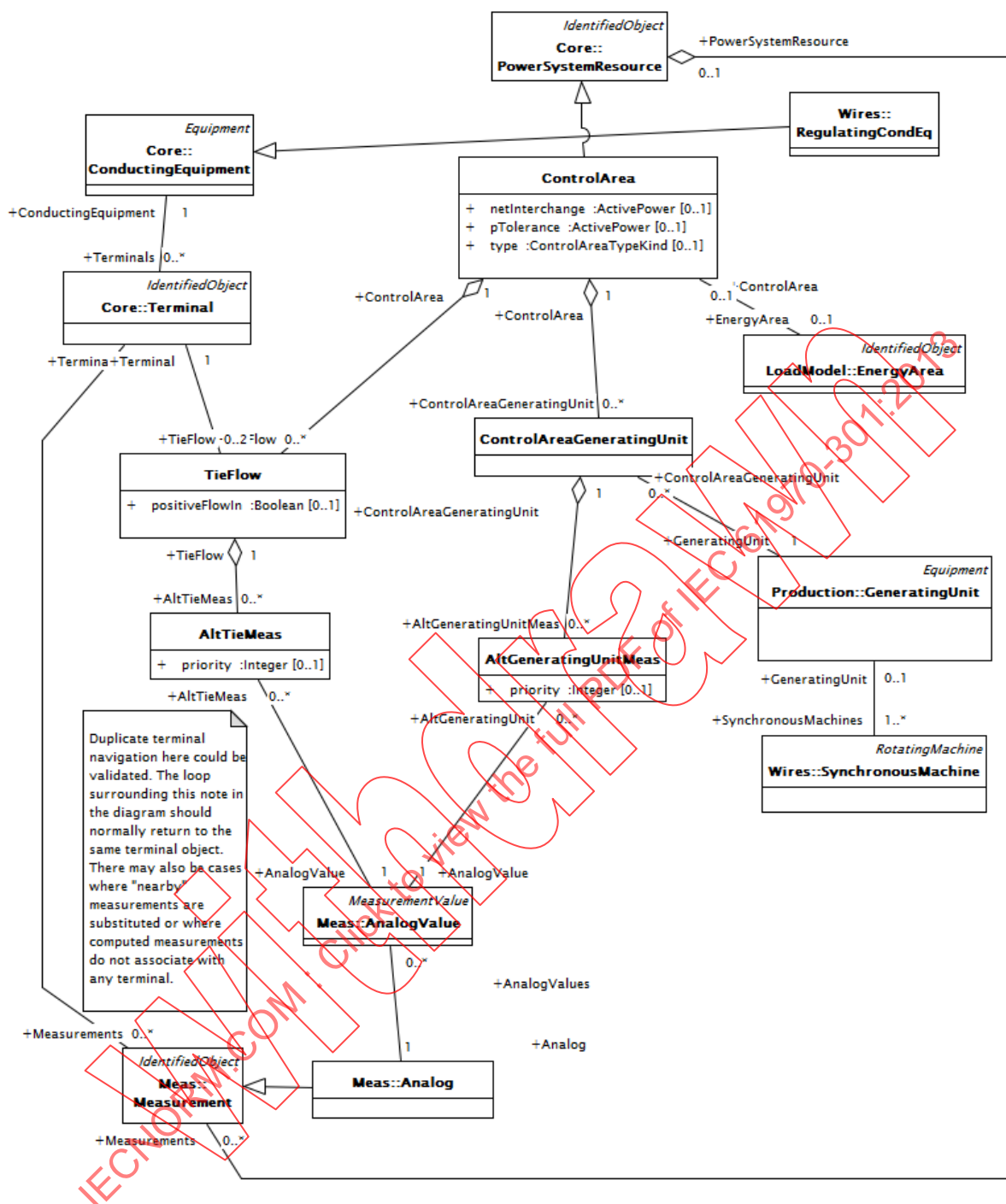


Figure 82 – Class diagram ControlArea::ControlArea

This diagram shows control area specification and some related classes. The Terminal to AnalogValue linkages are shown for clarity in understanding the control area specification. The GeneratingUnit to Terminal linkages are also shown to illustrate how generation flows are specifically tied to the network.

Figure 83 shows class diagram ControlAreaInheritance.

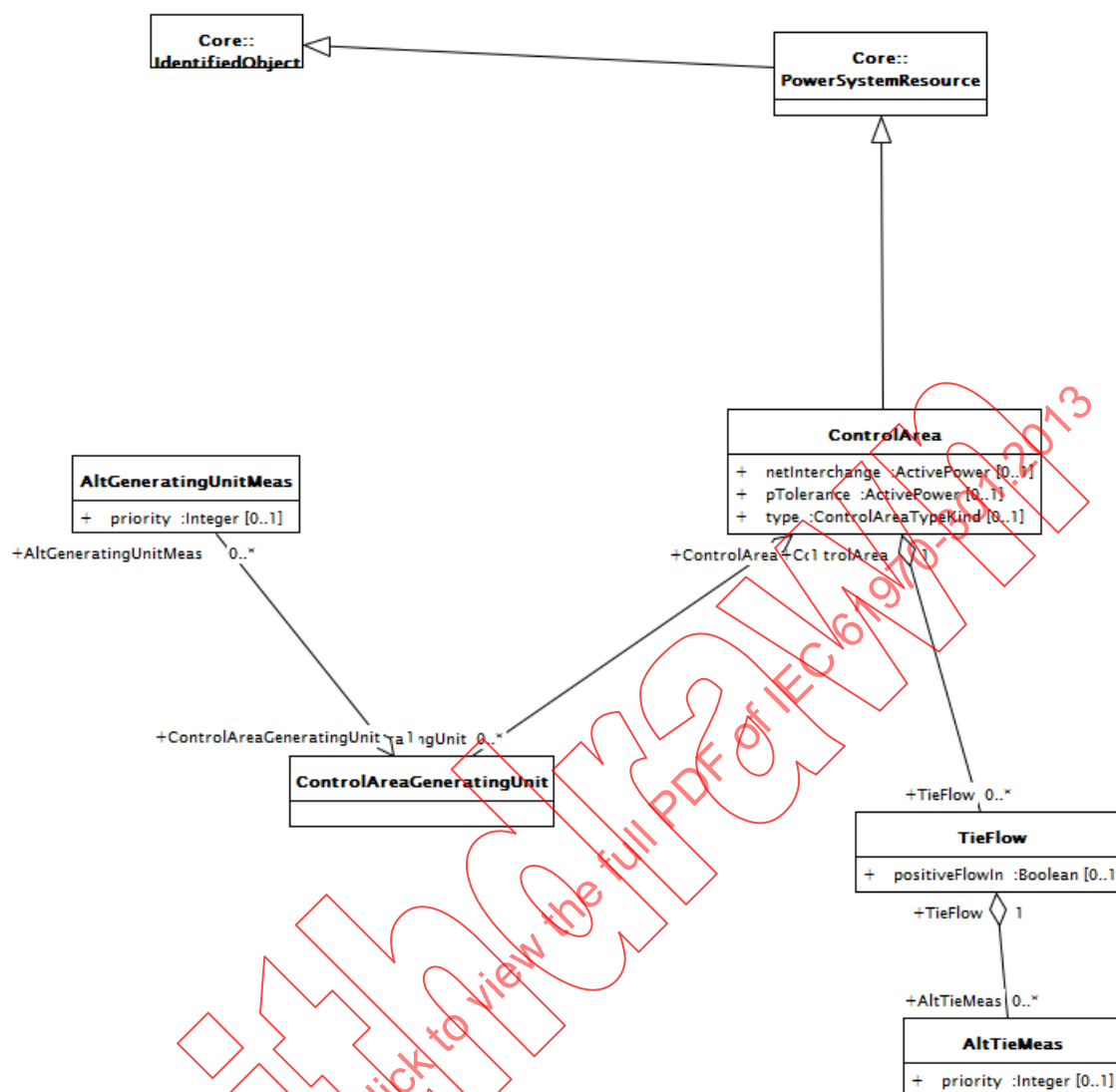


Figure 83 – Class diagram ControlArea::ControlAreaInheritance

Shows the inheritance of classes in this package.

Figure 84 shows class diagram Datatypes.

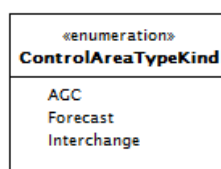


Figure 84 – Class diagram ControlArea::Datatypes

Shows datatypes for control areas.

6.17.2 AltGeneratingUnitMeas root class

A prioritized measurement to be used for the generating unit in the control area specification.

Table 566 shows all attributes of AltGeneratingUnitMeas.

Table 566 – Attributes of ControlArea::AltGeneratingUnitMeas

name	type	description
priority	Integer	Priority of a measurement usage. Lower numbers have first priority.

Table 567 shows all association ends of AltGeneratingUnitMeas with other classes.

Table 567 – Association ends of ControlArea::AltGeneratingUnitMeas with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] AnalogValue	AnalogValue	The specific analog value used as a source.
[0..*]	[1..1] ControlAreaGeneratingUnit	ControlAreaGeneratingUnit	The control area generating unit to which the prioritized measurement assignment is applied.

6.17.3 AltTieMeas root class

A prioritized measurement to be used for the tie flow as part of the control area specification.

Table 568 shows all attributes of AltTieMeas.

Table 568 – Attributes of ControlArea::AltTieMeas

name	type	description
priority	Integer	Priority of a measurement usage. Lower numbers have first priority.

Table 569 shows all association ends of AltTieMeas with other classes.

Table 569 – Association ends of ControlArea::AltTieMeas with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] AnalogValue	AnalogValue	The specific analog value used as a source.
[0..*]	[1..1] TieFlow	TieFlow	The tie flow of the alternate measurements.

6.17.4 ControlArea

A control area is a grouping of generating units and/or loads and a cutset of tie lines (as terminals) which may be used for a variety of purposes including automatic generation control, powerflow solution area interchange control specification, and input to load forecasting. Note that any number of overlapping control area specifications can be superimposed on the physical model.

Table 570 shows all attributes of ControlArea.

Table 570 – Attributes of ControlArea::ControlArea

name	type	description
netInterchange	ActivePower	The specified positive net interchange into the control area.
pTolerance	ActivePower	Active power net interchange tolerance
type	ControlAreaTypeKind	The type of control area definition used to determine if this is used for automatic generation control, for planning interchange control, or other purposes.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 571 shows all association ends of ControlArea with other classes.

Table 571 – Association ends of ControlArea::ControlArea with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..1] EnergyArea	EnergyArea	The energy area that is forecast from this control area specification.
[1..1]	[0..*] ControlAreaGeneratingUnit	ControlAreaGeneratingUnit	The generating unit specifications for the control area.
[1..1]	[0..*] TieFlow	TieFlow	The tie flows associated with the control area.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.17.5 ControlAreaGeneratingUnit root class

A control area generating unit. This class is needed so that alternate control area definitions may include the same generating unit. Note only one instance within a control area should reference a specific generating unit.

Table 572 shows all association ends of ControlAreaGeneratingUnit with other classes.

Table 572 – Association ends of ControlArea::ControlAreaGeneratingUnit with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] GeneratingUnit	GeneratingUnit	The generating unit specified for this control area. Note that a control area should include a GeneratingUnit only once.
[0..*]	[1..1] ControlArea	ControlArea	The parent control area for the generating unit specifications.
[1..1]	[0..*] AltGeneratingUnitMeas	AltGeneratingUnitMeas	The link to prioritized measurements for this GeneratingUnit.

6.17.6 ControlAreaTypeKind enumeration

The type of control area.

Table 573 shows all literals of ControlAreaTypeKind.

Table 573 – Literals of ControlArea::ControlAreaTypeKind

literal	description
AGC	Used for automatic generatoin control.
Forecast	Used for load forecast.
Interchange	Used for interchange specification or control.

6.17.7 TieFlow root class

A flow specification in terms of location and direction for a control area.

Table 574 shows all attributes of TieFlow.

Table 574 – Attributes of ControlArea::TieFlow

name	type	description
positiveFlowIn	Boolean	The flow is positive into the terminal. A flow is positive if it is an import into the control area.

Table 575 shows all association ends of TieFlow with other classes.

Table 575 – Association ends of ControlArea::TieFlow with other classes

[mult from]	[mult to] name	type	description
[0..2]	[1..1] Terminal	Terminal	The terminal to which this tie flow belongs.
[0..*]	[1..1] ControlArea	ControlArea	The control area of the tie flows.
[1..1]	[0..*] AltTieMeas	AltTieMeas	The primary and alternate tie flow measurements associated with the tie flow.

6.18 Package Contingency

6.18.1 General

Contingencies to be studied.

Figure 85 shows class diagram Contingency.

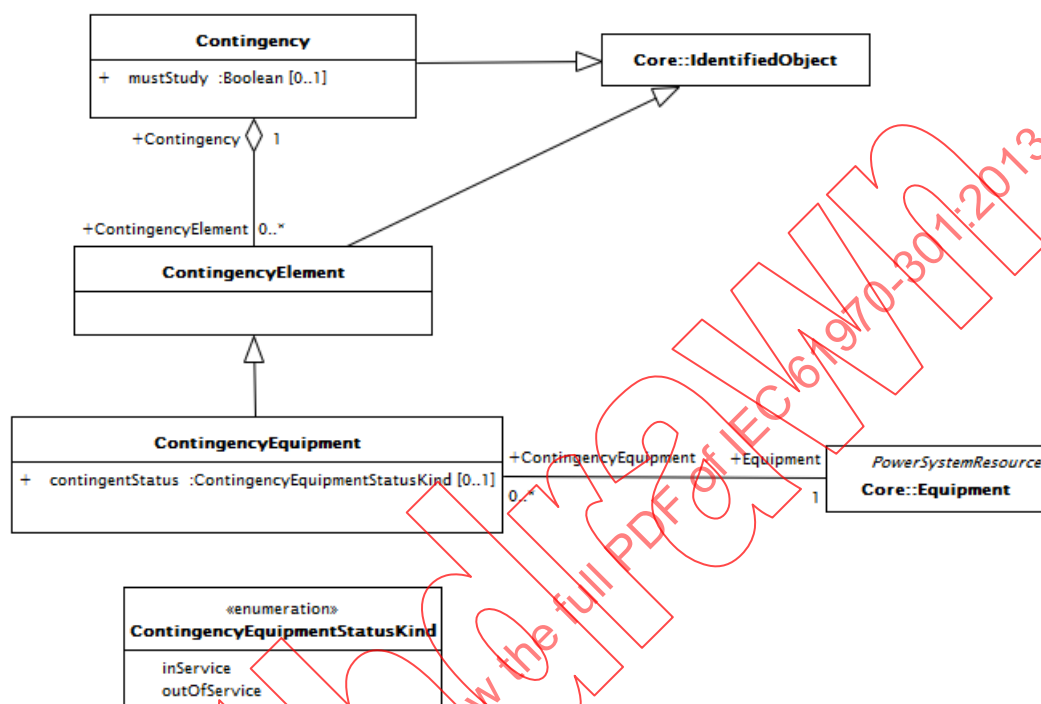


Figure 85 – Class diagram Contingency::Contingency

This diagram shows contingency model inheritance and associations.

6.18.2 Contingency

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

Table 576 shows all attributes of Contingency.

Table 576 – Attributes of Contingency::Contingency

name	type	description
mustStudy	Boolean	Set true if must study this contingency.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 577 shows all association ends of Contingency with other classes.

Table 577 – Association ends of Contingency::Contingency with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] ContingencyElement	ContingencyElement	A contingency can have any number of contingency elements.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.18.3 ContingencyElement

An element of a system event to be studied by contingency analysis, representing a change in status of a single piece of equipment.

Table 578 shows all attributes of ContingencyElement.

Table 578 – Attributes of Contingency::ContingencyElement

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 579 shows all association ends of ContingencyElement with other classes.

Table 579 – Association ends of Contingency::ContingencyElement with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Contingency	Contingency	A contingency element belongs to one contingency.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.18.4 ContingencyEquipment

A equipment to which the in service status is to change such as a power transformer or AC line segment.

Table 580 shows all attributes of ContingencyEquipment.

Table 580 – Attributes of Contingency::ContingencyEquipment

name	type	description
contingentStatus	ContingencyEquipmentStatusKind	The status for the associated equipment when in the contingency state. This status is independent of the case to which the contingency is originally applied, but defines the equipment status when the contingency is applied.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 581 shows all association ends of ContingencyEquipment with other classes.

Table 581 – Association ends of Contingency::ContingencyEquipment with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Equipment	Equipment	The single piece of equipment to which to apply the contingency.
[0..*]	[1..1] Contingency	Contingency	inherited from: ContingencyElement
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.18.5 ContingencyEquipmentStatusKind enumeration

Indicates the state which the contingency equipment is to be in when the contingency is applied.

Table 582 shows all literals of ContingencyEquipmentStatusKind.

Table 582 – Literals of Contingency::ContingencyEquipmentStatusKind

literal	description
inService	The equipment is in service.
outOfService	The equipment is to be taken out of service.

6.19 Package StateVariables

6.19.1 General

State variables for analysis solutions such as powerflow.

Figure 86 shows class diagram StateVariables.

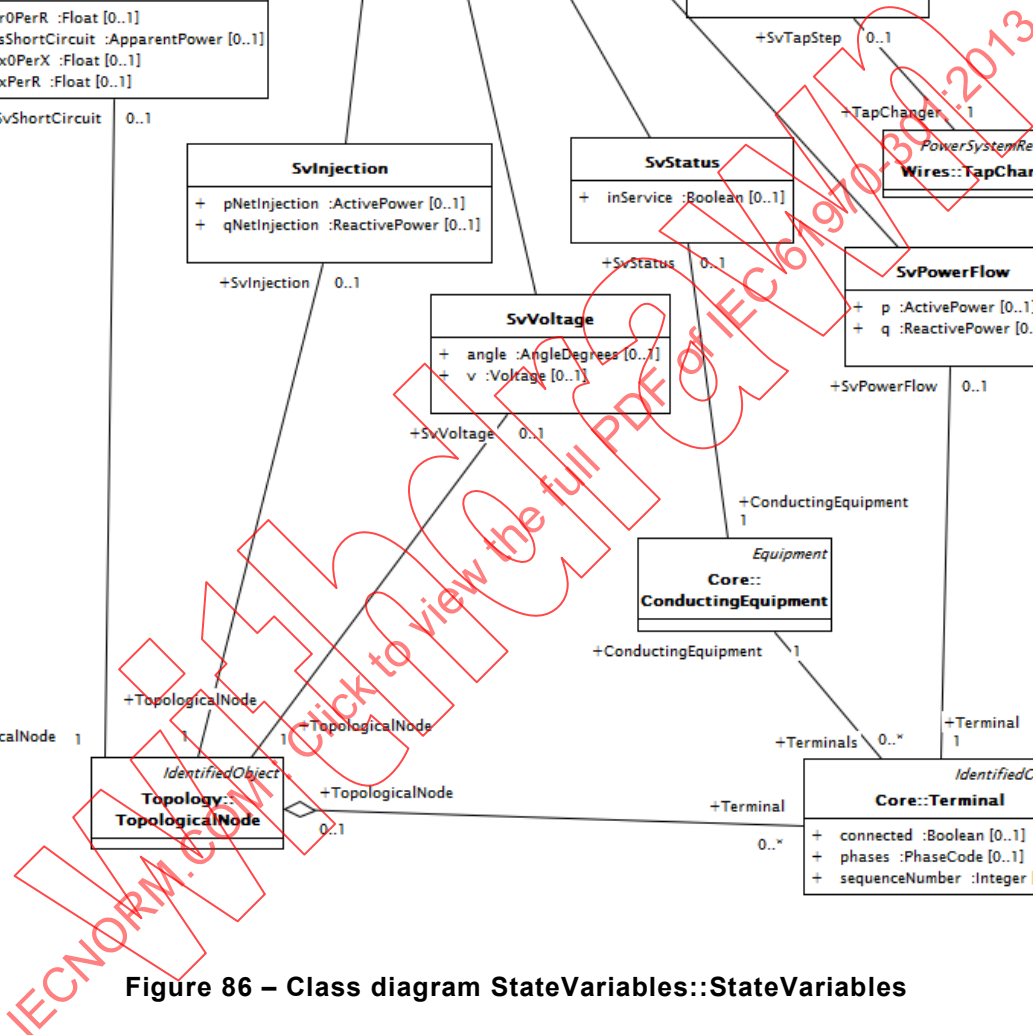


Figure 86 –

The state solution variables as derived from base `StateVariable` class. Note that specialized state variables are associated with objects derived from `IdentifiedObject`, but `StateVariable` does not derive from `IdentifiedObject`.

6.19.2 StateVariable root class

An abstract class for state variables.

6.19.3 SvInjection

Injection state variable. Used for addressing situations where the exchanged models have a Kirchof's law mismatch at a bus. This includes exchange of partial models with boundary flows and state estimator solutions with residual mismatch.

Table 583 shows all attributes of SvInjection.

Table 583 – Attributes of StateVariables::SvInjection

name	type	description
pNetInjection	ActivePower	The active power injected into the bus at this location. Positive sign means injection into the bus.
qNetInjection	ReactivePower	The reactive power injected into the bus at this location. Positive sign means injection into the bus.

Table 584 shows all association ends of SvInjection with other classes.

Table 584 – Association ends of StateVariables::SvInjection with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] TopologicalNode	TopologicalNode	The topological node associated with the flow injection state variable.

6.19.4 SvPowerFlow

State variable for power flow. Load convention is used for flow direction. This means flow out from the TopologicalNode into the equipment is positive.

Table 585 shows all attributes of SvPowerFlow.

Table 585 – Attributes of StateVariables::SvPowerFlow

name	type	description
p	ActivePower	The active power flow. Load sign convention is used, i.e. positive sign means flow out from a node.
q	ReactivePower	The reactive power flow. Load sign convention is used, i.e. positive sign means flow out from a node.

Table 586 shows all association ends of SvPowerFlow with other classes.

Table 586 – Association ends of StateVariables::SvPowerFlow with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] Terminal	Terminal	The terminal associated with the power flow state variable.

6.19.5 SvShortCircuit

State variable for short circuit.

Table 587 shows all attributes of SvShortCircuit.

Table 587 – Attributes of StateVariables::SvShortCircuit

name	type	description
r0PerR	Float	The ratio of zero sequence resistance to positive sequence resistance.
sShortCircuit	ApparentPower	The short circuit apparent power drawn at this node when faulted.
x0PerX	Float	The ratio of zero sequence reactance per positive sequence reactance.
xPerR	Float	Ratio of positive sequence reactance per positive sequence resistance.

Table 588 shows all association ends of SvShortCircuit with other classes.

Table 588 – Association ends of StateVariables::SvShortCircuit with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] TopologicalNode	TopologicalNode	The topological node associated with the state variables for short circuit.

6.19.6 SvShuntCompensatorSections

State variable for the number of sections in service for a shunt compensator.

Table 589 shows all attributes of SvShuntCompensatorSections.

Table 589 – Attributes of StateVariables::SvShuntCompensatorSections

name	type	description
sections	Float	The number of sections in service as a continuous variable. To get integer value scale with ShuntCompensator.bPerSection.

Table 590 shows all association ends of SvShuntCompensatorSections with other classes.

Table 590 – Association ends of StateVariables::SvShuntCompensatorSections with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] ShuntCompensator	ShuntCompensator	The shunt compensator for which the state applies.

6.19.7 SvStatus

State variable for status.

Table 591 shows all attributes of SvStatus.

Table 591 – Attributes of StateVariables::SvStatus

name	type	description
inService	Boolean	The in service status as a result of topology processing.

Table 592 shows all association ends of SvStatus with other classes.

Table 592 – Association ends of StateVariables::SvStatus with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] ConductingEquipment	ConductingEquipment	The conducting equipment associated with the status state variable.

6.19.8 SvTapStep

State variable for transformer tap step. This class is to be used for taps of LTC (load tap changing) transformers, not fixed tap transformers. Normally a profile specifies only one of the attributes "position" or "tapRatio".

Table 593 shows all attributes of SvTapStep.

Table 593 – Attributes of StateVariables::SvTapStep

name	type	description
position	Float	The floating point tap position. To get integer value scale with range and round off.

Table 594 shows all association ends of SvTapStep with other classes.

Table 594 – Association ends of StateVariables::SvTapStep with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] TapChanger	TapChanger	The tap changer associated with the tap step state.

6.19.9 SvVoltage

State variable for voltage.

Table 595 shows all attributes of SvVoltage.

Table 595 – Attributes of StateVariables::SvVoltage

name	type	description
angle	AngleDegrees	The voltage angle of the topological node complex voltage with respect to system reference.
v	Voltage	The voltage magnitude of the topological node.

Table 596 shows all association ends of SvVoltage with other classes.

Table 596 – Association ends of StateVariables::SvVoltage with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] TopologicalNode	TopologicalNode	The topological node associated with the voltage state.

Bibliography

IEC 61968-11, *Application integration at electric utilities – System interfaces for distribution management – Part 11: Common Information Model (CIM) Extensions for Distribution*

IEC 61970-501, *Energy management system application program interface (EMS-API) – Part 501: Common Information Model Resource Description Framework (CIM RDF) schema*

ISO 8601:2004, *Data elements and interchange formats – Information interchange – Representation of dates and times*

RP-3654-1 EPRI Control Center API (CCAPI) research project.

CIMTool – Open source CIM profile management tool. <http://www.cimtool.org>

CIM Users Group – The community of CIM users. <http://cimug.ucaiug.org>

DNP (Distributed Network Protocol): Series of standards. <http://www.dnp.org/>

RP570 (RTU Protocol based on IEC 57 part 5-1 (present IEC 870)).
[http://library.abb.com/GLOBAL/SCOT/scot229.NSF/VerityDisplay/9A5C1896695487E6C2256A7200361578/\\$File/REC501RP570_EN_A.pdf](http://library.abb.com/GLOBAL/SCOT/scot229.NSF/VerityDisplay/9A5C1896695487E6C2256A7200361578/$File/REC501RP570_EN_A.pdf)

W3C: RDF/XML Syntax Specification <http://www.w3.org/RDF/>

W3C: Extensible Markup Language (XML) 1.0 <http://www.w3.org/XML/>

IECNORM.COM : Click to view the full PDF of IEC 61970-301:2013

SOMMAIRE

AVANT-PROPOS	399
INTRODUCTION.....	401
1 Domaine d'application	403
2 Références normatives	403
3 Termes et définitions	404
4 Spécification CIM	405
4.1 Notation de modélisation du CIM	405
4.2 Paquetages CIM	405
4.3 Classes CIM et relations	407
4.3.1 Classes	407
4.3.2 Generalization	408
4.3.3 Simple Association (Association simple)	409
4.3.4 Aggregation	410
4.4 Concepts et exemples de modèles CIM.....	410
4.4.1 Concepts	410
4.4.2 Hiérarchies d'emboîtement, d'équipements et dénomination	411
4.4.3 Modèle de Names.....	413
4.4.4 Modèle de connectivité	414
4.4.5 Hiérarchie des héritages	418
4.4.6 Modèle de transformateur	419
4.4.7 Modélisation de prise du transformateur	422
4.4.8 Modélisation de la phase	426
4.4.9 Modèle de cuts, de serre-fils (clamps) et de cavaliers (jumpers)	427
4.4.10 Mesures et commande.....	431
4.4.11 Modèle de commande de régulation	436
4.5 Conseils de modélisation	437
4.5.1 Modélisation pour modifications	437
4.5.2 Processus pour les amendements au CIM	437
4.5.3 Modifications apportées au modèle UML du CIM	438
4.5.4 Modifications apportées aux documents normatifs du CIM	438
4.5.5 Eléments déconseillés	438
4.5.6 Profils CIM.....	439
4.6 Outils de modélisation.....	439
4.7 Conventions de mise en œuvre pour les utilisateurs	439
4.7.1 Conventions au-delà de l'UML	439
4.7.2 Nombre des Terminal pour les objets ConductingEquipment.....	439
4.8 Exemples de modélisation CIM	440
5 Modèle détaillé	440
5.1 Vue d'ensemble	440
5.2 Contexte	440
6 Paquetage de haut niveau (Top package) IEC61970	442
6.1 Généralités	442
6.2 Classe racine IEC61970CIMVersion.....	443
6.3 Paquetage Domain	444
6.3.1 Généralités	444

6.3.2	Type de données ActivePower	449
6.3.3	Type de données ActivePowerChangeRate	449
6.3.4	Type de données Admittance	449
6.3.5	Type de données AngleDegrees	449
6.3.6	Type de données AngleRadians	450
6.3.7	Type de données ApparentPower	450
6.3.8	Primitive Booléen	450
6.3.9	Type de données Capacitance	450
6.3.10	Type de données Conductance	451
6.3.11	Type de données ConductancePerLength	451
6.3.12	Type de données CostPerEnergyUnit	451
6.3.13	Type de données CostPerVolume	452
6.3.14	Type de données CostRate	452
6.3.15	Enumération Currency	452
6.3.16	Type de données CurrentFlow	453
6.3.17	Type de données Damping	453
6.3.18	Type de données primitif Date	454
6.3.19	Compound DateInterval	454
6.3.20	Type de données primitif DateTime	454
6.3.21	Compound DateTimeInterval	454
6.3.22	Type de données primitif Decimal	454
6.3.23	Type de données Displacement	454
6.3.24	Type de données primitif Duration	455
6.3.25	Primitive Float	455
6.3.26	Type de données FloatQuantity	455
6.3.27	Type de données Frequency	455
6.3.28	Type de données Hours	455
6.3.29	Type de données Impedance	456
6.3.30	Type de données Inductance	456
6.3.31	Primitive Integer	456
6.3.32	Type de données IntegerQuantity	456
6.3.33	Type de données KiloActivePower	457
6.3.34	Type de données Length	457
6.3.35	Type de données Minutes	457
6.3.36	Type de données Money	458
6.3.37	Type de données PU	458
6.3.38	Type de données PerCent	458
6.3.39	Type de données Pressure	458
6.3.40	Type de données Reactance	459
6.3.41	Type de données ReactancePerLength	459
6.3.42	Type de données ReactivePower	459
6.3.43	Type de données RealEnergy	460
6.3.44	Type de données Resistance	460
6.3.45	Type de données ResistancePerLength	460
6.3.46	Type de données RotationSpeed	460
6.3.47	Type de données Seconds	461
6.3.48	Type de données Speed	461
6.3.49	Primitive String	461
6.3.50	Type de données StringQuantity	461

6.3.51	Type de données Susceptance	462
6.3.52	Type de données SusceptancePerLength	462
6.3.53	Type de données Temperature	462
6.3.54	Type de données primitif Time	463
6.3.55	Composé TimeInterval	463
6.3.56	Enumération UnitMultiplier	463
6.3.57	Enumération UnitSymbol	463
6.3.58	Type de données Voltage	464
6.3.59	Type de données VoltagePerReactivePower	465
6.3.60	Type de données Volume	465
6.3.61	Type de données VolumeFlowRate	465
6.3.62	Type de données WaterLevel	465
6.3.63	Type de données Weight	466
6.4	Paquetage Core	466
6.4.1	Généralités	466
6.4.2	BasePower	470
6.4.3	BaseVoltage	471
6.4.4	BasicIntervalSchedule	471
6.4.5	Bay	472
6.4.6	Enumération BreakerConfiguration	473
6.4.7	Enumération BusbarConfiguration	473
6.4.8	ConductingEquipment	474
6.4.9	ConnectivityNode	475
6.4.10	ConnectivityNodeContainer	475
6.4.11	Curve	476
6.4.12	Classe racine CurveData	477
6.4.13	Enumération CurveStyle	477
6.4.14	Equipement	478
6.4.15	EquipmentContainer	479
6.4.16	GeographicalRegion	480
6.4.17	Classe racine IdentifiedObject	480
6.4.18	IrregularIntervalSchedule	481
6.4.19	Classe racine IrregularTimePoint	482
6.4.20	Classe racine Name	482
6.4.21	Classe racine NameType	483
6.4.22	Classe racine NameTypeAuthority	484
6.4.23	OperatingParticipant	484
6.4.24	Classe racine OperatingShare	485
6.4.25	Enumération PhaseCode	485
6.4.26	PowerSystemResource	486
6.4.27	PsrList	487
6.4.28	PSRType	488
6.4.29	RegularIntervalSchedule	488
6.4.30	Classe racine RegularTimePoint	489
6.4.31	ReportingGroup	489
6.4.32	ReportingSuperGroup	490
6.4.33	SubGeographicalRegion	491
6.4.34	Substation	491
6.4.35	Terminal	492

6.4.36	VoltageLevel.....	494
6.5	Package DiagramLayout	495
6.5.1	Généralités	495
6.5.2	Diagram (diagramme)	496
6.5.3	DiagramObject.....	497
6.5.4	Classe racine DiagramObjectGluePoint	498
6.5.5	Classe racine DiagramObjectPoint.....	498
6.5.6	DiagramObjectStyle	499
6.5.7	Enumération OrientationKind	500
6.5.8	TextDiagramObject.....	500
6.5.9	VisibilityLayer	501
6.6	Paquetage OperationalLimits	502
6.6.1	Généralités	502
6.6.2	ActivePowerLimit	503
6.6.3	ApparentPowerLimit.....	503
6.6.4	BranchGroup	504
6.6.5	Classe racine BranchGroupTerminal.....	505
6.6.6	CurrentLimit.....	505
6.6.7	OperationalLimit	506
6.6.8	Enumération OperationalLimitDirectionKind.....	507
6.6.9	OperationalLimitSet	507
6.6.10	OperationalLimitType.....	508
6.6.11	VoltageLimit	509
6.7	Paquetage Topology.....	509
6.7.1	Généralités	509
6.7.2	BusNameMarker.....	512
6.7.3	TopologicalIsland.....	513
6.7.4	TopologicalNode.....	514
6.8	Paquetage Wires.....	515
6.8.1	Généralités	515
6.8.2	ACLineSegment.....	529
6.8.3	ACLineSegmentPhase	531
6.8.4	AsynchronousMachine.....	532
6.8.5	Breaker.....	533
6.8.6	BusbarSection	535
6.8.7	Clamp.....	536
6.8.8	CompositeSwitch	537
6.8.9	Type de données CompositeSwitchType.....	538
6.8.10	Conducteur	538
6.8.11	Connecteur	539
6.8.12	Enumération CoolantType	540
6.8.13	Cut	540
6.8.14	DCLineSegment	542
6.8.15	Disconnecter	543
6.8.16	EnergyConsumer	544
6.8.17	EnergyConsumerPhase	546
6.8.18	EnergySource.....	547
6.8.19	FrequencyConverter	549
6.8.20	Fuse	550

6.8.21	Ground	551
6.8.22	GroundDisconnecter	552
6.8.23	Jumper	553
6.8.24	Junction	554
6.8.25	Line	555
6.8.26	LoadBreakSwitch	556
6.8.27	MutualCoupling	557
6.8.28	Type de données OperatingMode	558
6.8.29	PerLengthImpedance	559
6.8.30	PerLengthPhaseImpedance	559
6.8.31	PerLengthSequenceImpedance	560
6.8.32	Classe racine PhaseImpedanceData	561
6.8.33	Enumération PhaseShuntConnectionKind	561
6.8.34	PhaseTapChanger	562
6.8.35	PhaseTapChangerAsymetrical	563
6.8.36	PhaseTapChangerLinear	565
6.8.37	PhaseTapChangerNonLinear	566
6.8.38	PhaseTapChangerSymetrical	567
6.8.39	PhaseTapChangerTabular	568
6.8.40	Classe racine PhaseTapChangerTabularPoint	569
6.8.41	Plant	570
6.8.42	PowerTransformer	570
6.8.43	PowerTransformerEnd	572
6.8.44	ProtectedSwitch	574
6.8.45	RatioTapChanger	575
6.8.46	RatioTapChangerTabular	577
6.8.47	Classe racine RatioTapChangerTabularPoint	577
6.8.48	ReactiveCapabilityCurve	578
6.8.49	Recloser	579
6.8.50	RectifierInverter	581
6.8.51	RegulatingCondEq	582
6.8.52	RegulatingControl	583
6.8.53	Enumération RegulatingControlModeKind	585
6.8.54	RegulationSchedule	586
6.8.55	Resistor	586
6.8.56	RotatingMachine	587
6.8.57	Sectionaliser	589
6.8.58	SeriesCompensator	590
6.8.59	ShuntCompensator	591
6.8.60	ShuntCompensatorPhase	593
6.8.61	Enumération SinglePhaseKind	594
6.8.62	StaticVarCompensator	595
6.8.63	Enumération SVCControlMode	596
6.8.64	Switch	596
6.8.65	SwitchPhase	597
6.8.66	SwitchSchedule	598
6.8.67	Enumération SynchronousGeneratorType	599
6.8.68	SynchronousMachine	599
6.8.69	Enumération SynchronousMachineOperatingMode	602

6.8.70	Enumération SynchronousMachineType	603
6.8.71	TapChanger	603
6.8.72	TapChangerControl	604
6.8.73	Enumération TapChangerKind	606
6.8.74	TapSchedule	606
6.8.75	Enumération TransformerControlMode	607
6.8.76	TransformerCoreAdmittance	607
6.8.77	TransformerEnd	608
6.8.78	TransformerMeshImpedance	609
6.8.79	TransformerStarImpedance	610
6.8.80	TransformerTank	611
6.8.81	TransformerTankEnd	612
6.8.82	VoltageControlZone	613
6.8.83	Enumération WindingConnection	614
6.9	Paquetage Generation	614
6.9.1	Généralités	614
6.9.2	Paquetage GenerationDynamics	615
6.9.3	Paquetage Production	632
6.10	Paquetage LoadModel	684
6.10.1	Généralités	684
6.10.2	ConformLoad	686
6.10.3	ConformLoadGroup	687
6.10.4	ConformLoadSchedule	688
6.10.5	DayType	689
6.10.6	EnergyArea	689
6.10.7	LoadArea	690
6.10.8	LoadGroup	691
6.10.9	LoadResponseCharacteristic	691
6.10.10	NonConformLoad	692
6.10.11	NonConformLoadGroup	694
6.10.12	NonConformLoadSchedule	695
6.10.13	PowerCutZone	696
6.10.14	Classe racine Season	697
6.10.15	SeasonDayTypeSchedule	697
6.10.16	Enumération SeasonName	698
6.10.17	StationSupply	698
6.10.18	SubLoadArea	700
6.11	Paquetage Outage	700
6.11.1	Généralités	700
6.11.2	ClearanceTag	701
6.11.3	ClearanceTagType	702
6.11.4	OutageSchedule	703
6.11.5	SwitchingOperation	704
6.11.6	Enumération SwitchState	704
6.12	Paquetage AuxiliaryEquipment	705
6.12.1	Généralités	705
6.12.2	AuxiliaryEquipment	705
6.12.3	CurrentTransformer	706
6.12.4	FaultIndicator	707

6.12.5	PostLineSensor	708
6.12.6	PotentialTransformer	709
6.12.7	Sensor	710
6.12.8	SurgeProtector	711
6.13	Paquetage Protection	712
6.13.1	Généralités	712
6.13.2	CurrentRelay	713
6.13.3	ProtectionEquipment	714
6.13.4	RecloseSequence	715
6.13.5	SynchrocheckRelay	716
6.14	Paquetage Equivalents	717
6.14.1	Généralités	717
6.14.2	EquivalentBranch	718
6.14.3	EquivalentEquipment	719
6.14.4	EquivalentInjection	720
6.14.5	EquivalentNetwork	721
6.14.6	EquivalentShunt	722
6.15	Paquetage Meas	723
6.15.1	Généralités	723
6.15.2	Accumulator	727
6.15.3	AccumulatorLimit	727
6.15.4	AccumulatorLimitSet	728
6.15.5	AccumulatorValue	729
6.15.6	Analog	729
6.15.7	AnalogLimit	730
6.15.8	AnalogLimitSet	731
6.15.9	AnalogValue	732
6.15.10	Command	732
6.15.11	Control	733
6.15.12	ControlType	734
6.15.13	Discrete	735
6.15.14	DiscreteValue	736
6.15.15	Limit	737
6.15.16	LimitSet	737
6.15.17	Measurement	738
6.15.18	MeasurementValue	739
6.15.19	MeasurementValueQuality	740
6.15.20	MeasurementValueSource	741
6.15.21	Classe racine Quality61850	741
6.15.22	SetPoint	742
6.15.23	StringMeasurement	743
6.15.24	StringMeasurementValue	744
6.15.25	Enumération Validity	745
6.15.26	ValueAliasSet	745
6.15.27	ValueToAlias	746
6.16	Paquetage SCADA	746
6.16.1	Généralités	746
6.16.2	CommunicationLink	747
6.16.3	RemoteControl	748

6.16.4	RemotePoint.....	749
6.16.5	RemoteSource.....	749
6.16.6	RemoteUnit	750
6.16.7	Enumération RemoteUnitType	751
6.16.8	Enumération Source	751
6.17	Paquetage ControlArea.....	752
6.17.1	Généralités	752
6.17.2	Classe racine AltGeneratingUnitMeas.....	754
6.17.3	Classe racine AltTieMeas	754
6.17.4	ControlArea	755
6.17.5	Classe racine ControlAreaGeneratingUnit.....	756
6.17.6	Enumération ControlAreaTypeKind	756
6.17.7	Classe racine TieFlow	757
6.18	Paquetage Contingency	757
6.18.1	Généralités	757
6.18.2	Contingency	758
6.18.3	ContingencyElement.....	759
6.18.4	ContingencyEquipment.....	759
6.18.5	Enumération ContingencyEquipmentStatusKind.....	760
6.19	Paquetage StateVariables.....	760
6.19.1	Généralités	760
6.19.2	Classe racine StateVariable.....	761
6.19.3	SvInjection.....	762
6.19.4	SvPowerFlow.....	762
6.19.5	SvShortCircuit	763
6.19.6	SvShuntCompensatorSections.....	763
6.19.7	SvStatus.....	764
6.19.8	SvTapStep.....	764
6.19.9	SvVoltage.....	765
	Bibliographie.....	766
	Figure 1 – Exemple de dépendances du paquetage de groupe de travail	406
	Figure 2 – Diagramme du paquetage CIM CEI 61970-301.....	407
	Figure 3 – Exemple de généralisation	409
	Figure 4 – Exemple d'association simple.....	410
	Figure 5 – Exemple d'agrégation.....	410
	Figure 6 – EquipmentContainers.....	412
	Figure 7 – Names	413
	Figure 8 – Modèle de connectivité.....	415
	Figure 9 – Exemple de réseau simple	417
	Figure 10 – Connectivité d'un réseau simple modélisé avec la topologie du CIM.....	418
	Figure 11 – Hiérarchie des héritages de Equipment.....	419
	Figure 12 – Modèle de Transformer et de Tank.....	420
	Figure 13 – Modèle de transformateur	422
	Figure 14 – Modèle de prise de transformateur	423
	Figure 15 – Déphaseur symétrique	424
	Figure 16 – Déphaseur asymétrique	425

Figure 17 – Connectivité de phase	427
Figure 18 – Modèle de cuts, de serre-fils et de cavaliers de l'UML	428
Figure 19 – Exemple avant d'appliquer des cuts et des cavaliers	429
Figure 20 – Exemple après que des cuts et des cavaliers ont été appliqués	430
Figure 21 – Exemple de cavalier sans cut ni serre-fil	431
Figure 22 – Navigation de PSR à MeasurementValue	433
Figure 23 – Placement de Measurement	436
Figure 24 – Modèles de commande de régulation	437
Figure 25 – Paquetages de haut niveau du CIM	441
Figure 26 – Diagramme de classe IEC61970::IEC61970Dependencies	443
Figure 27 – Diagramme de classe Domain::BasicDatatypes	444
Figure 28 – Diagramme de classe Domain::ElectricityDatatypes	445
Figure 29 – Diagramme de classe Domain::EnumeratedUnitDatatype	446
Figure 30 – Diagramme de classe Domain::GeneralDatatypes	447
Figure 31 – Diagramme de classe Domain::MonetaryDatatypes	448
Figure 32 – Diagramme de classe Domain::TimeDatatypes	448
Figure 33 – Diagramme de classe Core::Main	467
Figure 34 – Diagramme de classe Core::Names	468
Figure 35 – Diagramme de classe Core::CurveSchedule	468
Figure 36 – Diagramme de classe Core::Datatypes	469
Figure 37 – Diagramme de classe Core::Reporting	469
Figure 38 – Diagramme de classe Core::OperatingShare	470
Figure 39 – Diagramme de classe DiagramLayout::DiagramLayout	496
Figure 40 – Diagramme de classe OperationalLimits::OperationalLimits	502
Figure 41 – Diagramme de classe OperationalLimits::BranchGroup	503
Figure 42 – Diagramme de classe Topology::TopologicalNodeTerminal	510
Figure 43 – Diagramme de classe Topology::TopologyMeasRelations	511
Figure 44 – Diagramme de classe Topology::TopologyReporting	512
Figure 45 – Diagramme de classe Topology::Main	512
Figure 46 – Diagramme de classe Wires::CutsAndJumpers	516
Figure 47 – Diagramme de classe Wires::MutualCoupling	517
Figure 48 – Diagramme de classe Wires::Schedules	518
Figure 49 – Diagramme de classe Wires::SwitchingEquipment	519
Figure 50 – Diagramme de classe Wires::WiresPhaseModel	520
Figure 51 – Diagramme de classe Wires::Datatypes	521
Figure 52 – Diagramme de classe Wires::InheritanceHierarchy	522
Figure 53 – Diagramme de classe Wires::LineModel	523
Figure 54 – Diagramme de classe Wires::NamingHierarchyPart1	524
Figure 55 – Diagramme de classe Wires::NamingHierarchyPart2	525
Figure 56 – Diagramme de classe Wires::RegulatingEquipment	526
Figure 57 – Diagramme de classe Wires::TapChanger	527
Figure 58 – Diagramme de classe Wires::VoltageControl	528
Figure 59 – Diagramme de classe Wires::Transformer	529

Figure 60 – Diagramme de classe Generation::Main	614
Figure 61 – Diagramme de classe GenerationDynamics::Main	615
Figure 62 – Diagramme de classe GenerationDynamics::Datatypes	615
Figure 63 – Diagramme de classe Production::Nuclear	633
Figure 64 – Diagramme de classe Production::Main	634
Figure 65 – Diagramme de classe Production::Datatypes	635
Figure 66 – Diagramme de classe Production::Hydro	636
Figure 67 – Diagramme de classe Production::Thermal	637
Figure 68 – Diagramme de classe LoadModel::Main	685
Figure 69 – Diagramme de classe LoadModel::Datatypes	685
Figure 70 – Diagramme de classe Outage::Datatypes	700
Figure 71 – Diagramme de classe Outage::Main	701
Figure 72 – Diagramme de classe AuxiliaryEquipment::AuxiliaryEquipment	705
Figure 73 – Diagramme de classe Protection::Main	712
Figure 74 – Diagramme de classe Equivalents::Main	718
Figure 75 – Diagramme de classe Meas::Datatypes	723
Figure 76 – Diagramme de classe Meas::Control	724
Figure 77 – Diagramme de classe Meas::Measurement	725
Figure 78 – Diagramme de classe Meas::MeasurementInheritance	726
Figure 79 – Diagramme de classe Meas::Quality	726
Figure 80 – Diagramme de classe SCADA::Datatypes	747
Figure 81 – Diagramme de classe SCADA::Main	747
Figure 82 – Diagramme de classe ControlArea::ControlArea	753
Figure 83 – Diagramme de classe ControlArea::ControlAreaInheritance	753
Figure 84 – Diagramme de classe ControlArea::Datatypes	754
Figure 85 – Diagramme de classe Contingency::Contingency	758
Figure 86 – Diagramme de classe StateVariables::StateVariables	761
Tableau 1 – Conventions de dénomination de la classe NameType (type de nom)	414
Tableau 2 – Conventions de dénomination de measurementType	434
Tableau 3 – Conventions de dénomination de MeasurementValueSource (source de valeur de mesure)	435
Tableau 4 – Attributs de Package1::Class1	441
Tableau 5 – Extrémités d'association de Package1::Class1 avec d'autres classes	442
Tableau 6 – Libellés de Package1::Enumeration1	442
Tableau 7 – Attributs de IEC61970::IEC61970CIMVersion	444
Tableau 8 – Attributs de Domain::ActivePower	449
Tableau 9 – Attributs de Domain::ActivePowerChangeRate	449
Tableau 10 – Attributs de Domain::Admittance	449
Tableau 11 – Attributs de Domain::AngleDegrees	450
Tableau 12 – Attributs de Domain::AngleRadians	450
Tableau 13 – Attributs de Domain::ApparentPower	450
Tableau 14 – Attributs de Domain::Capacitance	451

Tableau 15 – Attributs de Domain::Conductance.....	451
Tableau 16 – Attributs de Domain::ConductancePerLength.....	451
Tableau 17 – Attributs de Domain::CostPerEnergyUnit	452
Tableau 18 – Attributs de Domain::CostPerVolume.....	452
Tableau 19 – Attributs de Domain::CostRate.....	452
Tableau 20 – Libellés de Domain::Currency.....	453
Tableau 21 – Attributs de Domain::CurrentFlow	453
Tableau 22 – Attributs de Domain::Damping	453
Tableau 23 – Attributs de Domain::DateInterval	454
Tableau 24 – Attributs de Domain::DateTimeInterval	454
Tableau 25 – Attributs de Domain::Displacement.....	455
Tableau 26 – Attributs de Domain::FloatQuantity	455
Tableau 27 – Attributs de Domain::Frequency.....	455
Tableau 28 – Attributs de Domain::Hours.....	456
Tableau 29 – Attributs de Domain::Impedance.....	456
Tableau 30 – Attributs de Domain::Inductance.....	456
Tableau 31 – Attributs de Domain::IntegerQuantity.....	457
Tableau 32 – Attributs de Domain::KiloActivePower.....	457
Tableau 33 – Attributs de Domain::Length.....	457
Tableau 34 – Attributs de Domain::Minutes.....	457
Tableau 35 – Attributs de Domain::Money.....	458
Tableau 36 – Attributs de Domain::PU.....	458
Tableau 37 – Attributs de Domain::PerCent.....	458
Tableau 38 – Attributs de Domain::Pressure.....	459
Tableau 39 – Attributs de Domain::Reactance.....	459
Tableau 40 – Attributs de Domain::ReactancePerLength.....	459
Tableau 41 – Attributs de Domain::ReactivePower.....	459
Tableau 42 – Attributs de Domain::RealEnergy.....	460
Tableau 43 – Attributs de Domain::Resistance.....	460
Tableau 44 – Attributs de Domain::ResistancePerLength.....	460
Tableau 45 – Attributs de Domain::RotationSpeed	461
Tableau 46 – Attributs de Domain::Seconds.....	461
Tableau 47 – Attributs de Domain::Speed	461
Tableau 48 – Attributs de Domain::StringQuantity	462
Tableau 49 – Attributs de Domain::Susceptance.....	462
Tableau 50 – Attributs de Domain::SusceptancePerLength	462
Tableau 51 – Attributs de Domain::Temperature	462
Tableau 52 – Attributs de Domain::TimeInterval.....	463
Tableau 53 – Libellés de Domain::UnitMultiplier.....	463
Tableau 54 – Libellés de Domain::UnitSymbol	464
Tableau 55 – Attributs de Domain::Voltage	464
Tableau 56 – Attributs de Domain::VoltagePerReactivePower.....	465
Tableau 57 – Attributs de Domain::Volume	465

Tableau 58 – Attributs de Domain::VolumeFlowRate	465
Tableau 59 – Attributs de Domain::WaterLevel	466
Tableau 60 – Attributs de Domain::Weight	466
Tableau 61 – Attributs de Core::BasePower	470
Tableau 62 – Extrémités d'association de Core::BasePower avec d'autres classes	470
Tableau 63 – Attributs de Core::BaseVoltage	471
Tableau 64 – Extrémités d'association de Core::BaseVoltage avec d'autres classes	471
Tableau 65 – Attributs de Core::BasicIntervalSchedule	472
Tableau 66 – Extrémités d'association de Core::BasicIntervalSchedule avec d'autres classes	472
Tableau 67 – Attributs de Core::Bay	472
Tableau 68 – Extrémités d'association de Core::Bay avec d'autres classes	473
Tableau 69 – Libellés de Core::BreakerConfiguration	473
Tableau 70 – Libellés de Core::BusbarConfiguration	473
Tableau 71 – Attributs de Core::ConductingEquipment	474
Tableau 72 – Extrémités d'association de Core::ConductingEquipment avec d'autres classes	474
Tableau 73 – Attributs de Core::ConnectivityNode	475
Tableau 74 – Extrémités d'association de Core::ConnectivityNode avec d'autres classes	475
Tableau 75 – Attributs de Core::ConnectivityNodeContainer	475
Tableau 76 – Extrémités d'association de Core::ConnectivityNodeContainer avec d'autres classes	476
Tableau 77 – Attributs de Core::Curve	476
Tableau 78 – Extrémités d'association de Core::Curve avec d'autres classes	477
Tableau 79 – Attributs de Core::CurveData	477
Tableau 80 – Extrémités d'association de Core::CurveData avec d'autres classes	477
Tableau 81 – Libellés de Core::CurveStyle	478
Tableau 82 – Attributs de Core::Equipment	478
Tableau 83 – Extrémités d'association de Core::Equipment avec d'autres classes	479
Tableau 84 – Attributs de Core::EquipmentContainer	479
Tableau 85 – Extrémités d'association de Core::EquipmentContainer avec d'autres classes	480
Tableau 86 – Attributs de Core::GeographicalRegion	480
Tableau 87 – Extrémités d'association de Core::GeographicalRegion avec d'autres classes	480
Tableau 88 – Attributs de Core::IdentifiedObject	481
Tableau 89 – Extrémités d'association de Core::IdentifiedObject avec d'autres classes	481
Tableau 90 – Attributs de Core::IrregularIntervalSchedule	481
Tableau 91 – Extrémités d'association de Core::IrregularIntervalSchedule avec d'autres classes	482
Tableau 92 – Attributs de Core::IrregularTimePoint	482
Tableau 93 – Extrémités d'association de Core::IrregularTimePoint avec d'autres classes	482
Tableau 94 – Attributs de Core::Name	483
Tableau 95 – Extrémités d'association de Core::Name avec d'autres classes	483

Tableau 96 – Attributs de Core::NameType.....	483
Tableau 97 – Extrémités d'association de Core::NameType avec d'autres classes.....	483
Tableau 98 – Attributs de Core::NameTypeAuthority.....	484
Tableau 99 – Extrémités d'association de Core::NameTypeAuthority avec d'autres classes.....	484
Tableau 100 – Attributs de Core::OperatingParticipant.....	484
Tableau 101 – Extrémités d'association de Core::OperatingParticipant avec d'autres classes.....	485
Tableau 102 – Attributs de Core::OperatingShare.....	485
Tableau 103 – Extrémités d'association de Core::OperatingShare avec d'autres classes.....	485
Tableau 104 – Libellés de Core::PhaseCode.....	486
Tableau 105 – Attributs de Core::PowerSystemResource.....	486
Tableau 106 – Extrémités d'association de Core::PowerSystemResource avec d'autres classes.....	487
Tableau 107 – Attributs de Core::PsrList.....	487
Tableau 108 – Extrémités d'association de Core::PsrList avec d'autres classes.....	487
Tableau 109 – Attributs de Core::PSRType.....	488
Tableau 110 – Extrémités d'association de Core::PSRType avec d'autres classes.....	488
Tableau 111 – Attributs de Core::RegularIntervalSchedule.....	488
Tableau 112 – Extrémités d'association de Core::RegularIntervalSchedule avec d'autres classes.....	489
Tableau 113 – Attributs de Core::RegularTimePoint.....	489
Tableau 114 – Extrémités d'association de Core::RegularTimePoint avec d'autres classes.....	489
Tableau 115 – Attributs de Core::ReportingGroup.....	490
Tableau 116 – Extrémités d'association de Core::ReportingGroup avec d'autres classes.....	490
Tableau 117 – Attributs de Core::ReportingSuperGroup.....	490
Tableau 118 – Extrémités d'association de Core::ReportingSuperGroup avec d'autres classes.....	491
Tableau 119 – Attributs de Core::SubGeographicalRegion.....	491
Tableau 120 – Extrémités d'association de Core::SubGeographicalRegion avec d'autres classes.....	491
Tableau 121 – Attributs de Core::Substation.....	492
Tableau 122 – Extrémités d'association de Core::Substation avec d'autres classes.....	492
Tableau 123 – Attributs de Core::Terminal.....	493
Tableau 124 – Extrémités d'association de Core::Terminal avec d'autres classes.....	493
Tableau 125 – Attributs de Core::VoltageLevel.....	494
Tableau 126 – Extrémités d'association de Core::VoltageLevel avec d'autres classes.....	495
Tableau 127 – Attributs de DiagramLayout::Diagram.....	496
Tableau 128 – Extrémités d'association de DiagramLayout::Diagram avec d'autres classes.....	497
Tableau 129 – Attributs de DiagramLayout::DiagramObject.....	497
Tableau 130 – Extrémités d'association de DiagramLayout::DiagramObject avec d'autres classes.....	498

Tableau 131 – Extrémités d'association de DiagramLayout::DiagramObjectGluePoint avec d'autres classes.....	498
Tableau 132 – Attributs de DiagramLayout::DiagramObjectPoint	499
Tableau 133 – Extrémités d'association de DiagramLayout::DiagramObjectPoint avec d'autres classes.....	499
Tableau 134 – Attributs de DiagramLayout::DiagramObjectStyle.....	499
Tableau 135 – Extrémités d'association de DiagramLayout::DiagramObjectStyle avec d'autres classes.....	500
Tableau 136 – Libellés de DiagramLayout::OrientationKind	500
Tableau 137 – Attributs de DiagramLayout::TextDiagramObject.....	500
Tableau 138 – Extrémités d'association de DiagramLayout::TextDiagramObject avec d'autres classes.....	501
Tableau 139 – Attributs de DiagramLayout::VisibilityLayer.....	501
Tableau 140 – Extrémités d'association de DiagramLayout::VisibilityLayer avec d'autres classes.....	502
Tableau 141 – Attributs de OperationalLimits::ActivePowerLimit.....	503
Tableau 142 – Extrémités d'association de OperationalLimits::ActivePowerLimit avec d'autres classes.....	503
Tableau 143 – Attributs de OperationalLimits::ApparentPowerLimit.....	504
Tableau 144 – Extrémités d'association de OperationalLimits::ApparentPowerLimit avec d'autres classes.....	504
Tableau 145 – Attributs de OperationalLimits::BranchGroup	504
Tableau 146 – Extrémités d'association de OperationalLimits::BranchGroup avec d'autres classes.....	505
Tableau 147 – Attributs de OperationalLimits::BranchGroupTerminal.....	505
Tableau 148 – Extrémités d'association de OperationalLimits::BranchGroupTerminal avec d'autres classes.....	505
Tableau 149 – Attributs de OperationalLimits::CurrentLimit.....	506
Tableau 150 – Extrémités d'association de OperationalLimits::CurrentLimit avec d'autres classes.....	506
Tableau 151 – Attributs de OperationalLimits::OperationalLimit	506
Tableau 152 – Extrémités d'association de OperationalLimits::OperationalLimit avec d'autres classes.....	506
Tableau 153 – Libellés de OperationalLimits::OperationalLimitDirectionKind.....	507
Tableau 154 – Attributs de OperationalLimits::OperationalLimitSet	507
Tableau 155 – Extrémités d'association de OperationalLimits::OperationalLimitSet avec d'autres classes	508
Tableau 156 – Attributs de OperationalLimits::OperationalLimitType.....	508
Tableau 157 – Extrémités d'association de OperationalLimits::OperationalLimitType avec d'autres classes	508
Tableau 158 – Attributs de OperationalLimits::VoltageLimit.....	509
Tableau 159 – Extrémités d'association de OperationalLimits::VoltageLimit avec d'autres classes.....	509
Tableau 160 – Attributs de Topology::BusNameMarker.....	513
Tableau 161 – Extrémités d'association de Topology::BusNameMarker avec d'autres classes	513
Tableau 162 – Attributs de Topology::TopologicalIsland	513

Tableau 163 – Extrémités d'association de Topology::TopologicalIsland avec d'autres classes	514
Tableau 164 – Attributs de Topology::TopologicalNode.....	514
Tableau 165 – Extrémités d'association de Topology::TopologicalNode avec d'autres classes	515
Tableau 166 – Attributs de Wires::ACLineSegment	530
Tableau 167 – Extrémités d'association de Wires::ACLineSegment avec d'autres classes	531
Tableau 168 – Attributs de Wires::ACLineSegmentPhase	531
Tableau 169 – Extrémités d'association de Wires::ACLineSegmentPhase avec d'autres classes	532
Tableau 170 – Attributs de Wires::AsynchronousMachine	532
Tableau 171 – Extrémités d'association de Wires::AsynchronousMachine avec d'autres classes	533
Tableau 172 – Attributs de Wires::Breaker	534
Tableau 173 – Extrémités d'association de Wires::Breaker avec d'autres classes	534
Tableau 174 – Attributs de Wires::BusbarSection	535
Tableau 175 – Extrémités d'association de Wires::BusbarSection avec d'autres classes	535
Tableau 176 – Attributs de Wires::Clamp	536
Tableau 177 – Extrémités d'association de Wires::Clamp avec d'autres classes	536
Tableau 178 – Attributs de Wires::CompositeSwitch	537
Tableau 179 – Extrémités d'association de Wires::CompositeSwitch avec d'autres classes	537
Tableau 180 – Attributs de Wires::CompositeSwitchType.....	538
Tableau 181 – Attributs de Wires::Conductor	538
Tableau 182 – Extrémités d'association de Wires::Conductor avec d'autres classes	539
Tableau 183 – Attributs de Wires::Connector	539
Tableau 184 – Extrémités d'association de Wires::Connector avec d'autres classes	540
Tableau 185 – Libellés de Wires::CoolantType	540
Tableau 186 – Attributs de Wires::Cut.....	541
Tableau 187 – Extrémités d'association de Wires::Cut avec d'autres classes	542
Tableau 188 – Attributs de Wires::DCLineSegment.....	542
Tableau 189 – Extrémités d'association de Wires::DCLineSegment avec d'autres classes	543
Tableau 190 – Attributs de Wires::Disconnecter	543
Tableau 191 – Extrémités d'association de Wires::Disconnecter avec d'autres classes	544
Tableau 192 – Attributs de Wires::EnergyConsumer	545
Tableau 193 – Extrémités d'association de Wires::EnergyConsumer avec d'autres classes	546
Tableau 194 – Attributs de Wires::EnergyConsumerPhase.....	547
Tableau 195 – Extrémités d'association de Wires::EnergyConsumerPhase avec d'autres classes	547
Tableau 196 – Attributs de Wires::EnergySource	548
Tableau 197 – Extrémités d'association de Wires::EnergySource avec d'autres classes	548
Tableau 198 – Attributs de Wires::FrequencyConverter	549

Tableau 199 – Extrémités d'association de Wires::FrequencyConverter avec d'autres classes	550
Tableau 200 – Attributs de Wires::Fuse	550
Tableau 201 – Extrémités d'association de Wires::Fuse avec d'autres classes	551
Tableau 202 – Attributs de Wires::Ground.....	551
Tableau 203 – Extrémités d'association de Wires::Ground avec d'autres classes.....	552
Tableau 204 – Attributs de Wires::GroundDisconnecter	552
Tableau 205 – Extrémités d'association de Wires::GroundDisconnecter avec d'autres classes	553
Tableau 206 – Attributs de Wires::Jumper.....	554
Tableau 207 – Extrémités d'association de Wires::Jumper avec d'autres classes.....	554
Tableau 208 – Attributs de Wires::Junction	555
Tableau 209 – Extrémités d'association de Wires::Junction avec d'autres classes	555
Tableau 210 – Attributs de Wires::Line	555
Tableau 211 – Extrémités d'association de Wires::Line avec d'autres classes.....	556
Tableau 212 – Attributs de Wires::LoadBreakSwitch	556
Tableau 213 – Extrémités d'association de Wires::LoadBreakSwitch avec d'autres classes	557
Tableau 214 – Attributs de Wires::MutualCoupling	558
Tableau 215 – Extrémités d'association de Wires::MutualCoupling avec d'autres classes	558
Tableau 216 – Attributs de Wires::OperatingMode	559
Tableau 217 – Attributs de Wires::PerLengthImpedance	559
Tableau 218 – Extrémités d'association de Wires::PerLengthImpedance avec d'autres classes	559
Tableau 219 – Attributs de Wires::PerLengthPhaseImpedance	560
Tableau 220 – Extrémités d'association de Wires::PerLengthPhaseImpedance avec d'autres classes.....	560
Tableau 221 – Attributs de Wires::PerLengthSequenceImpedance.....	560
Tableau 222 – Extrémités d'association de Wires::PerLengthSequenceImpedance avec d'autres classes.....	561
Tableau 223 – Attributs de Wires::PhaseImpedanceData	561
Tableau 224 – Extrémités d'association de Wires::PhaseImpedanceData avec d'autres classes	561
Tableau 225 – Libellés de Wires::PhaseShuntConnectionKind	562
Tableau 226 – Attributs de Wires::PhaseTapChanger	562
Tableau 227 – Extrémités d'association de Wires::PhaseTapChanger avec d'autres classes	563
Tableau 228 – Attributs de Wires::PhaseTapChangerAsymetrical	564
Tableau 229 – Extrémités d'association de Wires::PhaseTapChangerAsymetrical avec d'autres classes.....	564
Tableau 230 – Attributs de Wires::PhaseTapChangerLinear	565
Tableau 231 – Extrémités d'association de Wires::PhaseTapChangerLinear avec d'autres classes.....	566
Tableau 232 – Attributs de Wires::PhaseTapChangerNonLinear	566
Tableau 233 – Extrémités d'association de Wires::PhaseTapChangerNonLinear avec d'autres classes.....	567

Tableau 234 – Attributs de Wires::PhaseTapChangerSymetrical	567
Tableau 235 – Extrémités d'association de Wires::PhaseTapChangerSymetrical avec d'autres classes	568
Tableau 236 – Attributs de Wires::PhaseTapChangerTabular	568
Tableau 237 – Extrémités d'association de Wires::PhaseTapChangerTabular avec d'autres classes	569
Tableau 238 – Attributs de Wires::PhaseTapChangerTabularPoint	569
Tableau 239 – Extrémités d'association de Wires::PhaseTapChangerTabularPoint avec d'autres classes	570
Tableau 240 – Attributs de Wires::Plant	570
Tableau 241 – Extrémités d'association de Wires::Plant avec d'autres classes	570
Tableau 242 – Attributs de Wires::PowerTransformer	571
Tableau 243 – Extrémités d'association de Wires::PowerTransformer avec d'autres classes	572
Tableau 244 – Attributs de Wires::PowerTransformerEnd	573
Tableau 245 – Extrémités d'association de Wires::PowerTransformerEnd avec d'autres classes	574
Tableau 246 – Attributs de Wires::ProtectedSwitch	574
Tableau 247 – Extrémités d'association de Wires::ProtectedSwitch avec d'autres classes	575
Tableau 248 – Attributs de Wires::RatioTapChanger	576
Tableau 249 – Extrémités d'association de Wires::RatioTapChanger avec d'autres classes	576
Tableau 250 – Attributs de Wires::RatioTapChangerTabular	577
Tableau 251 – Extrémités d'association de Wires::RatioTapChangerTabular avec d'autres classes	577
Tableau 252 – Attributs de Wires::RatioTapChangerTabularPoint	577
Tableau 253 – Extrémités d'association de Wires::RatioTapChangerTabularPoint avec d'autres classes	578
Tableau 254 – Attributs de Wires::ReactiveCapabilityCurve	579
Tableau 255 – Extrémités d'association de Wires::ReactiveCapabilityCurve avec d'autres classes	579
Tableau 256 – Attributs de Wires::Recloser	580
Tableau 257 – Extrémités d'association de Wires::Recloser avec d'autres classes	580
Tableau 258 – Attributs de Wires::RectifierInverter	581
Tableau 259 – Extrémités d'association de Wires::RectifierInverter avec d'autres classes	582
Tableau 260 – Attributs de Wires::RegulatingCondEq	582
Tableau 261 – Extrémités d'association de Wires::RegulatingCondEq avec d'autres classes	583
Tableau 262 – Attributs de Wires::RegulatingControl	584
Tableau 263 – Extrémités d'association de Wires::RegulatingControl avec d'autres classes	585
Tableau 264 – Libellés de Wires::RegulatingControlModeKind	585
Tableau 265 – Attributs de Wires::RegulationSchedule	586
Tableau 266 – Extrémités d'association de Wires::RegulationSchedule avec d'autres classes	586

Tableau 267 – Attributs de Wires::Resistor	587
Tableau 268 – Extrémités d'association de Wires::Resistor avec d'autres classes	587
Tableau 269 – Attributs de Wires::RotatingMachine	588
Tableau 270 – Extrémités d'association de Wires::RotatingMachine avec d'autres classes	589
Tableau 271 – Attributs de Wires::Sectionaliser	589
Tableau 272 – Extrémités d'association de Wires::Sectionaliser avec d'autres classes	590
Tableau 273 – Attributs de Wires::SeriesCompensator	591
Tableau 274 – Extrémités d'association de Wires::SeriesCompensator avec d'autres classes	591
Tableau 275 – Attributs de Wires::ShuntCompensator	592
Tableau 276 – Extrémités d'association de Wires::ShuntCompensator avec d'autres classes	593
Tableau 277 – Attributs de Wires::ShuntCompensatorPhase	594
Tableau 278 – Extrémités d'association de Wires::ShuntCompensatorPhase avec d'autres classes	594
Tableau 279 – Libellés de Wires::SinglePhaseKind	595
Tableau 280 – Attributs de Wires::StaticVarCompensator	595
Tableau 281 – Extrémités d'association de Wires::StaticVarCompensator avec d'autres classes	596
Tableau 282 – Libellés de Wires::SVCControlMode	596
Tableau 283 – Attributs de Wires::Switch	596
Tableau 284 – Extrémités d'association de Wires::Switch avec d'autres classes	597
Tableau 285 – Attributs de Wires::SwitchPhase	598
Tableau 286 – Extrémités d'association de Wires::SwitchPhase avec d'autres classes	598
Tableau 287 – Attributs de Wires::SwitchSchedule	599
Tableau 288 – Extrémités d'association de Wires::SwitchSchedule avec d'autres classes	599
Tableau 289 – Libellés de Wires::SynchronousGeneratorType	599
Tableau 290 – Attributs de Wires::SynchronousMachine	600
Tableau 291 – Extrémités d'association de Wires::SynchronousMachine avec d'autres classes	602
Tableau 292 – Libellés de Wires::SynchronousMachineOperatingMode	603
Tableau 293 – Libellés de Wires::SynchronousMachineType	603
Tableau 294 – Attributs de Wires::TapChanger	603
Tableau 295 – Extrémités d'association de Wires::TapChanger avec d'autres classes	604
Tableau 296 – Attributs de Wires::TapChangerControl	605
Tableau 297 – Extrémités d'association de Wires::TapChangerControl avec d'autres classes	606
Tableau 298 – Libellés de Wires::TapChangerKind	606
Tableau 299 – Attributs de Wires::TapSchedule	607
Tableau 300 – Extrémités d'association de Wires::TapSchedule avec d'autres classes	607
Tableau 301 – Libellés de Wires::TransformerControlMode	607
Tableau 302 – Attributs de Wires::TransformerCoreAdmittance	608
Tableau 303 – Extrémités d'association de Wires::TransformerCoreAdmittance avec d'autres classes	608

Tableau 304 – Attributs de Wires::TransformerEnd	608
Tableau 305 – Extrémités d'association de Wires::TransformerEnd avec d'autres classes	609
Tableau 306 – Attributs de Wires::TransformerMeshImpedance	610
Tableau 307 – Extrémités d'association de Wires::TransformerMeshImpedance avec d'autres classes	610
Tableau 308 – Attributs de Wires::TransformerStarImpedance	611
Tableau 309 – Extrémités d'association de Wires::TransformerStarImpedance avec d'autres classes	611
Tableau 310 – Attributs de Wires::TransformerTank	611
Tableau 311 – Extrémités d'association de Wires::TransformerTank avec d'autres classes	612
Tableau 312 – Attributs de Wires::TransformerTankEnd	612
Tableau 313 – Extrémités d'association de Wires::TransformerTankEnd avec d'autres classes	613
Tableau 314 – Attributs de Wires::VoltageControlZone	613
Tableau 315 – Extrémités d'association de Wires::VoltageControlZone avec d'autres classes	613
Tableau 316 – Libellés de Wires::WindingConnection	614
Tableau 317 – Libellés de GenerationDynamics::BoilerControlMode	616
Tableau 318 – Attributs de GenerationDynamics::BWRSteamSupply	617
Tableau 319 – Extrémités d'association de GenerationDynamics::BWRSteamSupply avec d'autres classes	618
Tableau 320 – Attributs de GenerationDynamics::CombustionTurbine	618
Tableau 321 – Extrémités d'association de GenerationDynamics::CombustionTurbine avec d'autres classes	619
Tableau 322 – Attributs de GenerationDynamics::CTTempActivePowerCurve	619
Tableau 323 – Extrémités d'association de GenerationDynamics::CTTempActivePowerCurve avec d'autres classes	620
Tableau 324 – Attributs de GenerationDynamics::DrumBoiler	620
Tableau 325 – Extrémités d'association de GenerationDynamics::DrumBoiler avec d'autres classes	621
Tableau 326 – Attributs de GenerationDynamics::FossilSteamSupply	621
Tableau 327 – Extrémités d'association de GenerationDynamics::FossilSteamSupply avec d'autres classes	622
Tableau 328 – Attributs de GenerationDynamics::HeatRecoveryBoiler	623
Tableau 329 – Extrémités d'association de GenerationDynamics::HeatRecoveryBoiler avec d'autres classes	624
Tableau 330 – Attributs de GenerationDynamics::HydroTurbine	624
Tableau 331 – Extrémités d'association de GenerationDynamics::HydroTurbine avec d'autres classes	625
Tableau 332 – Attributs de GenerationDynamics::PrimeMover	625
Tableau 333 – Extrémités d'association de GenerationDynamics::PrimeMover avec d'autres classes	625
Tableau 334 – Attributs de GenerationDynamics::PWRSteamSupply	626
Tableau 335 – Extrémités d'association de GenerationDynamics::PWRSteamSupply avec d'autres classes	627
Tableau 336 – Attributs de GenerationDynamics::SteamSupply	627

Tableau 337 – Extrémités d'association de GenerationDynamics::SteamSupply avec d'autres classes	628
Tableau 338 – Attributs de GenerationDynamics::SteamTurbine	628
Tableau 339 – Extrémités d'association de GenerationDynamics::SteamTurbine avec d'autres classes	629
Tableau 340 – Attributs de GenerationDynamics::Subcritical	629
Tableau 341 – Extrémités d'association de GenerationDynamics::Subcritical avec d'autres classes	630
Tableau 342 – Attributs de GenerationDynamics::Supercritical	631
Tableau 343 – Extrémités d'association de GenerationDynamics::Supercritical avec d'autres classes	632
Tableau 344 – Libellés de GenerationDynamics::TurbineType	632
Tableau 345 – Attributs de Production::AirCompressor	637
Tableau 346 – Extrémités d'association de Production::AirCompressor avec d'autres classes	638
Tableau 347 – Attributs de Production::CAESPlant	638
Tableau 348 – Extrémités d'association de Production::CAESPlant avec d'autres classes	638
Tableau 349 – Attributs de Production::Classification	639
Tableau 350 – Attributs de Production::CogenerationPlant	639
Tableau 351 – Extrémités d'association de Production::CogenerationPlant avec d'autres classes	640
Tableau 352 – Attributs de Production::CombinedCyclePlant	640
Tableau 353 – Extrémités d'association de Production::CombinedCyclePlant avec d'autres classes	640
Tableau 354 – Attributs de Production::CostPerHeatUnit	641
Tableau 355 – Attributs de Production::Emission	641
Tableau 356 – Attributs de Production::EmissionAccount	642
Tableau 357 – Extrémités d'association de Production::EmissionAccount avec d'autres classes	642
Tableau 358 – Attributs de Production::EmissionCurve	643
Tableau 359 – Extrémités d'association de Production::EmissionCurve avec d'autres classes	643
Tableau 360 – Libellés de Production::EmissionType	644
Tableau 361 – Libellés de Production::EmissionValueSource	644
Tableau 362 – Attributs de Production::FossilFuel	645
Tableau 363 – Extrémités d'association de Production::FossilFuel avec d'autres classes	645
Tableau 364 – Attributs de Production::FuelAllocationSchedule	646
Tableau 365 – Extrémités d'association de Production::FuelAllocationSchedule avec d'autres classes	646
Tableau 366 – Libellés de Production::FuelType	647
Tableau 367 – Attributs de Production::GeneratingUnit	647
Tableau 368 – Extrémités d'association de Production::GeneratingUnit avec d'autres classes	650
Tableau 369 – Libellés de Production::GeneratorControlMode	650
Tableau 370 – Libellés de Production::GeneratorControlSource	651

Tableau 371 – Libellés de Production::GeneratorOperatingMode	651
Tableau 372 – Attributs de Production::GenUnitOpCostCurve	652
Tableau 373 – Extrémités d'association de Production::GenUnitOpCostCurve avec d'autres classes	652
Tableau 374 – Attributs de Production::GenUnitOpSchedule	653
Tableau 375 – Extrémités d'association de Production::GenUnitOpSchedule avec d'autres classes	653
Tableau 376 – Attributs de Production::GrossToNetActivePowerCurve	654
Tableau 377 – Extrémités d'association de Production::GrossToNetActivePowerCurve avec d'autres classes	654
Tableau 378 – Attributs de Production::HeatInputCurve	655
Tableau 379 – Extrémités d'association de Production::HeatInputCurve avec d'autres classes	655
Tableau 380 – Attributs de Production::HeatRate	656
Tableau 381 – Attributs de Production::HeatRateCurve	656
Tableau 382 – Extrémités d'association de Production::HeatRateCurve avec d'autres classes	656
Tableau 383 – Libellés de Production::HydroEnergyConversionKind	657
Tableau 384 – Attributs de Production::HydroGeneratingEfficiencyCurve	657
Tableau 385 – Extrémités d'association de Production::HydroGeneratingEfficiencyCurve avec d'autres classes	658
Tableau 386 – Attributs de Production::HydroGeneratingUnit	658
Tableau 387 – Extrémités d'association de Production::HydroGeneratingUnit avec d'autres classes	660
Tableau 388 – Libellés de Production::HydroPlantType	660
Tableau 389 – Attributs de Production::HydroPowerPlant	661
Tableau 390 – Extrémités d'association de Production::HydroPowerPlant avec d'autres classes	662
Tableau 391 – Attributs de Production::HydroPump	662
Tableau 392 – Extrémités d'association de Production::HydroPump avec d'autres classes	663
Tableau 393 – Attributs de Production::HydroPumpOpSchedule	663
Tableau 394 – Extrémités d'association de Production::HydroPumpOpSchedule avec d'autres classes	664
Tableau 395 – Attributs de Production::IncrementalHeatRateCurve	664
Tableau 396 – Extrémités d'association de Production::IncrementalHeatRateCurve avec d'autres classes	665
Tableau 397 – Attributs de Production::InflowForecast	665
Tableau 398 – Extrémités d'association de Production::InflowForecast avec d'autres classes	665
Tableau 399 – Attributs de Production::LevelVsVolumeCurve	666
Tableau 400 – Extrémités d'association de Production::LevelVsVolumeCurve avec d'autres classes	666
Tableau 401 – Attributs de Production::NuclearGeneratingUnit	667
Tableau 402 – Extrémités d'association de Production::NuclearGeneratingUnit avec d'autres classes	668
Tableau 403 – Attributs de Production::PenstockLossCurve	669

Tableau 404 – Extrémités d'association de Production::PenstockLossCurve avec d'autres classes	669
Tableau 405 – Libellés de Production::PenstockType	669
Tableau 406 – Attributs de Production::Reservoir	670
Tableau 407 – Extrémités d'association de Production::Reservoir avec d'autres classes	671
Tableau 408 – Attributs de Production::ShutdownCurve	672
Tableau 409 – Extrémités d'association de Production::ShutdownCurve avec d'autres classes	672
Tableau 410 – Libellés de Production::SpillwayGateType	672
Tableau 411 – Attributs de Production::StartIgnFuelCurve	673
Tableau 412 – Extrémités d'association de Production::StartIgnFuelCurve avec d'autres classes	673
Tableau 413 – Attributs de Production::StartMainFuelCurve	674
Tableau 414 – Extrémités d'association de Production::StartMainFuelCurve avec d'autres classes	674
Tableau 415 – Attributs de Production::StartRampCurve	675
Tableau 416 – Extrémités d'association de Production::StartRampCurve avec d'autres classes	675
Tableau 417 – Attributs de Production::StartupModel	676
Tableau 418 – Extrémités d'association de Production::StartupModel avec d'autres classes	676
Tableau 419 – Attributs de Production::SteamSendoutSchedule	677
Tableau 420 – Extrémités d'association de Production::SteamSendoutSchedule avec d'autres classes	677
Tableau 421 – Libellés de Production::SurgeTankCode	677
Tableau 422 – Attributs de Production::TailbayLossCurve	678
Tableau 423 – Extrémités d'association de Production::TailbayLossCurve avec d'autres classes	678
Tableau 424 – Attributs de Production::TargetLevelSchedule	679
Tableau 425 – Extrémités d'association de Production::TargetLevelSchedule avec d'autres classes	679
Tableau 426 – Attributs de Production::ThermalGeneratingUnit	680
Tableau 427 – Extrémités d'association de Production::ThermalGeneratingUnit avec d'autres classes	681
Tableau 428 – Attributs de Production::WindGeneratingUnit	682
Tableau 429 – Extrémités d'association de Production::WindGeneratingUnit avec d'autres classes	684
Tableau 430 – Attributs de LoadModel::ConformLoad	686
Tableau 431 – Extrémités d'association de LoadModel::ConformLoad avec d'autres classes	687
Tableau 432 – Attributs de LoadModel::ConformLoadGroup	687
Tableau 433 – Extrémités d'association de LoadModel::ConformLoadGroup avec d'autres classes	688
Tableau 434 – Attributs de LoadModel::ConformLoadSchedule	688
Tableau 435 – Extrémités d'association de LoadModel::ConformLoadSchedule avec d'autres classes	689
Tableau 436 – Attributs de LoadModel::DayType	689

Tableau 437 – Extrémités d'association de LoadModel::DayType avec d'autres classes	689
Tableau 438 – Attributs de LoadModel::EnergyArea	690
Tableau 439 – Extrémités d'association de LoadModel::EnergyArea avec d'autres classes	690
Tableau 440 – Attributs de LoadModel::LoadArea	690
Tableau 441 – Extrémités d'association de LoadModel::LoadArea avec d'autres classes	690
Tableau 442 – Attributs de LoadModel::LoadGroup	691
Tableau 443 – Extrémités d'association de LoadModel::LoadGroup avec d'autres classes	691
Tableau 444 – Attributs de LoadModel::LoadResponseCharacteristic	692
Tableau 445 – Extrémités d'association de LoadModel::LoadResponseCharacteristic avec d'autres classes	692
Tableau 446 – Attributs de LoadModel::NonConformLoad	693
Tableau 447 – Extrémités d'association de LoadModel::NonConformLoad avec d'autres classes	694
Tableau 448 – Attributs de LoadModel::NonConformLoadGroup	694
Tableau 449 – Extrémités d'association de LoadModel::NonConformLoadGroup avec d'autres classes	695
Tableau 450 – Attributs de LoadModel::NonConformLoadSchedule	695
Tableau 451 – Extrémités d'association de LoadModel::NonConformLoadSchedule avec d'autres classes	696
Tableau 452 – Attributs de LoadModel::PowerCutZone	696
Tableau 453 – Extrémités d'association de LoadModel::PowerCutZone avec d'autres classes	696
Tableau 454 – Attributs de LoadModel::Season	697
Tableau 455 – Extrémités d'association de LoadModel::Season avec d'autres classes	697
Tableau 456 – Attributs de LoadModel::SeasonDayTypeSchedule	697
Tableau 457 – Extrémités d'association de LoadModel::SeasonDayTypeSchedule avec d'autres classes	698
Tableau 458 – Libellés de LoadModel::SeasonName	698
Tableau 459 – Attributs de LoadModel::StationSupply	699
Tableau 460 – Extrémités d'association de LoadModel::StationSupply avec d'autres classes	699
Tableau 461 – Attributs de LoadModel::SubLoadArea	700
Tableau 462 – Extrémités d'association de LoadModel::SubLoadArea avec d'autres classes	700
Tableau 463 – Attributs de Outage::ClearanceTag	702
Tableau 464 – Extrémités d'association de Outage::ClearanceTag avec d'autres classes	702
Tableau 465 – Attributs de Outage::ClearanceTagType	702
Tableau 466 – Extrémités d'association de Outage::ClearanceTagType avec d'autres classes	703
Tableau 467 – Attributs de Outage::OutageSchedule	703
Tableau 468 – Extrémités d'association de Outage::OutageSchedule avec d'autres classes	703
Tableau 469 – Attributs de Outage::SwitchingOperation	704

Tableau 470 – Extrémités d'association de Outage::SwitchingOperation avec d'autres classes	704
Tableau 471 – Libellés de Outage::SwitchState	704
Tableau 472 – Attributs de AuxiliaryEquipment::AuxiliaryEquipment	705
Tableau 473 – Extrémités d'association de AuxiliaryEquipment::AuxiliaryEquipment avec d'autres classes	706
Tableau 474 – Attributs de AuxiliaryEquipment::CurrentTransformer	706
Tableau 475 – Extrémités d'association de AuxiliaryEquipment::CurrentTransformer avec d'autres classes	707
Tableau 476 – Attributs de AuxiliaryEquipment::FaultIndicator	707
Tableau 477 – Extrémités d'association de AuxiliaryEquipment::FaultIndicator avec d'autres classes	708
Tableau 478 – Attributs de AuxiliaryEquipment::PostLineSensor	708
Tableau 479 – Extrémités d'association de AuxiliaryEquipment::PostLineSensor avec d'autres classes	709
Tableau 480 – Attributs de AuxiliaryEquipment::PotentialTransformer	709
Tableau 481 – Extrémités d'association de AuxiliaryEquipment::PotentialTransformer avec d'autres classes	710
Tableau 482 – Attributs de AuxiliaryEquipment::Sensor	710
Tableau 483 – Extrémités d'association de AuxiliaryEquipment::Sensor avec d'autres classes	711
Tableau 484 – Attributs de AuxiliaryEquipment::SurgeProtector	711
Tableau 485 – Extrémités d'association de AuxiliaryEquipment::SurgeProtector avec d'autres classes	712
Tableau 486 – Attributs de Protection::CurrentRelay	713
Tableau 487 – Extrémités d'association de Protection::CurrentRelay avec d'autres classes	713
Tableau 488 – Attributs de Protection::ProtectionEquipment	714
Tableau 489 – Extrémités d'association de Protection::ProtectionEquipment avec d'autres classes	715
Tableau 490 – Attributs de Protection::RecloseSequence	715
Tableau 491 – Extrémités d'association de Protection::RecloseSequence avec d'autres classes	716
Tableau 492 – Attributs de Protection::SynchrocheckRelay	716
Tableau 493 – Extrémités d'association de Protection::SynchrocheckRelay avec d'autres classes	717
Tableau 494 – Attributs de Equivalents::EquivalentBranch	718
Tableau 495 – Extrémités d'association de Equivalents::EquivalentBranch avec d'autres classes	719
Tableau 496 – Attributs de Equivalents::EquivalentEquipment	719
Tableau 497 – Extrémités d'association de Equivalents::EquivalentEquipment avec d'autres classes	720
Tableau 498 – Attributs de Equivalents::EquivalentInjection	720
Tableau 499 – Extrémités d'association de Equivalents::EquivalentInjection avec d'autres classes	721
Tableau 500 – Attributs de Equivalents::EquivalentNetwork	721
Tableau 501 – Extrémités d'association de Equivalents::EquivalentNetwork avec d'autres classes	722

Tableau 502 – Attributs de EquivalentShunt.....	722
Tableau 503 – Extrémités d'association de EquivalentShunt avec d'autres classes	723
Tableau 504 – Attributs de Meas::Accumulator	727
Tableau 505 – Extrémités d'association de Meas::Accumulator avec d'autres classes	727
Tableau 506 – Attributs de Meas::AccumulatorLimit.....	728
Tableau 507 – Extrémités d'association de Meas::AccumulatorLimit avec d'autres classes	728
Tableau 508 – Attributs de Meas::AccumulatorLimitSet.....	728
Tableau 509 – Extrémités d'association de Meas::AccumulatorLimitSet avec d'autres classes	729
Tableau 510 – Attributs de Meas::AccumulatorValue	729
Tableau 511 – Extrémités d'association de Meas::AccumulatorValue avec d'autres classes	729
Tableau 512 – Attributs de Meas::Analog.....	730
Tableau 513 – Extrémités d'association de Meas::Analog avec d'autres classes	730
Tableau 514 – Attributs de Meas::AnalogLimit	731
Tableau 515 – Extrémités d'association de Meas::AnalogLimit avec d'autres classes	731
Tableau 516 – Attributs de Meas::AnalogLimitSet.....	731
Tableau 517 – Extrémités d'association de Meas::AnalogLimitSet avec d'autres classes	731
Tableau 518 – Attributs de Meas::AnalogValue.....	732
Tableau 519 – Extrémités d'association de Meas::AnalogValue avec d'autres classes	732
Tableau 520 – Attributs de Meas::Command.....	733
Tableau 521 – Extrémités d'association de Meas::Command avec d'autres classes	733
Tableau 522 – Attributs de Meas::Control	734
Tableau 523 – Extrémités d'association de Meas::Control avec d'autres classes	734
Tableau 524 – Attributs de Meas::ControlType.....	734
Tableau 525 – Extrémités d'association de Meas::ControlType avec d'autres classes	735
Tableau 526 – Attributs de Meas::Discrete.....	735
Tableau 527 – Extrémités d'association de Meas::Discrete avec d'autres classes	736
Tableau 528 – Attributs de Meas::DiscreteValue.....	736
Tableau 529 – Extrémités d'association de Meas::DiscreteValue avec d'autres classes	736
Tableau 530 – Attributs de Meas::Limit	737
Tableau 531 – Extrémités d'association de Meas::Limit avec d'autres classes	737
Tableau 532 – Attributs de Meas::LimitSet.....	737
Tableau 533 – Extrémités d'association de Meas::LimitSet avec d'autres classes	738
Tableau 534 – Attributs de Meas::Measurement	739
Tableau 535 – Extrémités d'association de Meas::Measurement avec d'autres classes.....	739
Tableau 536 – Attributs de Meas::MeasurementValue.....	739
Tableau 537 – Extrémités d'association de Meas::MeasurementValue avec d'autres classes	740
Tableau 538 – Attributs de Meas::MeasurementValueQuality.....	740
Tableau 539 – Extrémités d'association de Meas::MeasurementValueQuality avec d'autres classes	741

Tableau 540 – Attributs de Meas::MeasurementValueSource	741
Tableau 541 – Extrémités d'association de Meas::MeasurementValueSource avec d'autres classes	741
Tableau 542 – Attributs de Meas::Quality61850	742
Tableau 543 – Attributs de Meas::SetPoint	743
Tableau 544 – Extrémités d'association de Meas::SetPoint avec d'autres classes	743
Tableau 545 – Attributs de Meas::StringMeasurement	744
Tableau 546 – Extrémités d'association de Meas::StringMeasurement avec d'autres classes	744
Tableau 547 – Attributs de Meas::StringMeasurementValue	744
Tableau 548 – Extrémités d'association de Meas::StringMeasurementValue avec d'autres classes	745
Tableau 549 – Libellés de Meas::Validity	745
Tableau 550 – Attributs de Meas::ValueAliasSet	745
Tableau 551 – Extrémités d'association de Meas::ValueAliasSet avec d'autres classes	746
Tableau 552 – Attributs de Meas::ValueToAlias	746
Tableau 553 – Extrémités d'association de Meas::ValueToAlias avec d'autres classes	746
Tableau 554 – Attributs de SCADA::CommunicationLink	748
Tableau 555 – Extrémités d'association de SCADA::CommunicationLink avec d'autres classes	748
Tableau 556 – Attributs de SCADA::RemoteControl	748
Tableau 557 – Extrémités d'association de SCADA::RemoteControl avec d'autres classes	749
Tableau 558 – Attributs de SCADA::RemotePoint	749
Tableau 559 – Extrémités d'association de SCADA::RemotePoint avec d'autres classes	749
Tableau 560 – Attributs de SCADA::RemoteSource	750
Tableau 561 – Extrémités d'association de SCADA::RemoteSource avec d'autres classes	750
Tableau 562 – Attributs de SCADA::RemoteUnit	750
Tableau 563 – Extrémités d'association de SCADA::RemoteUnit avec d'autres classes	751
Tableau 564 – Libellés de SCADA::RemoteUnitType	751
Tableau 565 – Libellés de SCADA::Source	751
Tableau 566 – Attributs de ControlArea::AltGeneratingUnitMeas	754
Tableau 567 – Extrémités d'association de ControlArea::AltGeneratingUnitMeas avec d'autres classes	754
Tableau 568 – Attributs de ControlArea::AltTieMeas	755
Tableau 569 – Extrémités d'association de ControlArea::AltTieMeas avec d'autres classes	755
Tableau 570 – Attributs de ControlArea::ControlArea	755
Tableau 571 – Extrémités d'association de ControlArea::ControlArea avec d'autres classes	756
Tableau 572 – Extrémités d'association de ControlArea::ControlAreaGeneratingUnit avec d'autres classes	756
Tableau 573 – Libellés de ControlArea::ControlAreaTypeKind	757
Tableau 574 – Attributs de ControlArea::TieFlow	757
Tableau 575 – Extrémités d'association de ControlArea::TieFlow avec d'autres classes	757

Tableau 576 – Attributs de Contingency::Contingency	758
Tableau 577 – Extrémités d'association de Contingency::Contingency avec d'autres classes	759
Tableau 578 – Attributs de Contingency::ContingencyElement.....	759
Tableau 579 – Extrémités d'association de Contingency::ContingencyElement avec d'autres classes	759
Tableau 580 – Attributs de Contingency::ContingencyEquipment	760
Tableau 581 – Extrémités d'association de Contingency::ContingencyEquipment avec d'autres classes	760
Tableau 582 – Libellés de Contingency::ContingencyEquipmentStatusKind	760
Tableau 583 – Attributs de StateVariables::SvInjection	762
Tableau 584 – Extrémités d'association de StateVariables::SvInjection avec d'autres classes	762
Tableau 585 – Attributs de StateVariables::SvPowerFlow	762
Tableau 586 – Extrémités d'association de StateVariables::SvPowerFlow avec d'autres classes	763
Tableau 587 – Attributs de StateVariables::SvShortCircuit.....	763
Tableau 588 – Extrémités d'association de StateVariables::SvShortCircuit avec d'autres classes	763
Tableau 589 – Attributs de StateVariables::SvShuntCompensatorSections	763
Tableau 590 – Extrémités d'association de StateVariables::SvShuntCompensatorSections avec d'autres classes	764
Tableau 591 – Attributs de StateVariables::SvStatus	764
Tableau 592 – Extrémités d'association de StateVariables::SvStatus avec d'autres classes	764
Tableau 593 – Attributs de StateVariables::SvTapStep	764
Tableau 594 – Extrémités d'association de StateVariables::SvTapStep avec d'autres classes	765
Tableau 595 – Attributs de StateVariables::SvVoltage	765
Tableau 596 – Extrémités d'association de StateVariables::SvVoltage avec d'autres classes	765

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

**INTERFACE DE PROGRAMMATION D'APPLICATION
POUR SYSTÈME DE GESTION D'ÉNERGIE (EMS-API) –****Partie 301: Base de modèle d'information commun (CIM)**

AVANT-PROPOS

- 1) La Commission Electrotechnique Internationale (CEI) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de la CEI). La CEI a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, la CEI – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de la CEI"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec la CEI, participent également aux travaux. La CEI collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de la CEI concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de la CEI intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de la CEI se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de la CEI. Tous les efforts raisonnables sont entrepris afin que la CEI s'assure de l'exactitude du contenu technique de ses publications; la CEI ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de la CEI s'engagent, dans toute la mesure possible, à appliquer de façon transparente, les Publications de la CEI dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de la CEI et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) La CEI elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de la CEI. La CEI n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à la CEI, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de la CEI, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de la CEI ou de toute autre Publication de la CEI, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.

La Norme internationale CEI 61970-301 a été établie par le comité d'études 57 de la CEI: Gestion des systèmes de puissance et échanges d'informations associés.

La présente cinquième édition annule et remplace la quatrième édition publiée en 2013-05. Cette cinquième édition constitue une révision technique.

Les modifications techniques majeures par rapport à l'édition précédente sont les suivantes:

- les modèles de transformateurs ont été modifiés pour qu'ils soient adaptés à une utilisation dans les domaines de la distribution et du transport. Le modèle de changeur de prise a en outre été mis à jour pour refléter plus clairement l'utilisation prévue sans se référer à des règles pour lesquelles des attributs sont adaptés dans ces situations;
- l'ajout d'une approche de dénomination plus générale et plus claire et suppression des attributs ambigus liés à la dénomination. Cette approche permet aux utilisateurs de définir de nouveaux domaines de noms et de leur donner leur propre description unique;

- des modèles de fils de phase pour les composants ont été améliorés pour décrire les connexions et les attributs spécifiques de phase interne;
- des modèles de présentation des diagrammes ont été ajoutés pour faciliter l'échange des informations sur la présentation des diagrammes;
- ajout de nouveaux types Decimal, mise au net de types date et heure;
- ajout de nouveaux types de données Compound au paquetage Domain.

Le texte de la présente norme est issu des documents suivants:

FDIS	Rapport de vote
57/1395/FDIS	57/1417/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de la présente norme.

La présente publication a été rédigée selon les Directives ISO/CEI, Partie 2.

Une liste de toutes les parties de la série CEI 61970, sous le titre général: *Interface de programmation d'application pour système de gestion d'énergie (EMS-API)*, peut être consultée sur le site web de la CEI.

Le comité a décidé que le contenu de cette publication ne sera pas modifié avant la date de stabilité indiquée sur le site web de la CEI sous "<http://webstore.iec.ch>" dans les données relatives à la publication recherchée. A cette date, la publication sera

- reconduite,
- supprimée,
- remplacée par une édition révisée, ou
- amendée.

IMPORTANT – Le logo "colour inside" qui se trouve sur la page de couverture de cette publication indique qu'elle contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer cette publication en utilisant une imprimante couleur.

INTRODUCTION

Cette norme est l'une des parties de la série de normes CEI 61970 qui définissent une interface de programmation d'application (API *Application program interface*) pour un système de gestion d'énergie (EMS *Energy management system*). La présente norme a été initialement fondée sur les travaux réalisés dans le cadre du projet de recherche (RP-3654-1) sur les API de centres de conduite (CCAPI) de l'EPRI. Le projet CCAPI de l'EPRI avait principalement pour objet de:

- réduire les coûts et le temps nécessaires à l'ajout de nouvelles applications à un EMS;
- protéger l'investissement dans les applications existantes qui fonctionnent efficacement dans un EMS.

Le principal but de la série de normes CEI 61970 est de produire les normes destinées à faciliter l'intégration d'applications EMS développées de façon indépendante par différents fournisseurs, entre des systèmes EMS complets développés de façon indépendante ou entre un système EMS et d'autres systèmes concernés par différents aspects de l'exploitation d'un système électrique, tels que les systèmes de gestion de la distribution (DMS *Distribution management systems*) ou de la production. Cela s'effectue par la définition de l'API normalisée pour permettre à ces applications ou systèmes d'accéder aux données publiques et d'échanger des informations indépendamment de la représentation interne de ces informations.

Le modèle d'information commun (CIM *Common information model*) spécifie la sémantique de cette API. Les spécifications des composants d'interface (CIS *Component interface specifications*), qui sont contenues dans d'autres parties des normes CEI 61970, précisent le contenu des messages échangés.

Le CIM est un modèle abstrait contenant tous les objets principaux d'une entreprise de distribution d'électricité habituellement nécessaires pour représenter les opérations d'une entreprise d'électricité. Ce modèle inclut les classes et les attributs publics de ces objets, ainsi que les relations entre eux.

Les objets représentés dans le CIM sont de nature abstraite et peuvent être utilisés dans une large gamme d'applications. L'utilisation du CIM n'est pas limitée à son application dans un EMS. Il convient que cette norme soit comprise comme un outil permettant l'intégration dans tout domaine où un modèle commun de réseau est nécessaire pour faciliter l'interopérabilité et la compatibilité de connexion entre des applications et des systèmes indépendants de toute mise en œuvre particulière.

La présente norme, CEI 61970-301, définit la base du CIM constituée d'un ensemble de paquetages qui offrent une vue logique des aspects fonctionnels d'un EMS incluant SCADA (*Supervisory control and data acquisition*). D'autres domaines fonctionnels sont normalisés dans des documents CEI distincts qui augmentent et référencent la présente norme CIM de base. Par exemple, la CEI 61968-11 traite des modèles de distribution et référence la présente norme CIM de base. Alors qu'il existe plusieurs normes de la CEI qui traitent des différentes parties du CIM, un seul modèle d'information unifié comprenant le CIM est derrière tous ces documents normatifs individuels.

La commission électrotechnique internationale (CEI) attire l'attention sur le fait qu'il est déclaré que la conformité avec le présent document peut impliquer l'utilisation d'un brevet concernant une mise en œuvre gérée par ordinateur d'un modèle de système électrique orienté objet dans une base de données relationnelle. À ce titre, elle n'est en conflit avec aucun développement modèle logique de système électrique incluant le CIM, lorsque la mise en œuvre du modèle n'est pas définie.

La CEI ne prend pas position eu égard à la preuve, la validité et la portée de ce droit de propriété.

Le détenteur de ces droits de propriété a donné l'assurance à la CEI qu'il consent à négocier des licences avec des demandeurs du monde entier, soit sans frais soit à des termes et conditions raisonnables et non discriminatoires. A ce propos, la déclaration du détenteur de ce droit de propriété est enregistrée à la CEI. Des informations peuvent être obtenues auprès de:

ICL

Wenlock Way

West Gorton

Manchester

M12 5DR

Royaume-Uni (R.U.)

L'attention est d'autre part attirée sur le fait que certains des éléments du présent document peuvent faire l'objet de droits de propriété autres que ceux qui ont été mentionnés ci-dessus. La CEI ne saurait être tenue pour responsable de l'identification de ces droits de propriété en tout ou partie.

L'ISO (www.iso.org/patents) et la CEI (<http://patents.iec.ch>) maintiennent des bases de données, consultables en ligne, des droits de propriété pertinents à leurs normes. Les utilisateurs sont encouragés à consulter ces bases de données pour obtenir l'information la plus récente concernant les droits de propriété.

INTERFACE DE PROGRAMMATION D'APPLICATION POUR SYSTÈME DE GESTION D'ÉNERGIE (EMS-API) –

Partie 301: Base de modèle d'information commun (CIM)

1 Domaine d'application

La CIM est un modèle abstrait qui représente tous les objets principaux d'une entreprise de distribution d'électricité habituellement nécessaires aux opérations d'une entreprise d'électricité. En fournissant une façon normalisée de représenter des ressources de réseaux électriques comme classes et attributs d'objets ainsi que leurs relations, le CIM facilite l'intégration des applications de l'EMS développées de façon indépendante par différents fournisseurs, entre des systèmes EMS complets développés de façon indépendante ou entre un système EMS et d'autres systèmes concernés par différents aspects des opérations d'un réseau électrique tels que la gestion de la production ou de la distribution. Le système SCADA est modélisé dans toute la mesure nécessaire pour prendre en charge la simulation des systèmes énergétiques et la communication entre des centres de commande. Le CIM facilite l'intégration en définissant un langage commun (c'est-à-dire une sémantique) fondé sur le modèle CIM pour permettre à ces applications ou systèmes d'accéder aux données publiques et d'échanger des informations indépendamment de la représentation interne de ces informations.

Les classes d'objets représentées dans le CIM sont de nature abstraite et peuvent être utilisées dans une large gamme d'applications. L'utilisation du CIM n'est pas limitée à son application dans un EMS. Il convient que cette norme soit comprise comme un outil permettant l'intégration dans tout domaine où un modèle commun de réseau est nécessaire pour faciliter l'interopérabilité et la compatibilité de connexion entre des applications et des systèmes indépendants de toute mise en œuvre particulière.

A cause de la taille du CIM complet, les classes d'objets qui le composent sont regroupées en un certain nombre de Paquetages logiques, qui représentent chacun une certaine partie du système électrique modélisé. Les collections de ces Paquetages sont fournies progressivement sous forme de Normes internationales distinctes. La présente Norme internationale particulière spécifie un ensemble de Base de paquetages qui offrent une vue logique sur les aspects fonctionnels des informations relatives à un EMS d'une entreprise de service public de distribution d'électricité qui sont partagées par toutes les applications. D'autres normes spécifient des aspects plus spécifiques du modèle qui ne sont nécessaires qu'à certaines applications. Le Paragraphe 4.2 définit le découpage actuel des paquetages dans les documents normatifs.

2 Références normatives

Les documents suivants sont cités en référence de manière normative, en intégralité ou en partie, dans le présent document et sont indispensables pour son application. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

CEI 60050 (toutes les parties): *Vocabulaire Electrotechnique International (VEI)*:
<http://www.electropedia.org>

CEI 60870-6 (toutes les parties), *Matériels et systèmes de téléconduite – Partie 6: Protocoles de téléconduite compatibles avec les normes ISO et les recommandations de l'UIT-T*

CEI 61850 (toutes les parties), *Réseaux et systèmes de communication pour l'automatisation des systèmes électriques*

CEI 61850-7-3:2010, *Réseaux et systèmes de communication pour l'automatisation des systèmes électriques – Partie 7-3: Structure de communication de base – Classes de données communes*

CEI 61850-7-4:2010, *Communication networks and systems for power utility automation – Part 7-4: Basic communication structure – Compatible logical node classes and data object classes* (disponible en anglais seulement)

CEI 61968 (toutes les parties), *Intégration d'applications pour les services électriques – Interfaces système pour la gestion de distribution*

CEI/TS 61970-2, *Energy management system application program interface (EMS-API) – Part 2: Glossary* (disponible en anglais seulement)

CEI 62325 (toutes les parties), *Cadre pour les communications pour le marché de l'énergie*

Object Management Group: UML 2.0 Specification – <http://www.omg.org>

3 Termes et définitions

Pour les besoins du présent document, les termes et définitions de la CEI 60050 et de la CEI/TS 61970-2 ainsi que les suivants s'appliquent.

3.1

système de gestion d'énergie

EMS (energy management system)

système informatique comprenant une plate-forme logicielle offrant les services de support de base et un ensemble d'applications offrant les fonctionnalités requises pour le bon fonctionnement des installations de production et de transport d'électricité afin d'assurer la sécurité adéquate d'approvisionnement énergétique à un coût minimal

3.2

interface de programmation d'application

API (application program interface)

ensemble des fonctions publiques qu'offre un composant exécutable d'application pour être utilisées par d'autres composants exécutables d'application

3.3

langage de modélisation unifié

UML (unified modeling language)

langage descriptif formel et complet avec des techniques de présentation de diagrammes utilisées pour représenter des systèmes logiciels, depuis l'analyse des exigences, en passant par la conception et la mise en œuvre, jusqu'à la documentation

Note 1 à l'article: Le langage UML a évolué pour devenir une Norme internationale en passant d'une collection de méthodes apportées par différents pratiquants. La CIM se réfère à l'UML pour définir le modèle, des outils automatiques génèrent directement à partir de l'UML la documentation, les schémas ainsi que d'autres artefacts. Il est nécessaire d'avoir des notions d'UML pour comprendre le CIM.

3.4

profil

sous-ensemble des classes, associations et attributs du DCIM nécessaires pour accomplir un type spécifique d'interface

Note 1 à l'article: Ceci peut être exprimé dans des fichiers XSD, RDF et/ou OWL. Un profil peut être soumis à essai entre des applications. Un profil est nécessaire pour "utiliser" le DCIM. Plusieurs profils sont définis dans d'autres parties de la série CEI 61968.

4 Spécification CIM

4.1 Notation de modélisation du CIM

Le CIM est défini en utilisant des techniques de modélisation orientées objet. Plus précisément, la spécification du CIM utilise la notation du Langage de modélisation unifié (UML) qui définit le CIM comme un groupe de paquetages.

Chaque paquetage du CIM contient un ou plusieurs diagrammes de classe montrant sous forme graphique toutes les classes de ce paquetage et leurs relations. Chaque classe est ensuite définie de façon textuelle en mettant l'accent sur ses attributs et ses relations avec d'autres classes.

La notation UML est décrite dans les documents de l'Object Management Group (OMG) et dans plusieurs autres manuels.

4.2 Paquetages CIM

Le CIM est fractionné en un ensemble de paquetages. Un paquetage est un moyen général permettant de grouper des éléments liés à un modèle. Les paquetages ont été choisis pour simplifier la conception, la compréhension et la révision du modèle. Le modèle d'information commun se compose de l'ensemble complet de paquetages. Des entités peuvent avoir des associations dépassant les limites de plusieurs paquetages. Chaque application utilisera des informations représentées dans plusieurs paquetages.

Le CIM complet est fractionné en groupes de paquetages afin de les gérer et maintenir commodément.

Les paquetages inclus dans la CEI 61970-301 (le présent document) sont contenus dans le paquetage de la CEI 61970. Les paquetages de la CEI 61970 ne dépendent pas d'autres paquetages externes. Comme le montre Figure 1, le paquetage de la CEI 61970 et ses sous-paquetages servent de modèle central ou de modèle de base à d'autres paquetages du CIM. Les lignes tiretées indiquent des relations de dépendance, la flèche allant du paquetage dépendant vers le paquetage dont il dépend. Les paquetages dans la série de normes CEI 61968 décrivent des parties supplémentaires du CIM qui traitent d'autres vues logiques des opérations d'une entreprise de service public, y compris la gestion du patrimoine, des emplacements, des activités, des clients, de la documentation et des travaux et des comptages. Les paquetages dans les séries de normes CEI 62325 décrivent les marchés de l'énergie électrique.

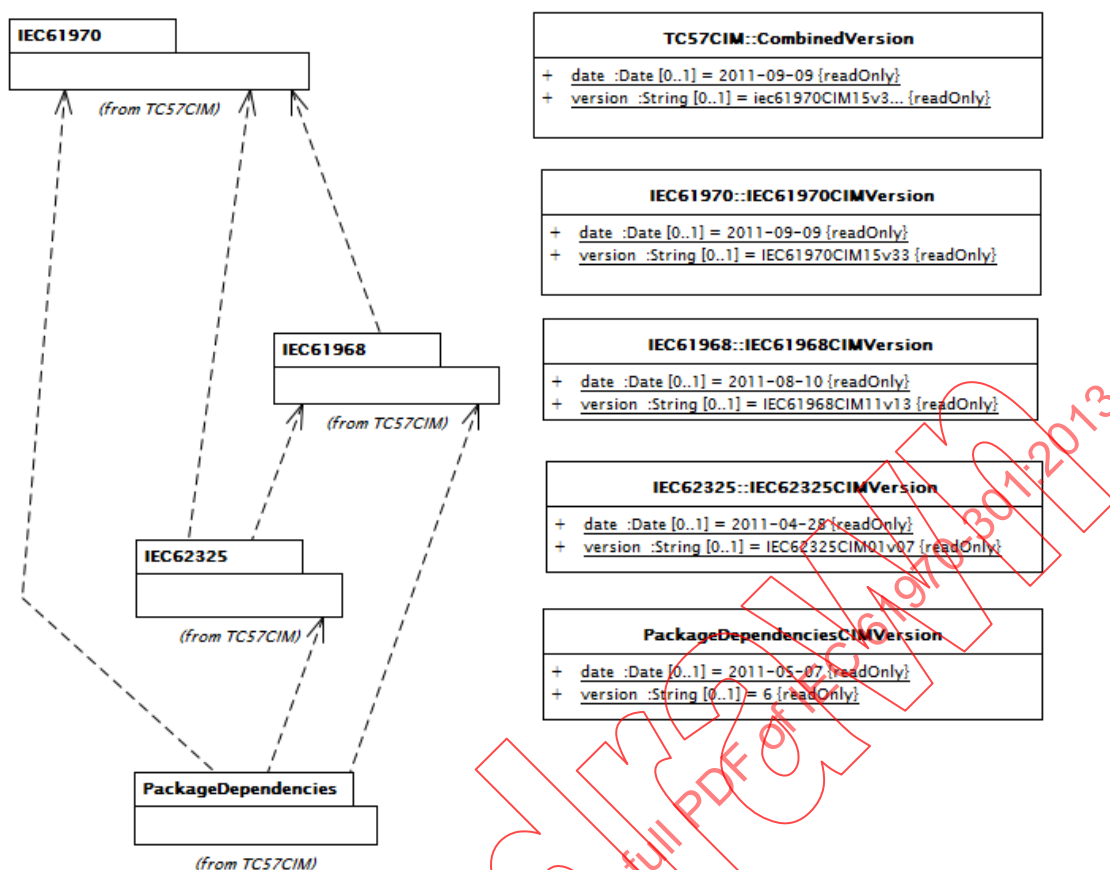


Figure 1 – Exemple de dépendances du packaging de groupe de travail

Noter que les limites du packaging n'impliquent pas de limites pour une application. Une application peut utiliser des entités CIM de différents packagings. Il est également prévu que des packagings du CIM en dehors de ceux décrits dans la présente norme auront des dépendances avec certains des packagings décrits dans la présente norme, et en particulier les packagings Domain et Core, même si d'autres dépendances existeront aussi.

La Figure 2 représente les packagings définis pour la Base de CIM de la CEI 61970-301 et leurs relations de dépendance.

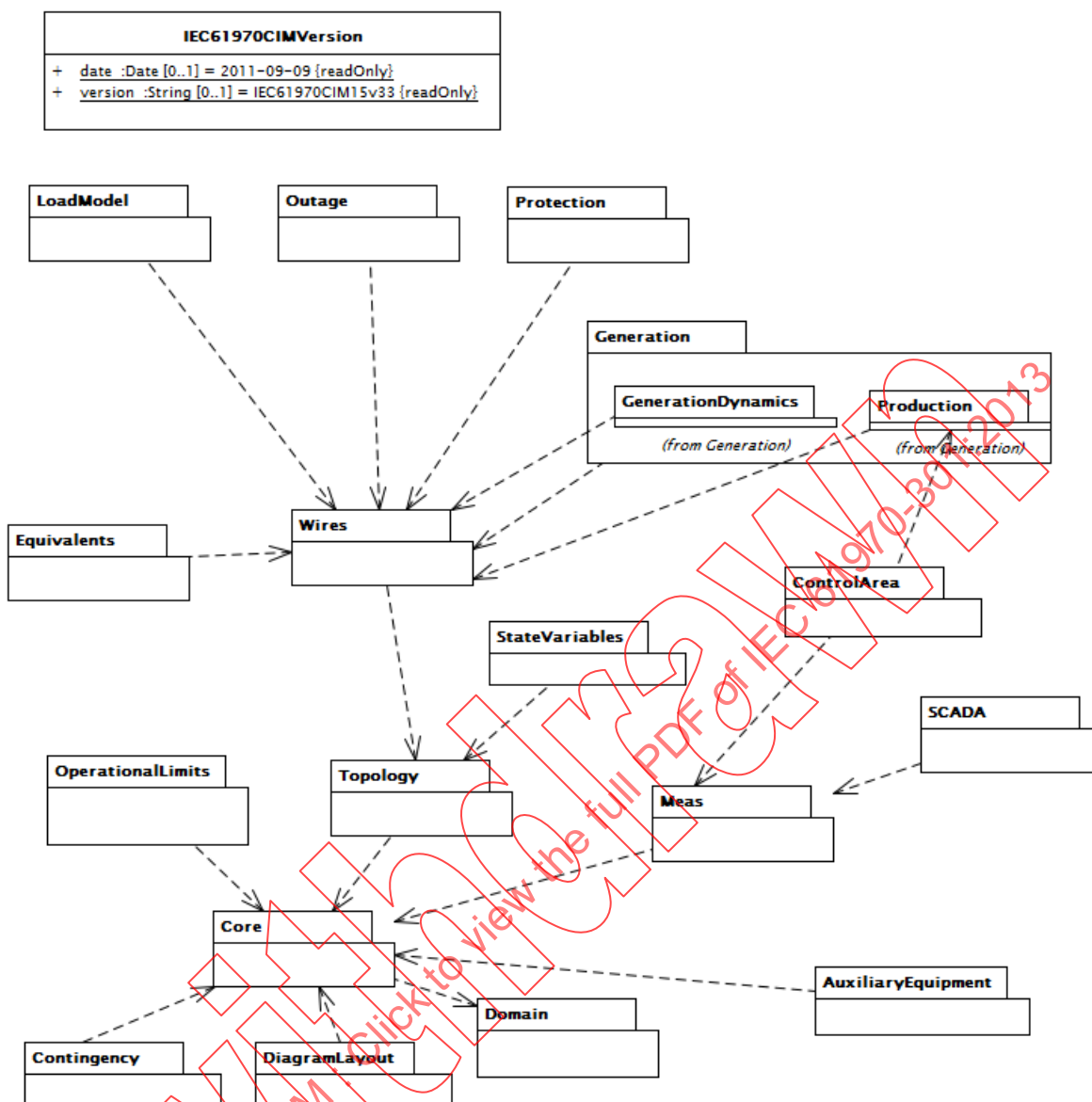


Figure 2 – Diagramme du packaging CIM CEI 61970-301

L'Article 6 contient la spécification pour chacun des paquetages Base du CIM.

NOTE Le contenu du CIM défini dans la présente spécification a été créé automatiquement à partir de la version IEC61970CIM15v33 du modèle électronique UML du CIM, qui est disponible auprès du CIM Users Group (groupe des utilisateurs du CIM).

4.3 Classes CIM et relations

4.3.1 Classes

Le ou les diagramme(s) de classe de chaque paquetage CIM représentent toutes les classes du paquetage et leurs relations. Lorsque des relations existent avec des classes dans d'autres paquetages, ces classes peuvent être également représentées.

Les classes et les objets modélisent ce qui se trouve dans un réseau électrique et qui doit être représenté de la même façon pour des applications de réseau électrique. Une classe est une description d'un objet réel, tel qu'un transformateur électrique, un générateur ou une charge, qui doit être représenté comme faisant partie du modèle complet de réseau électrique dans un EMS par exemple. Parmi les autres types d'objets figurent des éléments tels que les

programmes et les mesures que des applications EMS doivent également traiter, analyser et enregistrer. De tels objets nécessitent une représentation commune pour atteindre les objectifs de compatibilité de connexion et d'interopérabilité de la norme EMS-API. Un objet particulier dans un réseau électrique ayant une identité unique est modélisé comme une instance de la classe à laquelle il appartient.

Il convient également de noter que le CIM est défini pour faciliter l'échange de données. Comme défini dans le présent document, les entités du CIM n'ont pas de comportement. Pour une interface spécifique, un profil est défini et consiste en des sous-ensembles de classes, attributs et associations du CIM. Un profil définit la charge utile de message pour une interface.

Les classes comportent des attributs qui décrivent les caractéristiques des objets. Chaque classe du CIM contient les attributs décrivant et identifiant une instance spécifique de la classe. Seuls les attributs d'intérêt public pour les applications EMS sont inclus dans les descriptions de classes.

Chaque attribut possède un type qui identifie le type d'attribut dont il s'agit. Les attributs typiques sont de type Integer (classe CIM Integer), Float (classe CIM Float), Boolean (classe CIM Boolean), String (classe CIM String), Date (classe CIM Date) et Decimal (classe CIM Decimal), qui sont appelés types primitifs. Cependant, de nombreux types supplémentaires sont définis comme partie intégrante de la spécification CIM. Par exemple, ShuntCompensator a un attribut "maxU" du type Voltage (classe CIM Voltage). La définition de nombreux types partagés est contenue dans le Paquetage Domain que décrit 6.3. Les stéréotypes Primitive, enumeration, CIMDatatype et Compound de l'UML sont ajoutés aux classes utilisées comme types. Le stéréotype CIMDatatype est utilisé avec une sémantique spécifique du CIM pour un triplet d'attributs {value, unit, multiplier}, qui implique une mise en correspondance personnalisée avec des artefacts de sérialisation tels que RDFS, OWL et XSD. Les classes ayant ces stéréotypes ne participent pas à des relations de généralisation ou d'association et sont simplement utilisées comme types pour des attributs. Le stéréotype enumeration est utilisé pour décrire un attribut avec une liste énumérée de choix. Le stéréotype Compound est utilisé pour décrire des ensembles d'attributs liés qui sont couramment réutilisés. Les classes Compound peuvent se composer de Primitive, enumeration, CIMDatatype ou d'autres classes Compound à condition qu'elles ne participent pas à une récursion.

Tous les attributs du CIM sont implicitement facultatifs en ce sens que les profils utilisant le CIM peuvent les éliminer tous.

Les relations entre les classes révèlent la façon dont elles sont structurées l'une par rapport à l'autre. Les classes CIM sont reliées de différentes façons, comme décrit dans les paragraphes ci-dessous.

4.3.2 Generalization

Une generalization (généralisation) est une relation entre une classe générale et une classe plus spécifique. La classe la plus spécifique ne peut contenir que des informations supplémentaires. L'association de type Generalization permet à une classe spécifique d'hériter d'attributs et de relations de toutes les classes plus générales situées au-dessus d'elle.

La Figure 3 est un exemple de generalization (généralisation). Dans cet exemple tiré du paquetage Wires, un Breaker (disjoncteur) est un type plus spécifique de ProtectedSwitch (commutateur protégé), qui à son tour est un type plus spécifique de Switch (commutateur), qui est un type plus spécifique de ConductingEquipment (équipement conducteur), etc. Un PowerTransformer (transformateur de puissance) est un autre type plus spécifique de Equipment. Noter que PowerSystemResource hérite de la classe IdentifiedObject qui ne se trouve pas dans le diagramme, IdentifiedObject est donc représenté en italiques dans l'angle supérieur droit de la classe PowerSystemResource.

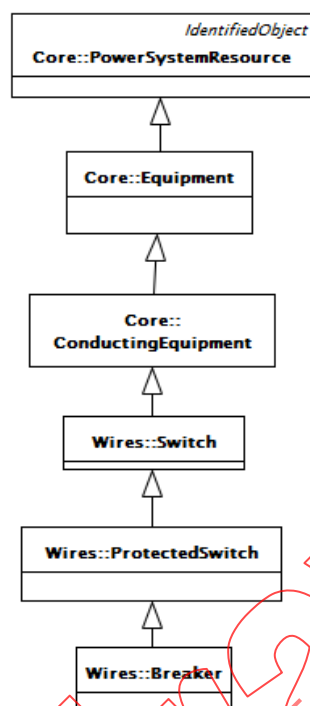
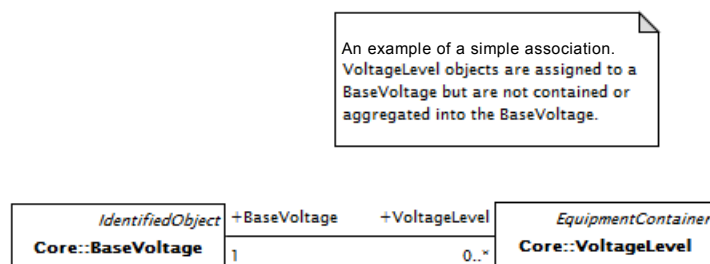


Figure 3 – Exemple de généralisation

4.3.3 Simple Association (Association simple)

Une association est une connexion conceptuelle entre des classes. Chaque association a deux "extrémités d'association". Les "extrémités d'association" étaient appelées "rôles" avant la spécification UML 2.0. Chaque extrémité d'association décrit le rôle que la classe cible (la classe vers laquelle l'extrémité d'association part) a en relation avec la classe source (la classe à partir de laquelle l'extrémité d'association part). La coutume est d'attribuer aux extrémités d'association le nom de la classe cible avec ou sans un qualifieur. Chaque extrémité d'association possède aussi une cardinalité/multiplicité, qui est une indication du nombre d'objets qui peuvent prendre part à la relation décrite. Dans le CIM, les associations ne sont pas nommées, seules les extrémités d'association sont nommées. Par exemple comme le montre la Figure 4, il y a dans le CIM une association entre la classe BaseVoltage et la classe VoltageLevel. La multiplicité est montrée aux deux extrémités de l'association. Dans cet exemple, un objet VoltageLevel (niveau de tension) peut faire référence à 1 BaseVoltage (tension de base), et une BaseVoltage peut être référencée par 0 ou plus d'objets VoltageLevel.



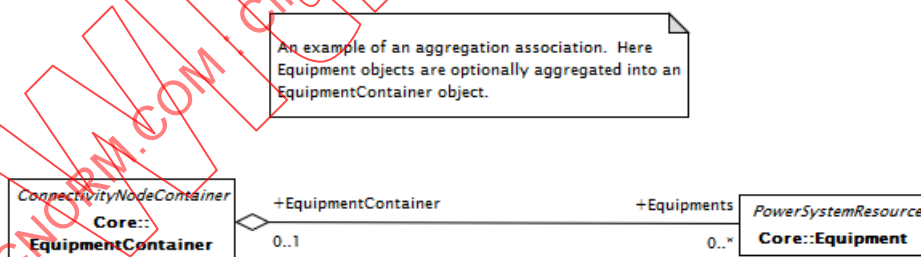
Légende

Anglais	Français
An example of a simple association. VoltageLevel objects are assigned to a BaseVoltage but are not contained or aggregated into the BaseVoltage.	Un exemple d'association simple. Des objets VoltageLevel sont affectés à une BaseVoltage, mais ne sont ni contenus ni agrégés dans la BaseVoltage.

Figure 4 – Exemple d'association simple

4.3.4 Aggregation

Aggregation (agrégation) est un cas spécial d'association. L'agrégation indique que la relation entre les classes est une sorte de relation classe entière/classe partielle où la classe entière "consists of" (se compose de) ou "contains" (contient) la classe partielle, et cette classe partielle est une "part of" (partie de) la classe entière. La classe partielle n'hérite pas de la classe entière comme dans la généralisation. La Figure 5 représente une agrégation entre la classe EquipmentContainer (Conteneur d'équipements) et la classe Equipment, qui provient du paquetage Core. Comme on le voit, un Equipment peut être un élément de un ou zéro objet EquipmentContainer mais l'objet EquipmentContainer peut contenir un nombre quelconque d'objets Equipment. Dans le cadre de l'utilisation du CIM comme modèle d'information, l'agrégation n'a pas d'interprétation précise ou formelle au-delà d'une association simple, et permet d'aider à représenter visuellement une utilisation normale.



Légende

Anglais	Français
An example of an aggregation association. Here Equipment objects are optionally aggregated into an EquipmentContainer object.	Un exemple d'association d'agrégation. Ici, des objets Equipment sont facultativement agrégés en un objet EquipmentContainer.

Figure 5 – Exemple d'agrégation

4.4 Concepts et exemples de modèles CIM

4.4.1 Concepts

Les classes, les attributs, les types et les relations du CIM sont spécifiés dans les Articles 5 et 6. L'Article 6 comprend une description complète du modèle de la Base du CIM de la

CEI 61970-301. Afin d'aider à comprendre comment interpréter le CIM, certains concepts-clés de modèle utilisés dans le CIM sont présentés et décrits dans les paragraphes suivants.

4.4.2 Hiérarchies d'emboîtement, d'équipements et dénomination

4.4.2.1 Structure de l'emboîtement

La Figure 6 montre le concept de conteneurs d'équipement (EquipmentContainer) pour former des hiérarchies dans le CIM. Les conteneurs d'équipement sont un moyen de regrouper et de nommer des équipements se trouvant, typiquement, dans un poste (substation). Comme on peut le constater de visu, les conteneurs offrent une certaine souplesse d'utilisation pour se plier aux différents usages internationaux, ainsi qu'aux différences trouvées habituellement entre les postes de transport et de distribution. Bay (une cellule), VoltageLevel (un niveau de tension), Substation (un poste), Line (une ligne) et Plant (une centrale) sont tous des types de EquipmentContainers. En général, une Bay est contenue dans un VoltageLevel spécifique, qui à son tour est contenu dans un Substation. Les Substation et les Line peuvent être contenus dans des SubGeographicRegion (des sous-régions géographiques) et GeographicRegion (des régions géographiques).

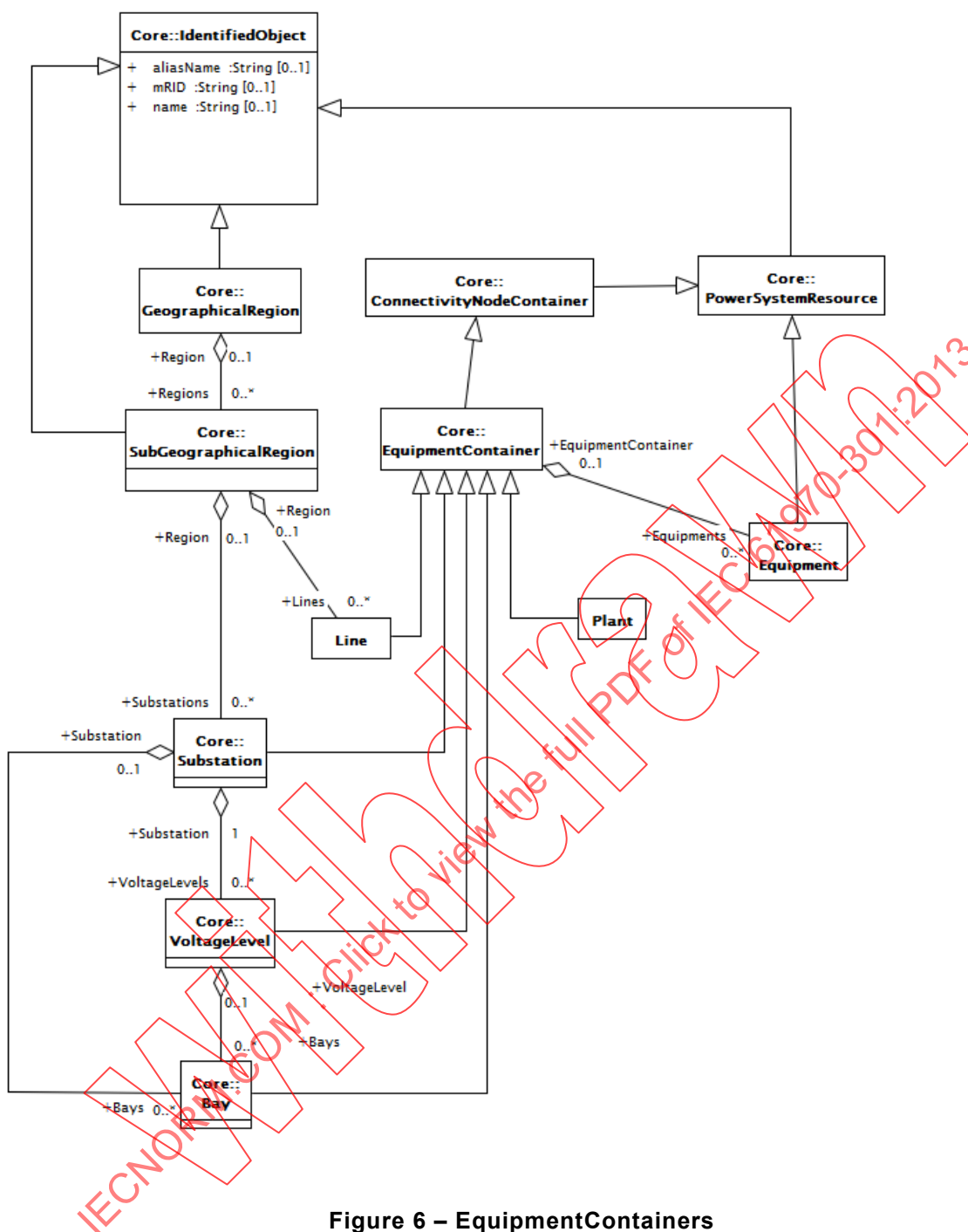


Figure 6 – EquipmentContainers

Les diagrammes NamingHierarchyPart1 (Partie 1 de la hiérarchie de dénomination) et NamingHierarchyPart2 (Partie 2 de la hiérarchie de dénomination) dans les paquetages Wires montrent la hiérarchie de dénomination fonctionnelle (se référer à la documentation du paquetage Wires donnée à en 6.8 pour les détails).

Une hiérarchie d'emboîtement est utilisée avec la classe IdentifiedObject pour créer une désignation hiérarchique destinée à décrire la consommation humaine. Cette hiérarchie est utilisée spécifiquement pour nommer les équipements selon leur fonction dans le système de puissance. Il s'agit de la hiérarchie de dénomination fonctionnelle. Outre les noms fonctionnels, d'autres identifications communes sont les numéros de série des biens. Le nom fonctionnel diffère d'un numéro de série en ce qu'il renvoie à la fonction d'une position ou localisation particulière d'un équipement dans le système de puissance. Quel que soit l'élément spécifique d'équipement physique qui est placé en un endroit, le nom fonctionnel est

le même, mais le numéro de série varie en fonction de l'équipement physique actuellement utilisé.

4.4.2.2 Classe IdentifiedObject

La classe IdentifiedObject contenu dans le paquetage Core est héritée par tous les objets PowerSystemResource et de nombreuses autres classes. Cette classe comporte les attributs et les associations à utiliser pour nommer tous les objets du CIM. L'attribut mRID de la classe IdentifiedObject est un moyen direct et rigoureux pour l'identité des objets du CIM.

Les éléments qui suivent sont des définitions et conventions concernant les modalités d'utilisation des attributs IdentifiedObject pour nommer les objets PowerSystemResource (pour plus de détails, se référer à la documentation concernant IdentifiedObject et ses attributs à l'Article 6):

- mRID (Master Resource ID): Un identificateur lisible par une machine et unique au niveau global pour une instance d'objet.
- name: Le nom est un texte, librement lisible par l'homme et pouvant être non unique, qui nomme l'objet.
- aliasName: Cet attribut est déconseillé, les classes Name de 4.4.3 Modèle de Names sont une meilleure solution. L'aliasName est un nom de l'objet en texte libre lisible par l'homme en remplacement de IdentifiedObject.name. Il peut ne pas être unique et peut ne pas se corrélér à une hiérarchie de dénomination. L'attribut aliasName a été retenu parce qu'il est compatible avec les précédentes versions du CIM. Il est cependant recommandé de remplacer aliasName par la classe Name puisqu'il est prévu de supprimer aliasName à l'avenir.

4.4.3 Modèle de Names

Un modèle de dénomination souple et extensible est fourni pour spécifier des noms différents pour les objets. La Figure 7 montre le diagramme de classe pour des noms de remplacement et la manière dont ils peuvent être associés à un type de nom défini par un utilisateur spécifique. Ce modèle permet de définir spécifiquement les noms dans un domaine particulier et ne fait pas intervenir de règles spécifiques de dénomination. Les concepts de noms de remplacement, de noms de chemin et de noms locaux peuvent tous être utilisés avec ce modèle. Les modèles du CIM n'utilisent ni ne favorisent des conventions spécifiques de dénomination mais permettent d'échanger ces conventions de dénomination dans un contexte clair.

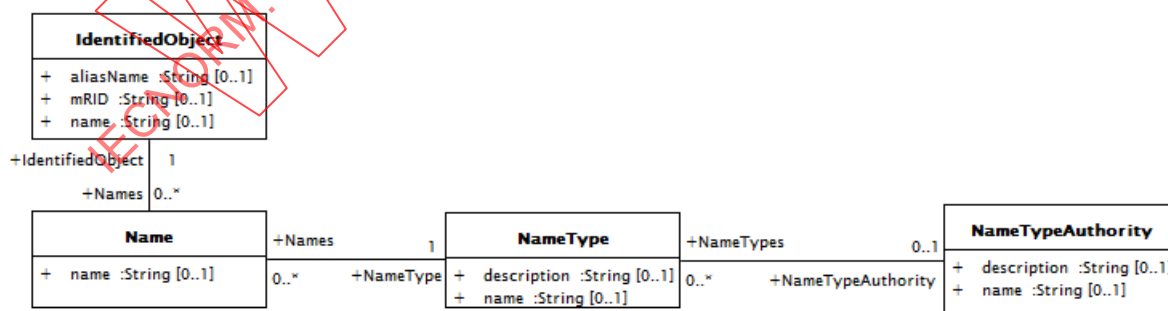


Figure 7 – Names

Tableau 1 – Conventions de dénomination de la classe NameType (type de nom)

name	description
ICCP	Réservé aux noms ICCP (TASE-2). Utilisée pour décrire les noms de point ICCP pour la classe MeasurementValue et les noms de source ICCP pour la classe MeasurementValueSource.

Le Tableau 1 donne différentes conventions de nom pour les types de nom. Les colonnes du Tableau 1 ont la signification suivante:

name: valeur de l'attribut name.

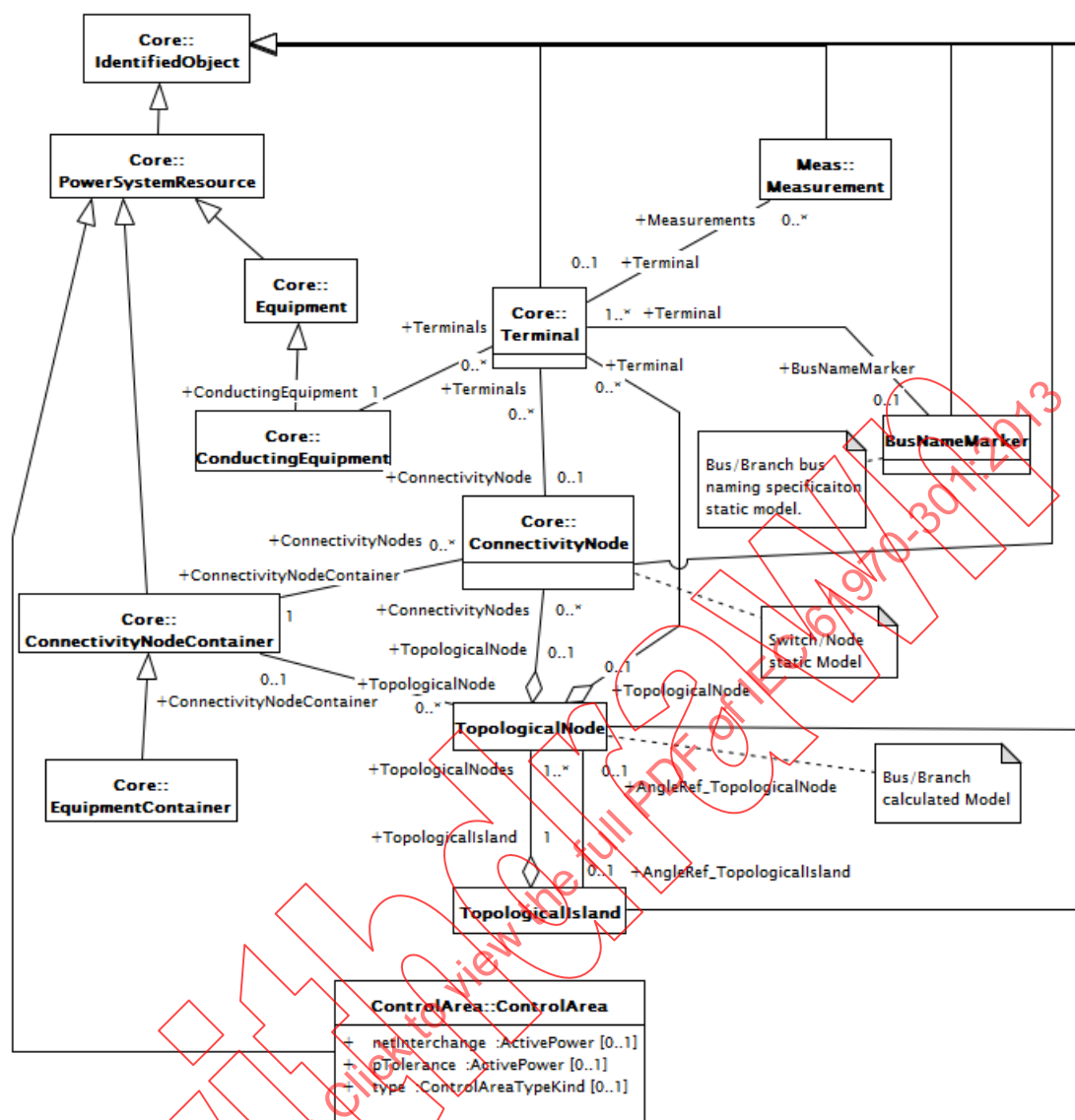
description: utilisation du type de nom.

Noter que le Tableau 1 ne donne pas de liste exhaustive des valeurs qui peuvent être utilisées pour l'attribut nom de la classe Name.

4.4.4 Modèle de connectivité

4.4.4.1 Description de la connectivité

La Figure 8 montre le diagramme de classe Topology qui modélise la connectivité entre différents types de ConductingEquipment. Elle inclut également une partie du diagramme de classe du paquetage Meas traitant des mesures pour illustrer la façon dont les mesures sont associées aux équipements conducteurs.



Légende

Anglais	Français
Bus/Branch bus naming specification static model.	Modèle statique de spécification de dénomination de topologie nodale
Switch/Node static model	Modèle statique Commutateur/Nœud
Bus/Branch calculated model	Modèle calculé à topologie nodale

Figure 8 – Modèle de connectivité

Pour modéliser la connectivité, les classes Terminal (borne) et ConnectivityNode (nœud de connectivité) ont été définies. Une Terminal appartient à un ConductingEquipment, mais un ConductingEquipment peut avoir un nombre quelconque de Terminal (bornes). Chaque Terminal peut être connecté à un ConnectivityNode qui est un point où les bornes des équipements conducteurs sont connectées ensemble avec une impédance nulle. Un ConnectivityNode peut avoir un nombre quelconque de bornes connectées et peut être membre d'un TopologicalNode (nœud topologique) (à savoir un bus) qui est membre d'un TopologicalIsland (îlot topologique). Les TopologicalNode et TopologicalIsland sont créés comme résultat d'un programme qui prend en compte la topologie existante et les positions réelles des Switch (commutateurs).

Il est possible d'échanger des messages impliquant directement `TopologicalNode` et contournant les détails de `ConnectivityNode` en utilisant l'association `TopologicalNode` à `Terminal`. Cela est souvent utile pour les échanges impliquant des modèles de topologie nodale qui ne contiennent pas typiquement des détails relatifs aux commutateurs. Quand le modèle `TopologicalNode` est échangé, il est utile d'échanger aussi l'attribut "connected" (connecté) dans la classe `Terminal`. Cela permet de retenir la structure topologique du modèle de topologie nodale mais indique qu'une borne n'est pas entièrement connectée. Cette déconnexion résulte normalement de ce qu'un ou plusieurs commutateurs s'ouvrent dans le modèle détaillé qui n'est pas représenté dans un échange de modèle de topologie nodale.

Les `EquipmentContainers`, qui sont une spécialisation d'un `ConnectivityNodeContainer` (Conteneur de nœud de connectivité), peuvent avoir zéro à plusieurs de `ConnectivityNodes`. Les associations `ConductingEquipment-Terminal` et `Terminal-ConnectivityNode` permettent de représenter la topologie d'un réseau électrique réel. Pour chaque `Terminal` connecté à un `ConnectivityNode`, les associations des autres `Terminal(s)` connectées au même `ConnectivityNode` identifient le ou les objet(s) `ConductingEquipment` qui sont connectés électriquement. Une connectivité similaire, au niveau de détail "topologie nodale", peut être exprimée à l'aide de `TopologicalNode` au lieu du `ConnectivityNode`. La classe `BusNameMarker`, qui a une association à `ConnectivityNode`, est utilisée pour contenir les noms de bus utilisateur devant être appliqués aux `TopologicalNodes`. Les `TopologicalNodes` sont créés à partir de `ConnectivityNodes` et résultent du traitement de la topologie, il est possible d'appliquer de nombreux noms en fonction des instances de `ConnectivityNode` en présence.

Pour modéliser les valeurs analogiques telles que la tension et la puissance, chaque `Terminal` a une association avec une classe `Measurement` du paquetage `Meas`. Bien que la Figure 8 ne le montre pas, un objet `Measurement` est associé à au moins un objet `MeasurementValue` (valeur de mesure). Chaque `MeasurementValue` est une instance de mesure d'une source spécifique, par exemple une mesure télémetrée.

L'Article 6 contient une description complète de chaque classe de la Figure 8 avec la définition de tous les attributs et de toutes les relations pris en charge dans chaque classe.

4.4.4.2 Exemple de connectivité et d'emboîtement

Pour illustrer comment le modèle de connectivité et le modèle d'emboîtement peuvent être représentés comme des objets, un petit exemple est présenté à la Figure 9. L'exemple montre une ligne de transport avec une jonction en T entre deux postes et un poste avec deux niveaux de tension et un transformateur entre eux. La ligne de transport contient deux câbles différents. Un des niveaux de tension est montré avec un tronçon de jeux de barres contenant un seul jeu de barres et deux baies de commutation très simples connectées au jeu de barres.

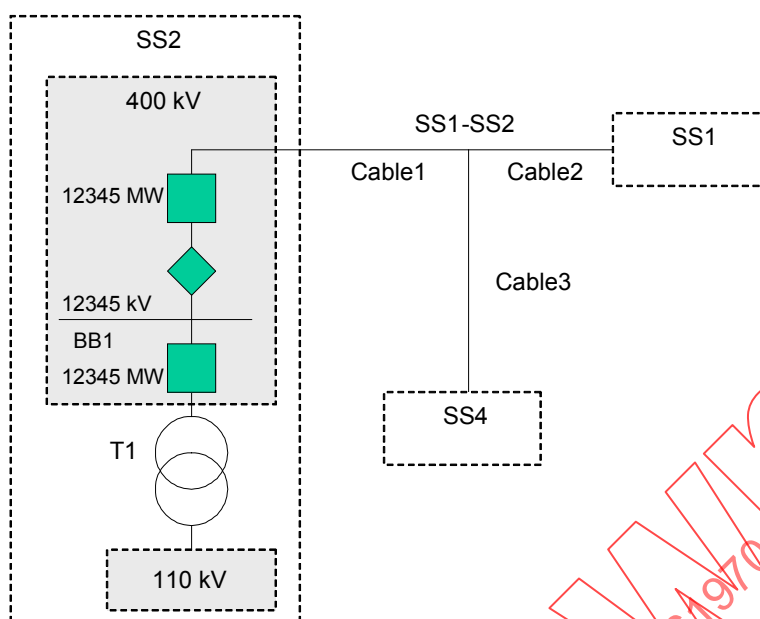


Figure 9 – Exemple de réseau simple

La Figure 10 montre comment la connectivité est modélisée dans le CIM ainsi qu'une façon (mais pas nécessairement la seule) dont l'emboîtement est modélisé pour le diagramme de la Figure 10. Les carrés ombrés représentent les EquipmentContainers, et les carrés blancs les ConductingEquipments. Un ombrage plus sombre indique que l'EquipmentContainer est plus haut dans la hiérarchie d'emboîtement (à savoir, Substation est le plus haut, VoltageLevel vient ensuite, etc.). Les cercles blancs représentent les ConnectivityNodes, et les petits cercles noirs les Terminal. Une Terminal appartient à un ConductingEquipment, et un ConnectivityNode appartient à un EquipmentContainer. Cela signifie que les frontières (ou points de contact) entre ConductingEquipments sont leurs Terminal interconnectées via les ConnectivityNode.

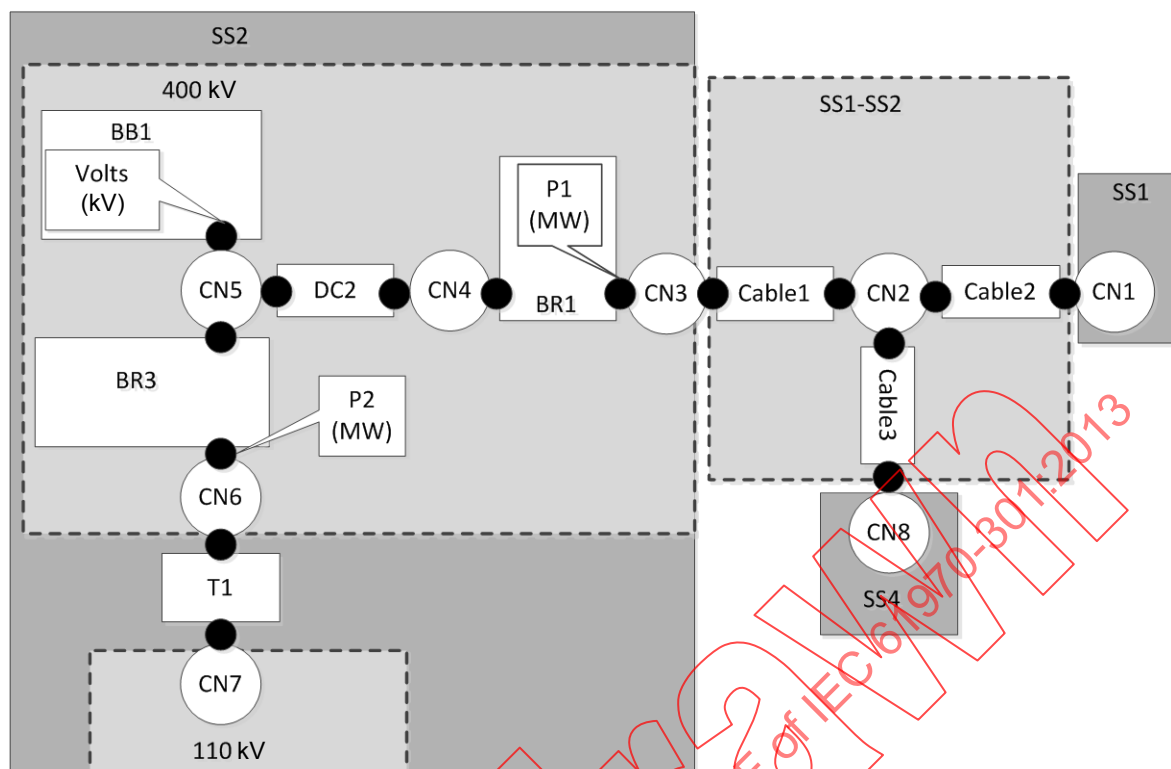


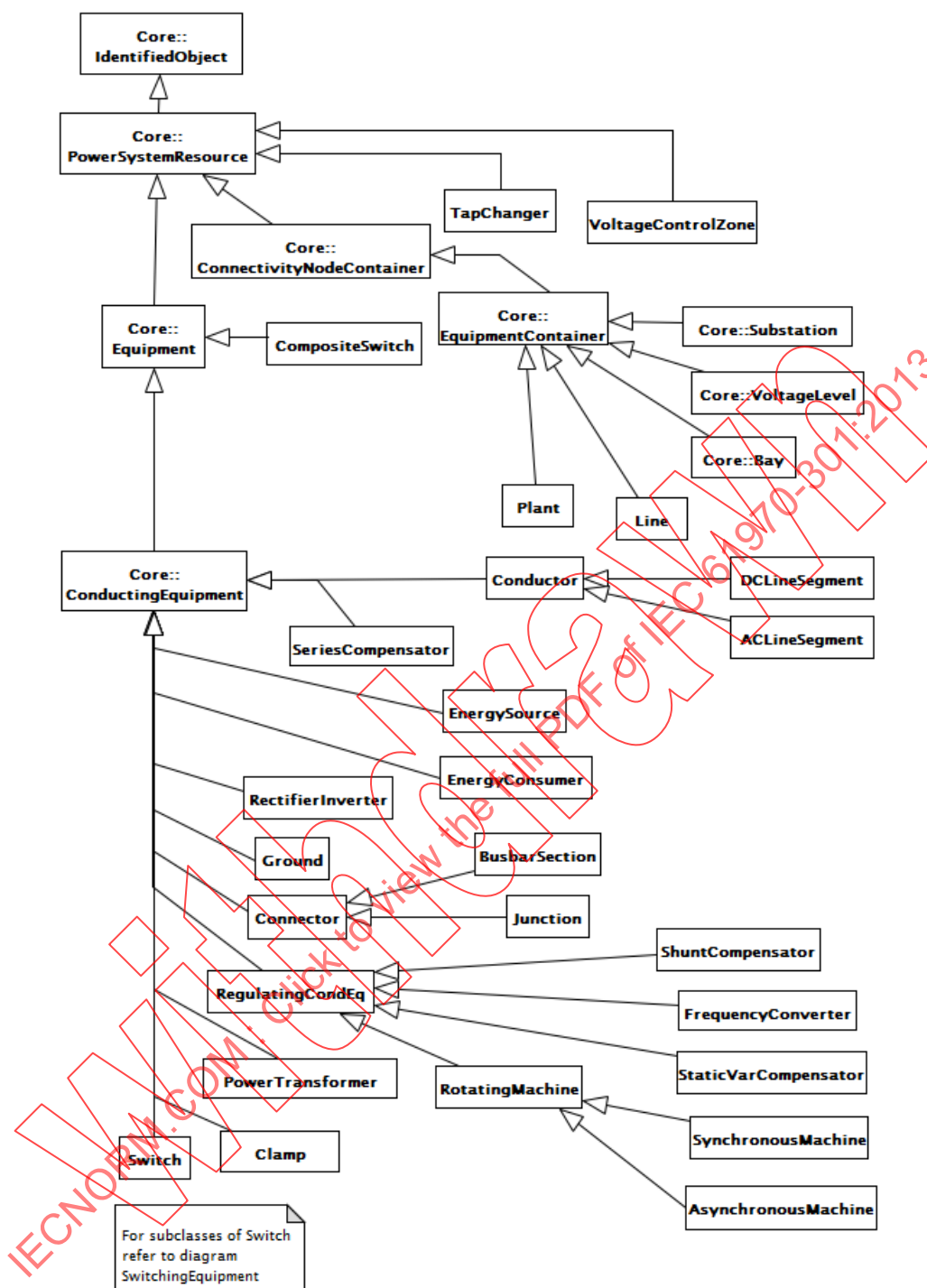
Figure 10 – Connectivité d'un réseau simple modélisé avec la topologie du CIM

La Line (c'est-à-dire ligne) SS1-SS2 contient trois ACLineSegments (Cable1, Cable2, et Cable3) et le ConnectivityNode (CN2) associé pour modéliser une jonction en T, qui assure une connexion à SS4. Cela n'est qu'une façon dont cette configuration pourrait être modélisée. Chaque ACLineSegment a deux Terminal. Cable1 est connecté à CN3 et CN2 via ces Terminal. CN3 est contenu par VoltageLevel 400 kV. Le disjoncteur ("Breaker") BR1 a deux bornes dont l'une est connectée à CN3.

Les Measurements sont représentées par des bulles carrées où les flèches pointent vers une Terminal. P1 est connecté à la Terminal de droite appartenant au Breaker BR1. Noter que P1 est tracé dans la boîte représentant BR1. C'est parce qu'une Measurement peut appartenir à une PowerSystemResource (PSR), comme c'est le cas avec BR1. P2 est tracé dans le VoltageLevel 400 kV, ce qui signifie qu'il appartient au VoltageLevel 400 kV et non à BR3.

4.4.5 Hiérarchie des héritages

La Figure 11 montre une vue générale de la hiérarchie des héritages modélisée dans le CIM. Cette vue générale qui est incluse comme l'un des diagrammes du paquetage Wires couvre plusieurs paquetages du CIM.



Légende

Anglais	Français
For subclasses of Switch refer to diagram SwitchingEquipment	Pour les sous-classes de Switch, se référer au diagramme SwitchingEquipment

Figure 11 – Hiérarchie des héritages de Equipment

4.4.6 Modèle de transformateur

La Figure 13 montre une partie du paquetage Wires dans une classe diagramme qui modélise un dispositif PowerTransformer (Transformateur de puissance).

Le modèle de transformateur a été révisé par rapport aux versions antérieures pour qu'il soit possible d'utiliser le même modèle d'instance à la fois dans le modèle équilibré et le modèle déséquilibré. Le PowerTransformer est en outre désormais en soi un ConductingEquipment qui a potentiellement plusieurs bornes pour modéliser plus directement des transformateurs qui comportent une, deux, trois bornes ou davantage. Comme cela est typique pour un ConductingEquipment, quand il est en service le PowerTransformer conduit le courant d'une de ses bornes à une autre.

Comme le montre la Figure 12, le PowerTransformer est aussi facultativement en mesure de modéliser des détails de cuve (Tank), on peut l'utiliser pour décrire en détail les déséquilibres et les connexions des phases de l'enroulement intérieur du transformateur. Dans tous les cas, un PowerTransformer modélise un groupe d'appareils physiques qui agissent conjointement pour transformer le courant sur les bornes et dans un endroit physique. Il arrive souvent que pour des systèmes de transport une instance de PowerTransformer représente trois appareils physiques monophasés. Il est recommandé de modéliser en outre les objets TransformerTank s'il est nécessaire d'obtenir des détails sur les appareils monophasés en question.

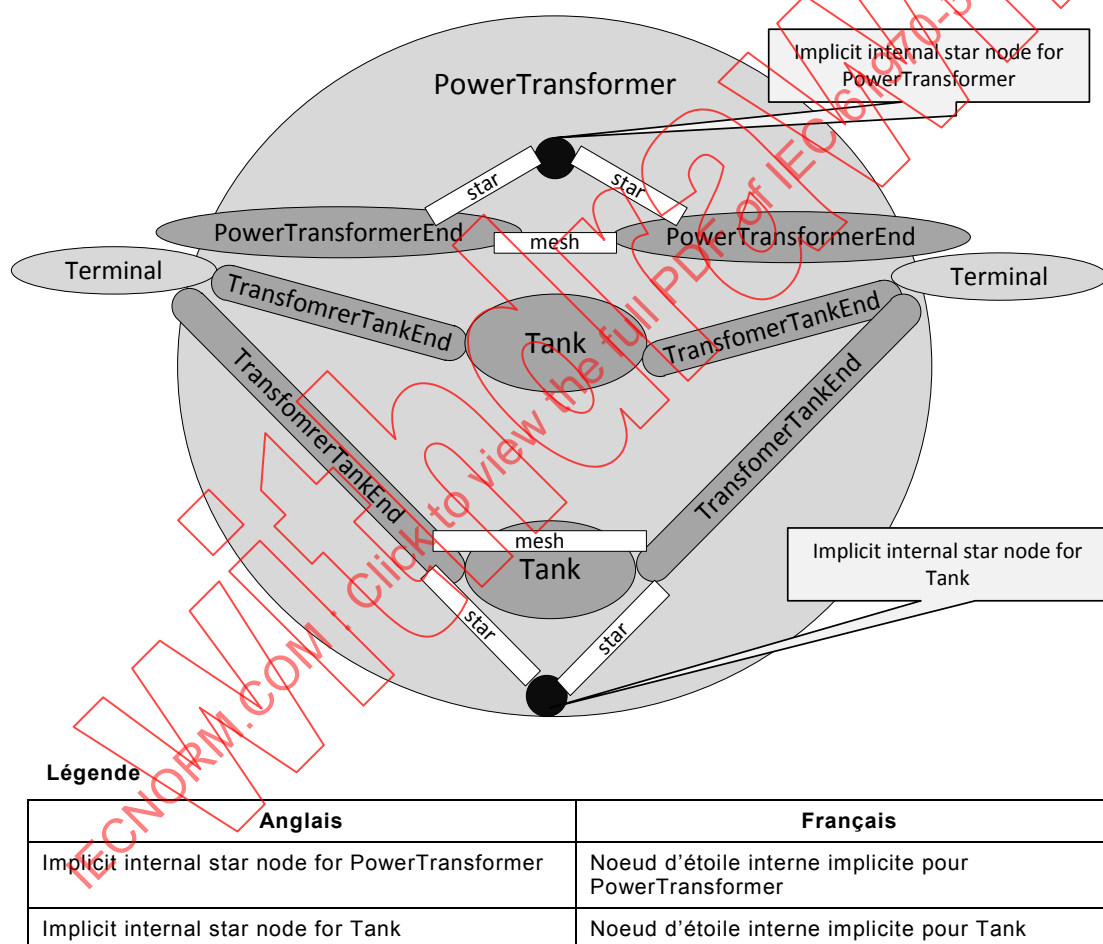


Figure 12 – Modèle de Transformer et de Tank

Un PowerTransformer et un TransformerTank peuvent tous deux avoir l'impédance décrite soit dans une connexion étoile supposée avec la connexion centrale implicite soit sous forme d'impédance de maille en utilisant la classe facultative TransformerMeshImpedance. La forme en maille de l'impédance est nécessaire pour modéliser correctement des transformateurs qui ont plus de trois bornes. Si la forme d'impédance en maille est utilisée, il faudrait spécifier la classe TransformerMeshImpedance pour chaque connexion possible borne à borne, donc une instance pour deux bornes, trois pour trois bornes, six pour quatre bornes et ainsi de suite. Si on utilise une connexion étoile, les paramètres d'impédance peuvent être spécifiés directement dans la classe PowerTransformerEnd, une autre solution consiste à les répartir entre les transformateurs en utilisant la classe TransformerStarImpedance.

Comme représenté, un PowerTransformer est une classe spécialisée de ConductingEquipment, elle-même classe spécialisée d'Equipment. Ceci est représenté par l'utilisation du type généralisation de la relation qui utilise une flèche pour pointer vers la classe générale. Le ConductingEquipment est représenté comme une classe spécialisée d'Equipment en utilisant le nom de classe "*Equipment*" en italiques dans l'angle supérieur droit. L'héritage permet à PowerTransformer d'hériter des attributs de ConductingEquipment, d'Equipment et de toutes les généralisations d'Equipment que ne représente pas ce diagramme.

Un PowerTransformer a aussi des relations avec PowerTransformerEnd et TransformerTank qui sont modélisés avec des relations de type association. Comme le montre la figure, un PowerTransformer peut être associé à zéro PowerTransformerEnds ou davantage mais un PowerTransformerEnd ne peut être associé qu'à zéro ou un PowerTransformer seulement. De façon similaire, le PowerTransformer peut être associé à zéro objets TransformerTank ou davantage mais un TransformerTank ne peut être associé qu'à zéro ou un PowerTransformer seulement.

Le PowerTransformerEnd spécialise la classe TransformerEnd qui a aussi d'autres relations:

- une relation de généralisation avec IdentifiedObject;
- deux relations d'association avec la classe TransformerMeshImpedance, faisant qu'un objet TransformerEnd peut être de (From) 0, 1 ou plusieurs objets TransformerMeshImpedance et vers (To) 0, 1 ou plusieurs objets TransformerMeshImpedance;
- des relations d'agrégation avec les classes PhaseTapChanger et RatioTapChanger, faisant qu'un objet TransformerWinding peut avoir 0 ou 1 objet PhaseTapChanger et 0 ou 1 objet RatioTapChanger qui lui sont associés.

L'Article 6 contient une description complète de chaque classe de la Figure 13 avec la définition de tous les attributs et de toutes les relations pris en charge dans chaque classe.

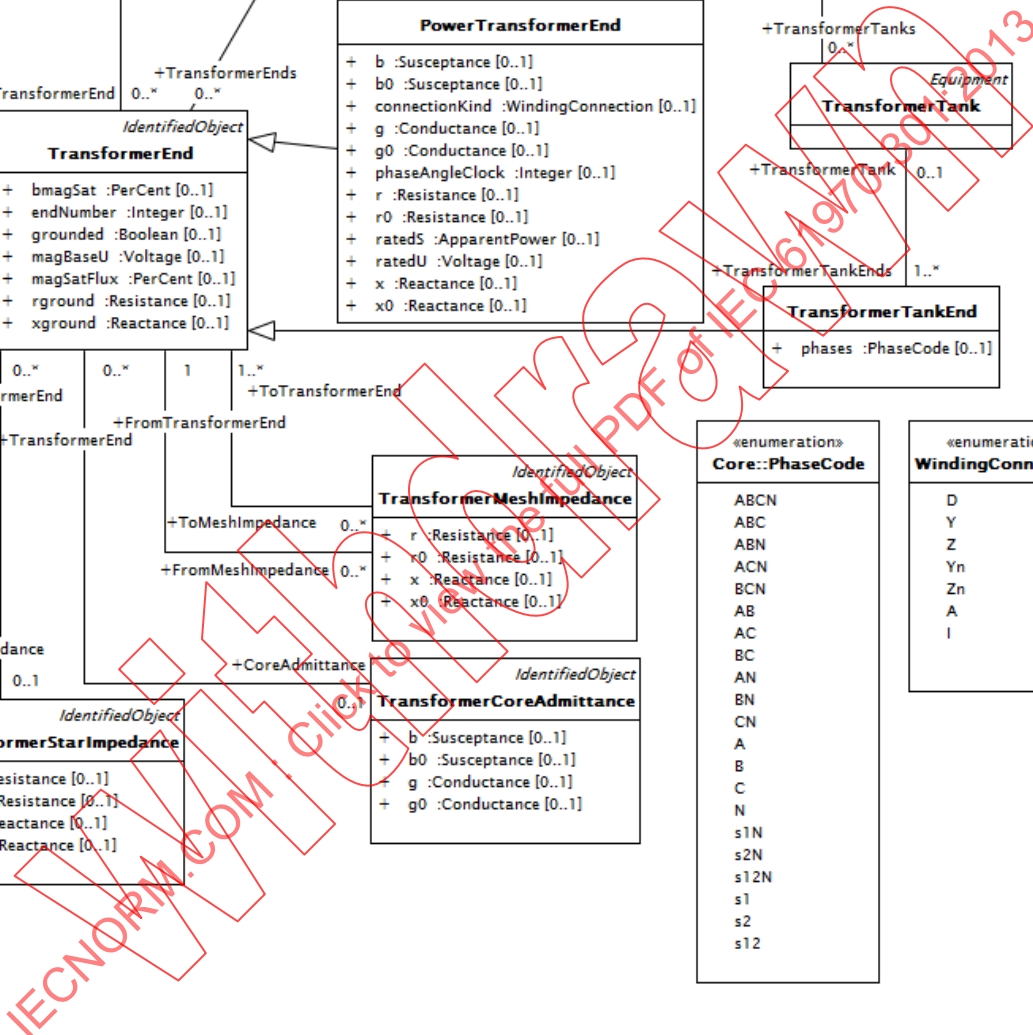


Figure 13 – Modèle de transformateur

4.4.7 Modélisation de prise du transformateur

4.4.7.1 Modélisation généralisée de prise

Plusieurs types de modèles de prise de transformateur sont pris en charge par le modèle CIM. En fonction de la structure physique, les effets du mouvement de prise seront définis en utilisant la classe `RatioTapChanger` ou la classe `PhaseTapChanger` et ses différentes spécialisations. Le `PhaseTapChanger` peut être spécialisé dans un certain nombre de types qui représentent la structure physique et donc le comportement. Les attributs de `TapChanger` sont applicables à toutes les spécialisations de `RatioTapChanger`. La Figure 14 représente le modèle UML pour les prises de transformateur.

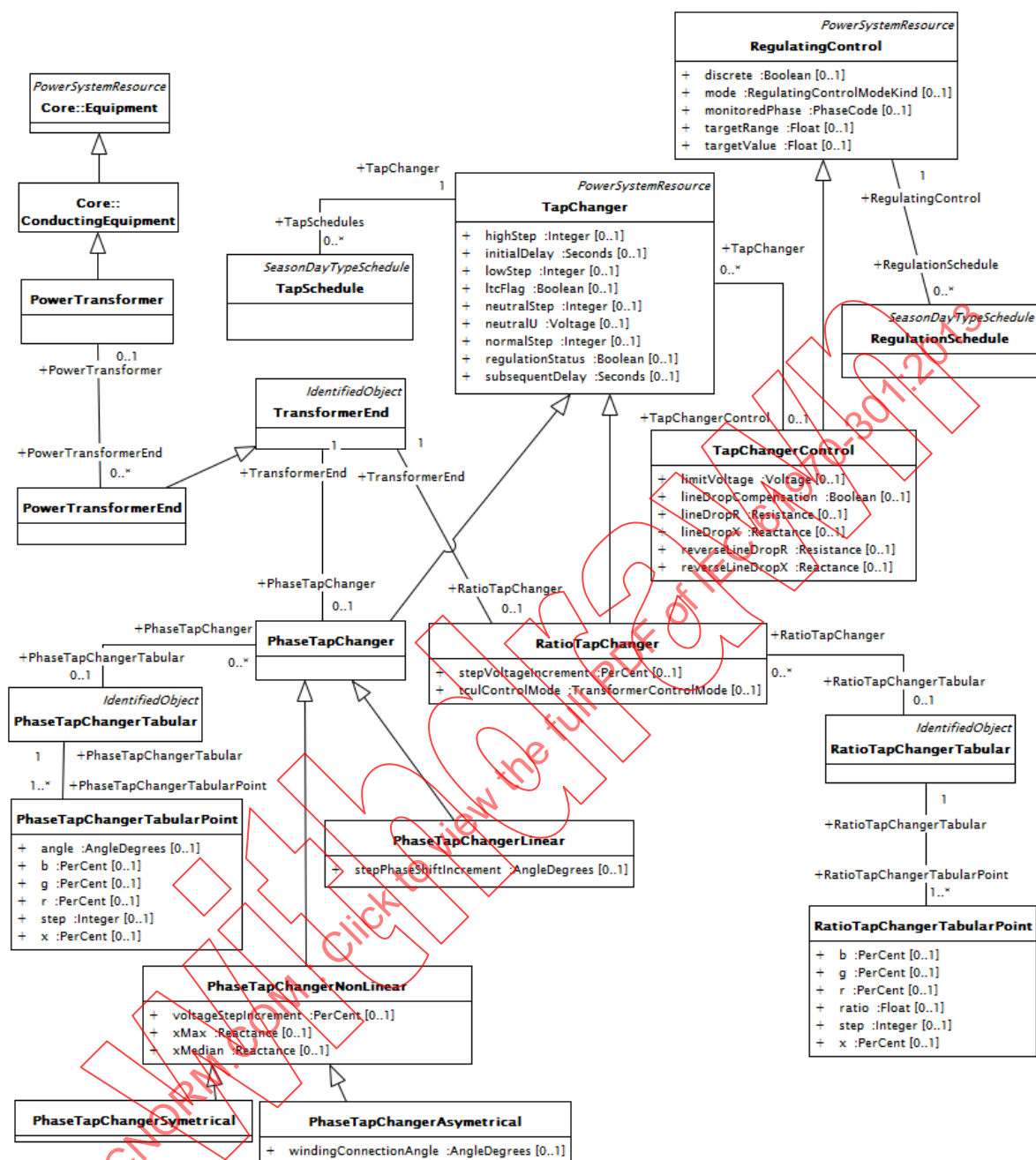


Figure 14 – Modèle de prise de transformateur

Plusieurs changeurs de prise peuvent être combinés à l'intérieur d'un transformateur de puissance mais un seul changeur de prise déphaseur et un seul changeur de prise qui modifie le rapport sont censés être modélisés sur une borne donnée. Le changement de rapport ou le déphasage sont documentés par rapport à l'autre ou aux autres borne(s) en supposant que ces bornes ont le rapport nominal (1 PU) et le déphasage nominal (zéro). Il est permis de combiner à la fois un changeur de prise déphaseur et un changeur de rapport de prise sur une borne bien que cette pratique ne soit pas recommandée si un modèle de transformateur asymétrique est une représentation appropriée.

On peut utiliser des représentations tabulaires à la fois pour des transformateurs déphaseurs et des transformateurs de rapport quand les modèles disponibles du CIM ne reflètent pas le comportement physique de manière appropriée. Les modèles tabulaires sont ajoutés en option aux autres modèles de prise du CIM. Il est recommandé d'utiliser les modèles linéaires

quand il s'agit d'ajouter des modèles tabulaires. Des modèles tabulaires peuvent être utilisés sur des appareils déphaseurs non linéaires ou des appareils changeurs de prise pour décrire le changement d'impédance dans la plage de prise.

4.4.7.2 Modélisation d'un transformateur de tension

La classe RatioTapChanger est utilisée pour définir une prise de transformateur qui change le rapport de la valeur de tension sans changer l'angle de phase à travers le transformateur. Un RatioTapChanger peut être utilisé dans des cas où le transformateur introduit intrinsèquement un déphasage fixe, comme à partir d'une connexion d'enroulement en étoile-triangle.

4.4.7.3 Modélisation tabulaire d'un transformateur de tension

Un RatioTapChangerTabular peut être facultativement associé à n'importe quel RatioTapChanger et représenter une relation non linéaire entre le nombre de prises, le rapport de prises et les impédances. S'il est associé, le modèle tabulaire de rapport a normalement la priorité.

4.4.7.4 Modélisation d'un transformateur déphaseur linéaire

Un transformateur déphaseur est linéaire si l'incrément (accroissement) de déphasage par échelon de prise est uniforme sur toute la plage du changeur de prise. Un déphaseur linéaire est modélisé en utilisant la classe PhaseTapChangerLinear.

4.4.7.5 Modélisation d'un déphaseur symétrique

Une construction de déphaseur symétrique est modélisée selon une classe PhaseTapChangerSymmetrical et les modifications apportées à la valeur de la tension et à la phase sur toute la plage des positions de prise pour un déphaseur symétrique sont représentées à la Figure 15. Un déphaseur symétrique modifiera l'angle de phase à travers le transformateur mais ne modifiera pas le rapport de tension.

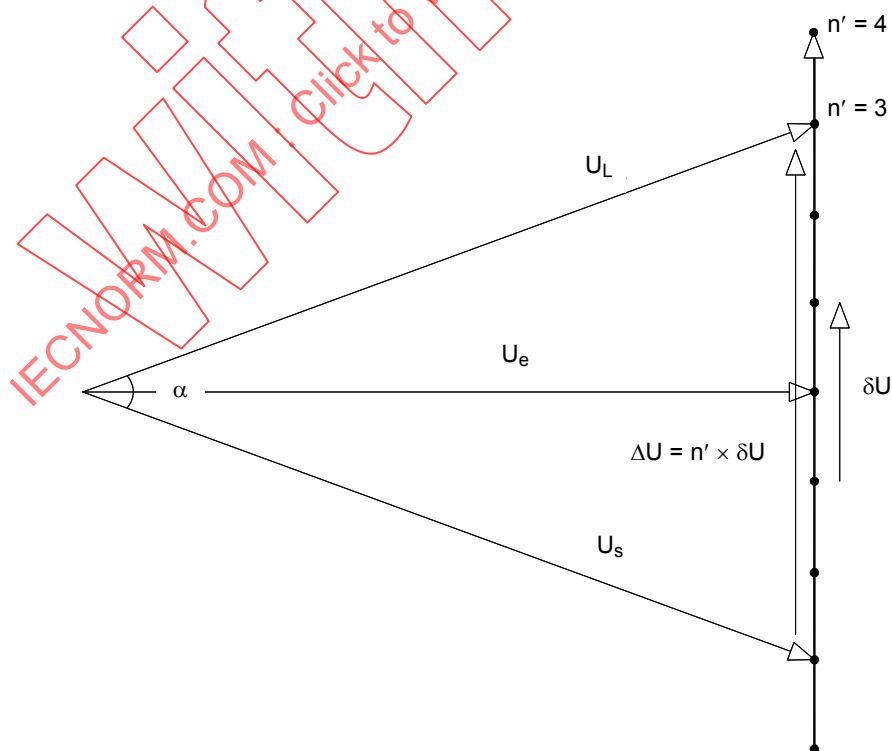


Figure 15 – Déphaseur symétrique

L'angle de déphasage peut être représenté comme étant:

$$\alpha = 2 \arctan \frac{\Delta U}{2U_e} = 2 \arctan \frac{n' \delta U}{2U_e}$$

$$\rho = 1$$

où

- α est l'angle de déphasage, borne locale par rapport à l'autre borne nominale;
- n' est la position actuelle de la prise;
- ΔU est le changement de tension en quadrature;
- U_e est la tension à la prise neutre;
- δU est le changement de tension pour une position de la prise;
- ρ est le rapport de transformation de la valeur de tension, de la borne locale à l'autre borne nominale;
- U_L est la tension de la borne locale;
- U_S est la tension du système ou de l'autre borne.

Le signe du terme $n' \delta U$ est utilisé pour indiquer le déphasage relatif à la borne associée. Si $n' \delta U$ est positif, l'angle de phase de la borne associée augmente. Si δU est négatif, l'angle de phase de la borne associée diminue.

4.4.7.6 Modélisation d'un déphaseur asymétrique

Un déphaseur asymétrique amène le rapport de tensions à varier à mesure que le déphasage varie. Les variations de l'amplitude et de la phase de la tension sur toute la plage des échelons des prises pour un déphaseur asymétrique sont montrées à la Figure 16.

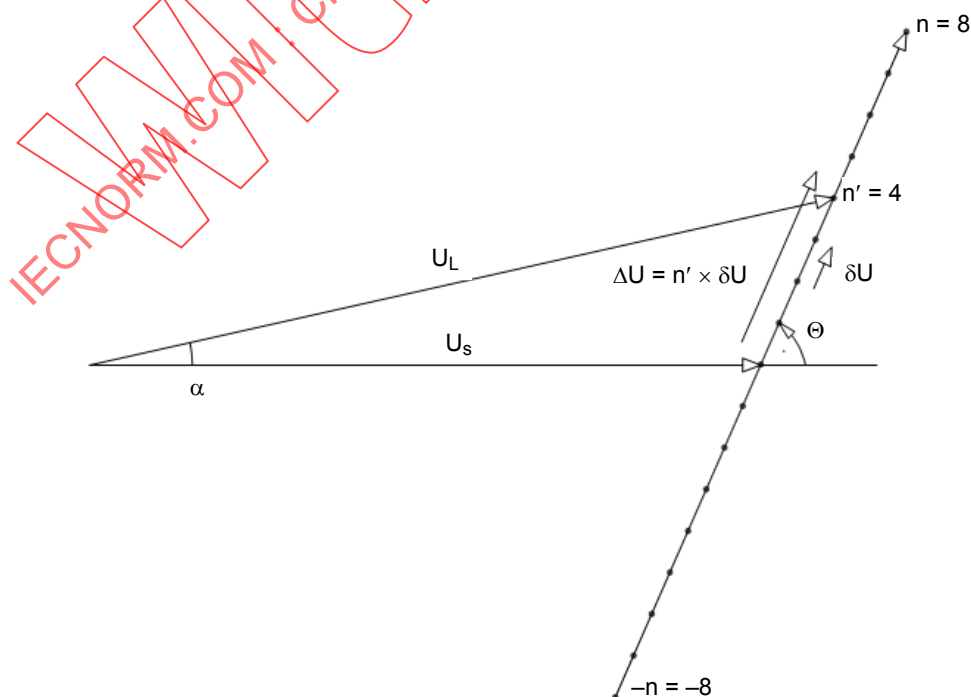


Figure 16 – Déphaseur asymétrique

L'angle de déphasage peut être représenté comme étant:

$$\alpha = \arctan\left(\frac{n' \delta U \sin \Theta}{U_S + n' \delta U \cos \Theta}\right)$$

$$\rho = \frac{U_S}{\sqrt{(n' \delta U \sin \Theta)^2 + (U_S + n' \delta U \cos \Theta)^2}}$$

Où

- α est l'angle de déphasage, borne locale par rapport à l'autre borne nominale;
- n' est la position actuelle de la prise;
- δU est le changement de tension pour une position de la prise;
- Θ est l'angle de connexion de l'enroulement;
- ρ est le rapport de transformation de la valeur de tension, de la borne locale à l'autre borne nominale;
- U_L est la tension de la borne locale;
- U_S est la tension du système ou de l'autre borne.

Un déphaseur asymétrique est modélisé dans la classe `PhaseTapChangerAsymetrical`. Le `PhaseTapChanger.windingConnectionAngle` doit être un multiple de 30 degrés: -90° , -60° , -30° , 30° , 60° , 90° ...). Un déphaseur asymétrique avec un angle de connexion d'enroulements de 90° n'est pas identique à un déphaseur symétrique.

4.4.7.7 Modélisation tabulaire du transformateur déphaseur

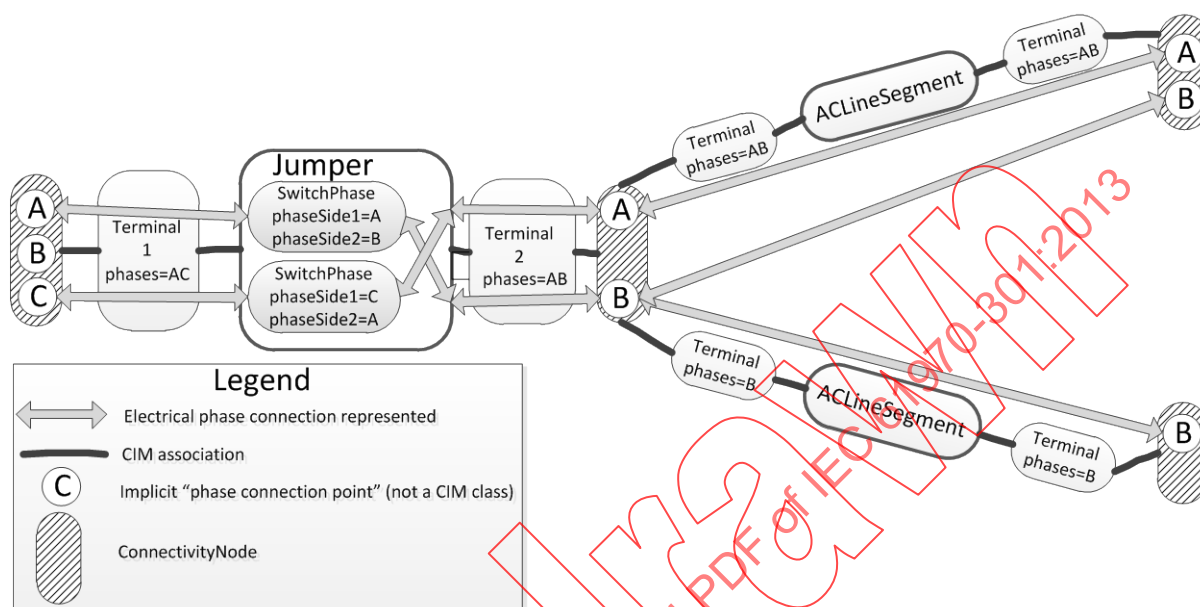
Un `PhaseTapChangerTabular` peut être facultativement associé à n'importe quel `PhaseTapChanger`, il représente une relation non linéaire entre le nombre de prises, le déphasage et les impédances. Un `PhaseTapChangerTabular` peut être utilisé pour décrire uniquement le changement d'impédance sur la plage du changeur de prise. Dans cette situation, l'attribut de déphasage est nul ou omis. S'il est associé, le modèle tabulaire de déphaseur a normalement la priorité.

4.4.8 Modélisation de la phase

Les modèles du CIM fournissent des moyens de spécifier la représentation unifilaire équivalente de réseaux triphasés équilibrés ou, en sus ou en option, de spécifier le détail d'une phase donnée. Dans les réseaux monophasés, il y a correspondance directe entre les objets `Terminal` et les connexions physiques des fils. Dans une représentation unifilaire de réseaux triphasés équilibrés, nous utilisons un `Terminal` pour représenter le groupe des connexions de phase de coordination. Un `Terminal` peut donc être utilisé pour spécifier une ou plusieurs connexions de phase de coordination en utilisant l'attribut `Terminal.phases` qui est un `PhaseCode` de type énumération. `PhaseCode` a pour combinaisons de phase possibles ABC, ABCN, AB, BC, CA, A, B, C ainsi de suite. La spécification des phases à chaque borne permet de modéliser des cavaliers ou des commutateurs qui connectent différentes phases. Cette spécification de la connectivité de phase est entendue comme spécification de phase nominale ou normale. Le résultat de la source de mise sous tension n'est pas reflété dans cette spécification. Cette spécification de phase fournit les moyens de connecter localement des phases semblables.

La Figure 17 montre un exemple pour spécifier des connexions de phase. Cet exemple montre une instance `ConnectivityNode` sur la gauche, les phases A, B et C étant concernées. Noter que `ConnectivityNode` ne spécifie pas le contenu de sa phase mais connecte plutôt implicitement des phases semblables de bornes associées au `ConnectivityNode`. L'exemple montre un cavalier biphasé qui connecte sur la gauche (`Terminal.sequenceNumber=1`) les phases A et C et sur la droite (`Terminal.sequenceNumber=2`) les phases A et B, avec les parties intérieures des objets `SwitchPhase` du modèle de cavalier qui spécifient en détail le

chemin des connexions de phase. L'attribut `Terminal.phases` spécifie uniquement la connexion externe d'un appareil. Plus loin sur la droite, deux tronçons de ligne sont raccordés. Le segment supérieur de la ligne est une ligne biphasée sur les phases A et B. Le segment inférieur de ligne est monophasé de phase A. Les lignes foncées indiquent les associations du CIM tandis que les flèches plus claires indiquent les connexions électriques représentées au niveau de la phase.



Légende

Anglais	Français
Legend	Légende
Electrical phase connection represented	Connexion de phase électrique représentée
CIM association	Association CIM
Implicit "phase connection point" (not a CIM class)	"Point de connexion de phase" implicite (pas une classe CI)
ConnectivityNode	ConnectivityNode

Figure 17 – Connectivité de phase

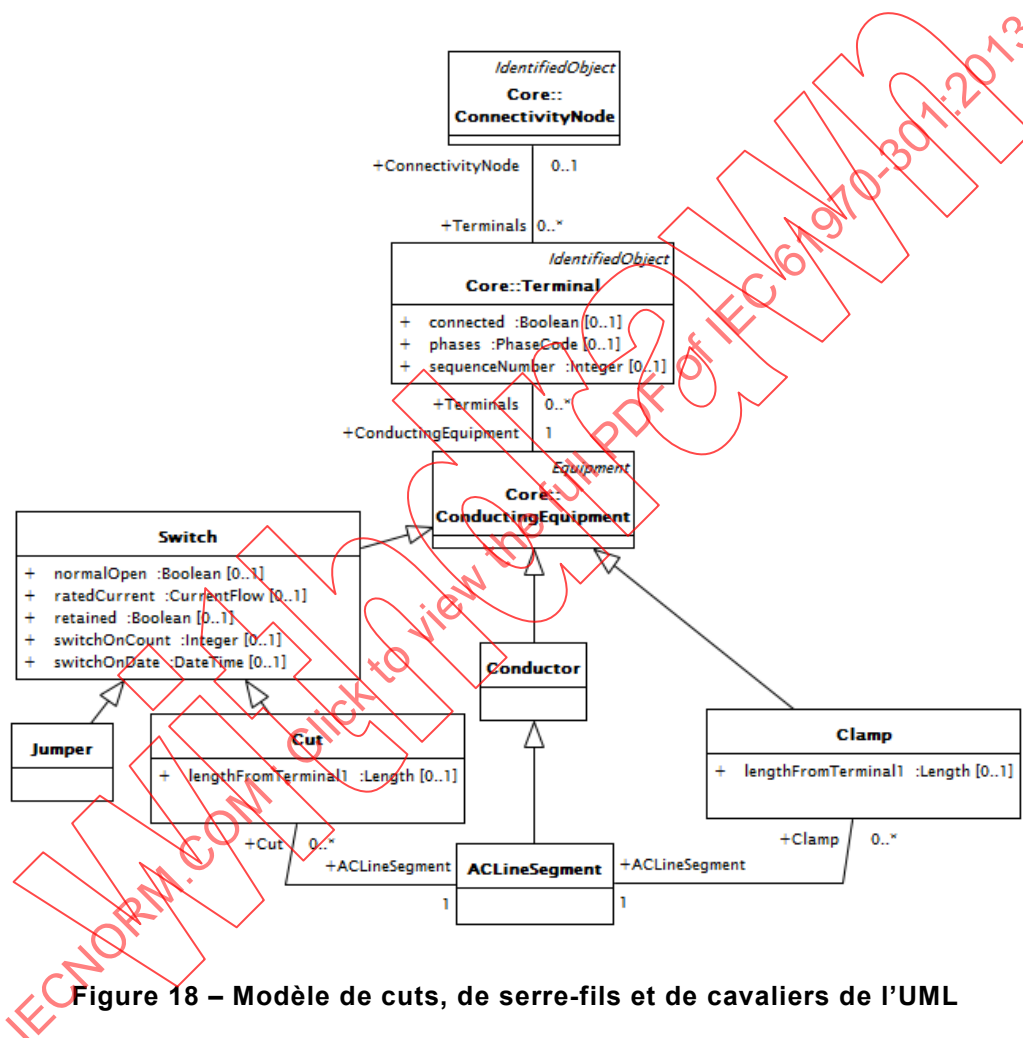
A titre d'explication, l'attribut `SwitchPhase` appelé `phaseSide1` spécifie une phase unique (comme A, B, C ou N) et se réfère au Terminal dont le `sequenceNumber` est égal à 1. L'attribut `SwitchPhase` appelé `phaseSide2` se réfère au Terminal dont le `sequenceNumber` est égal à 2.

Le modèle de commutateur permet de modéliser les connexions entre phases de façon générale à l'intérieur de l'appareil. Le modèle de transformateur comporte aussi une modélisation de la connectivité de phase interne de l'appareil. Actuellement, toutes les autres spécialisations de branche `ConductingEquipment` comme `ACLineSegment` supposent que des phases identiques sont connectées aux deux bornes. Le modèle du CIM est ouvert aux extensions futures pour introduire la connectivité de phase interne des appareils.

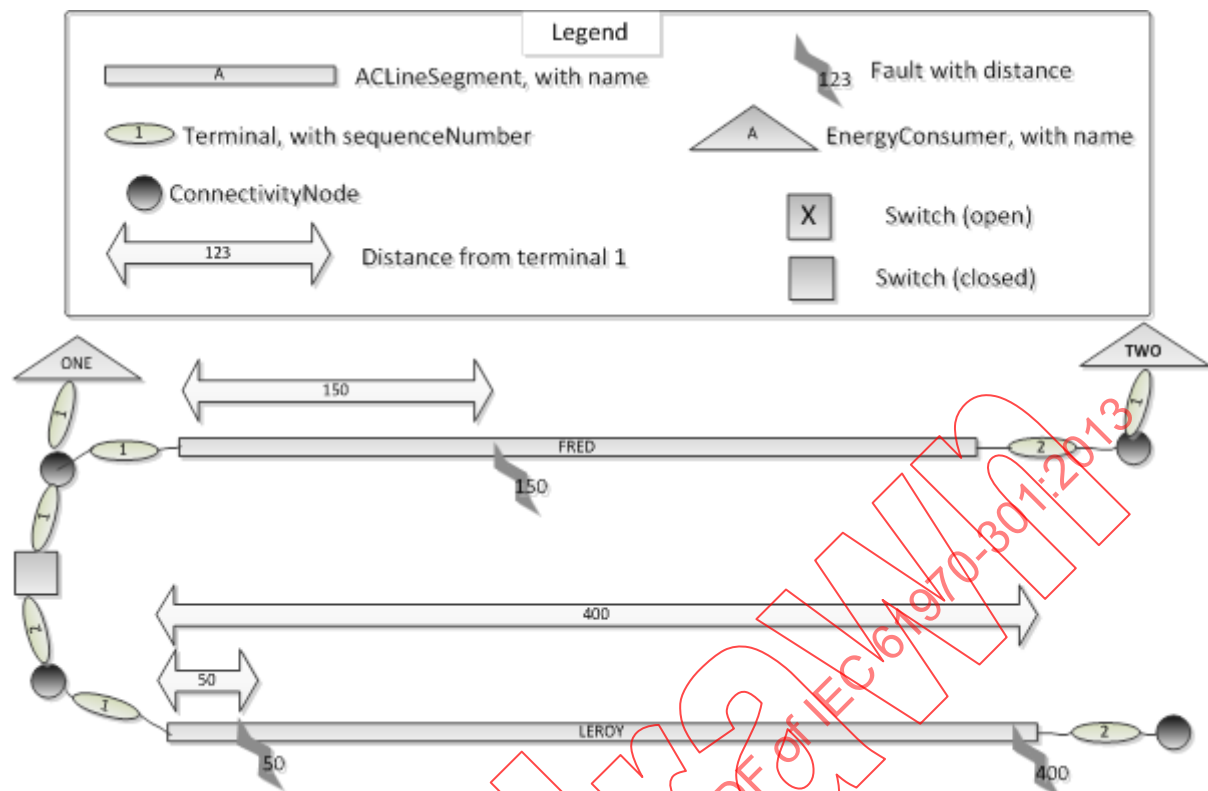
4.4.9 Modèle de cuts, de serre-fils (clamps) et de cavaliers (jumpers)

Le modèle de l'UML pour les cuts et les cavaliers sur un segment de ligne et les connexions supplémentaires sont décrites à la Figure 18. La classe `Jumper` est une spécialisation de la classe plus générale `Switch` et permet des changements de phase ainsi qu'une connectivité de phase comme le montre la Figure 17. Les cavaliers peuvent donc connecter les bornes de façon normale. Les classes `Cut` et `Clamp` du CIM fournissent des moyens d'ajouter

effectivement des Terminals (bornes) à un ACLineSegment à une distance spécifiée sur la ligne en aval, en associant de nouveaux objets du CIM à un objet ACLineSegment sans autre modification du modèle ACLineSegment. Chaque instance Clamp introduit une nouvelle borne connectée à la distance spécifiée en aval sur la ligne sans interrompre la ligne. Chaque instance Cut introduit deux nouvelles bornes qui sont effectivement les extrémités du cut. Par convention, chaque borne d'un cut est orientée vers la borne ACLineSegment dont le sequenceNumber est identique. Si le cut est ouvert, il n'y a pas de flux qui traverse le ACLineSegment au point du cut mais le flux peut passer par l'un ou l'autre côté du ACLineSegment et par la borne de cut appropriée. Habituellement, les cavaliers sont connectés aux extrémités des cuts, mais ils peuvent aussi être connectés à un type quelconque d'équipement conducteur de façon normale en utilisant ConnectivityNode ou TopologicalNode.



La Figure 19 et la Figure 20 montrent un exemple d'ajout de cuts et de serres-fils à un modèle existant de CIM. La Figure 19 montre deux tronçons de ligne physiques parallèles appelés "FRED" et "LEROY". Dans cet exemple, le système est mis sous tension à partir de la droite, le commutateur sur la gauche est fermé et les charges appelées "ONE" et "TWO" sont appliquées. Les trois défauts sont ensuite appliqués aux distances spécifiées par rapport au Terminal 1 des tronçons respectifs de ligne. Dans cet exemple, le détail de phase n'est pas spécifié pour des raisons de simplicité mais le détail de phase peut suivre la même approche présentée à 4.4.8.

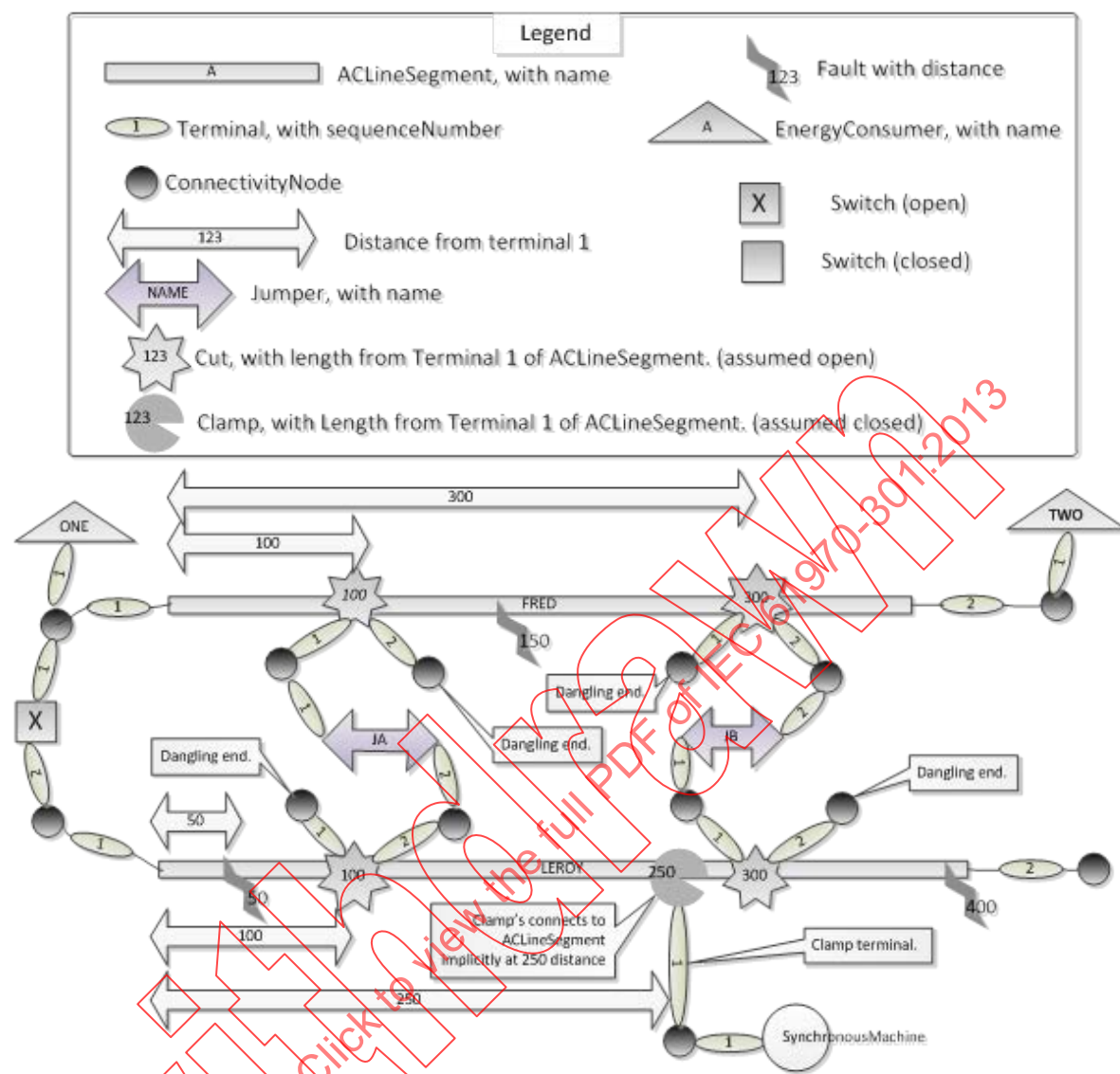


Légende

Anglais	Français
Legend	Légende
ACLineSegment, with name	ACLineSegment, avec nom
Terminal, with sequenceNumber	Terminal, avec sequenceNumber
ConnectivityNode	ConnectivityNode
Distance from terminal 1	Distance du terminal 1
Fault with distance	Erreur avec distance
EnergyConsumer, with name	EnergyConsumer, avec nom
Switch (open)	Commutateur (ouvert)
Switch (closed)	Commutateur (fermé)

Figure 19 – Exemple avant d'appliquer des cuts et des cavaliers

La Figure 20 montre comment le simple ajout de Cut, ConnectivityNode, Jumper, Clamp et éventuellement d'un autre Conducting Equipment peut représenter des changements de modèle qui reflètent un travail temporaire typique. Les mêmes modèles peuvent donc être utilisés pour des changements permanents de modèle. L'exemple montre comment des cuts appliqués aux distances spécifiées peuvent isoler les défauts en laissant pendre les extrémités défectueuses des cuts tandis que le réseau sans défaut est connecté pour fournir les charges. Cet exemple montre aussi comment plusieurs cuts peuvent être appliqués à un segment de ligne. Cet exemple montre des cuts ouverts, mais si un cut était fermé, le flux pourrait continuer à travers ACLineSegment, avec une partie du flux potentiellement redirigée vers les bornes du cut, à peu près comme dans le cas d'un serre-fil.



Légende

Anglais	Français
Legend	Légende
ACLLineSegment, with name	ACLLineSegment, avec nom
Terminal, with sequenceNumber	Terminal, avec sequenceNumber
ConnectivityNode	ConnectivityNode
Distance from terminal 1	Distance du terminal 1
Jumper, with name	Cavalier, avec nom
Cut, with length from Terminal 1 of ACLineSegment. (assumed open)	Coupure, avec longueur du Terminal 1 de ACLineSegment. (supposé ouvert)
Clamp, with length from Terminal 1 of ACLineSegment. (assumed closed)	Collier, avec longueur du Terminal 1 de ACLineSegment. (supposé fermé)
Fault with distance	Erreur avec distance
EnergyConsumer, with name	EnergyConsumer, avec nom
Switch (open)	Commutateur (ouvert)
Switch (closed)	Commutateur (fermé)
Dangling end	Bout qui pend
Clamp connects to ACLineSegment implicitly at 250 distance	Le collier se connecte à ACLineSegment de façon implicite à une distance de 250
Clamp terminal	Terminal du collier

Figure 20 – Exemple après que des cuts et des cavaliers ont été appliqués

Un cut fournit deux bornes et bien que l'exemple ci-dessus ne le représente pas, des connexions sont possibles à l'une des bornes du cut ou aux deux. Un cas d'exemple pourrait impliquer d'introduire un cut et de connecter les deux bornes de cut à l'aide d'un cavalier qui croise les phases.

Un cavalier peut être utilisé sans serres-fils ni cuts dans une situation où il faut connecter des bornes existantes. La Figure 21 représente un exemple où un cavalier est utilisé sans cut ni serre-fil. Dans ce cas, un cavalier temporaire pourrait être utilisé pour contourner un commutateur qui fait l'objet d'une maintenance.

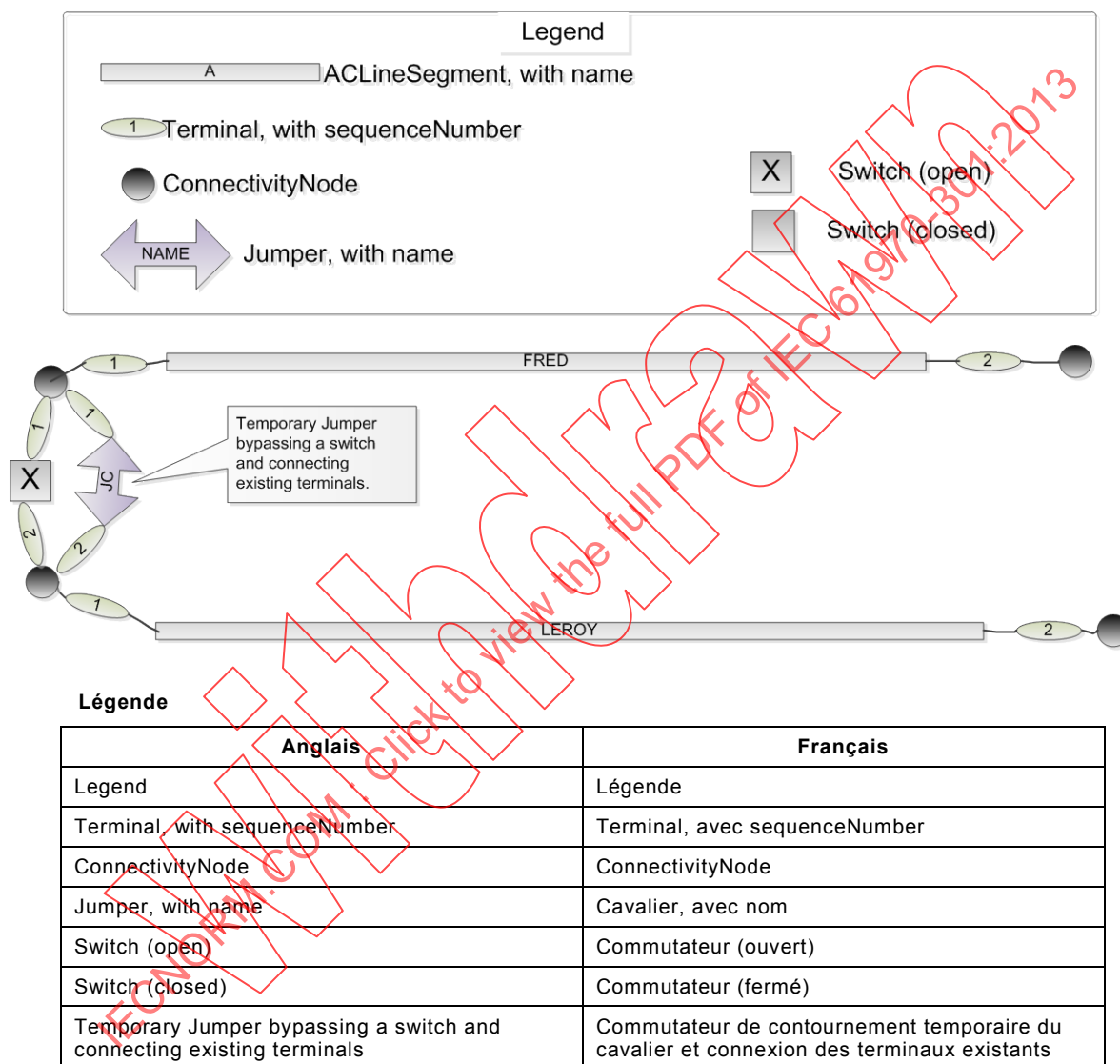


Figure 21 – Exemple de cavalier sans cut ni serre-fil

4.4.10 Mesures et commande

4.4.10.1 Vue d'ensemble des mesures

Les Measurement (mesures) sont utilisées pour représenter les variables d'état qui peuvent être rencontrées dans des processus industriels. Chaque processus industriel a ses types spécifiques de mesures. Un système de puissance a typiquement des flux de puissance, des tensions, des positions (par exemple, disjoncteurs, sectionneurs), des indicateurs de défauts (pression atmosphérique, pression d'huile, surchauffe, etc.), des compteurs (par exemple, énergie), etc.

Le nom "Measurement" semblerait indiquer que toutes les variables d'état seraient mesurées. Tel n'est pas toujours strictement le cas, car de nombreuses "Measurement" sont calculées par des fonctions SCADA ou EMS/DMS, comme les calculs d'estimateur d'état ou de répartition, par exemple. Par conséquent, un Measurement peut avoir un certain nombre de valeurs alternatives (par exemple, fournies manuellement, télémesurées, estimées par état, optimisées, etc.). Cela est pris en charge par les modèles Measurement et MeasurementValue dans le paquetage Meas. Les classes dans le paquetage StateVariable ont été ajoutées au modèle CIM, et ce, spécifiquement pour le besoin de prendre en charge l'échange de valeurs calculées par des fonctions telles que l'estimateur d'état et le calcul de répartition.

Le modèle de mesure permet désormais de spécifier facultativement les phases spécifiques d'une mesure.

4.4.10.2 Vue d'ensemble des commandes

Les commandes sont utilisées pour représenter les variables de commande. Les variables de commande du système d'alimentation électrique sont typiquement des points de consigne, des échelons de commande, des commandes Sélectionner avant d'exécuter et des commandes marche/arrêt. Le paquetage Meas prend en charge les variables de commande avec le modèle Control.

4.4.10.3 Utilisation des classes liées à la classe Measurement

Une PowerSystemResource (PSR) peut avoir zéro à plusieurs mesures qui lui sont associées en contenant une ou plusieurs mesures. Chaque Measurement peut contenir une ou plusieurs valeurs de mesure. Le respect des conseils suivants permettra aux applications de naviguer et de retrouver les valeurs de mesure requises de manière cohérente (voir Figure 22):

- a) Les Measurement d'une PowerSystemResource sont classées par l'attribut measurementType (type de mesure). Les valeurs devant être utilisées pour Measurement.measurementType sont spécifiées dans le Tableau 2.
- b) Les MeasurementValues (valeurs de mesure) d'une Measurement sont classées par MeasurementValueSource (c'est-à-dire source de valeur de mesure).
- c) MeasurementValueSource hérite de IdentifiedObject. Les valeurs devant être utilisées pour le nom MeasurementValueSource sont spécifiées dans le Tableau 3. Ce tableau fournit un certain nombre de noms de source à utiliser où cela est possible. Cependant, les noms exacts devant être utilisés pour des applications spécifiques sont définis dans le document correspondant CEI 61970 CIS.
- d) Les tableaux peuvent être étendus pour des besoins propriétaires. Les noms ajoutés doivent commencer par un nom unique (par exemple, le nom de la compagnie) et un caractère "souligné" ("underscore"). Exemple: xyz_AverageTemperature.
- e) La valeur ValueAliasSet est utilisée pour des mesures discrètes et décrit des correspondances de valeurs vers des noms symboliques, comme des libellés d'énumération qui ont des valeurs entières spécifiques. Des protocoles de communication différents (par exemple pour les RTU (c'est-à-dire Remote Units, à savoir unités distantes) et pour les centres de conduite tels que ICCP (c'est-à-dire Intelligent Customer Contact Platform, à savoir plate-forme intelligente de relations avec l'abonné) ou ELCOM) utilisent des codages de données différents. Un système peut avoir une mise en correspondance à l'échelle du système pour toutes les valeurs Discrete (c'est-à-dire discrètes) ou regrouper les valeurs Discrete et effectuer la mise en correspondance par groupe. La création d'une seule mise en correspondance à l'échelle du système qui couvrirait des protocoles de communication existants ne relève pas du domaine d'application de la présente spécification.

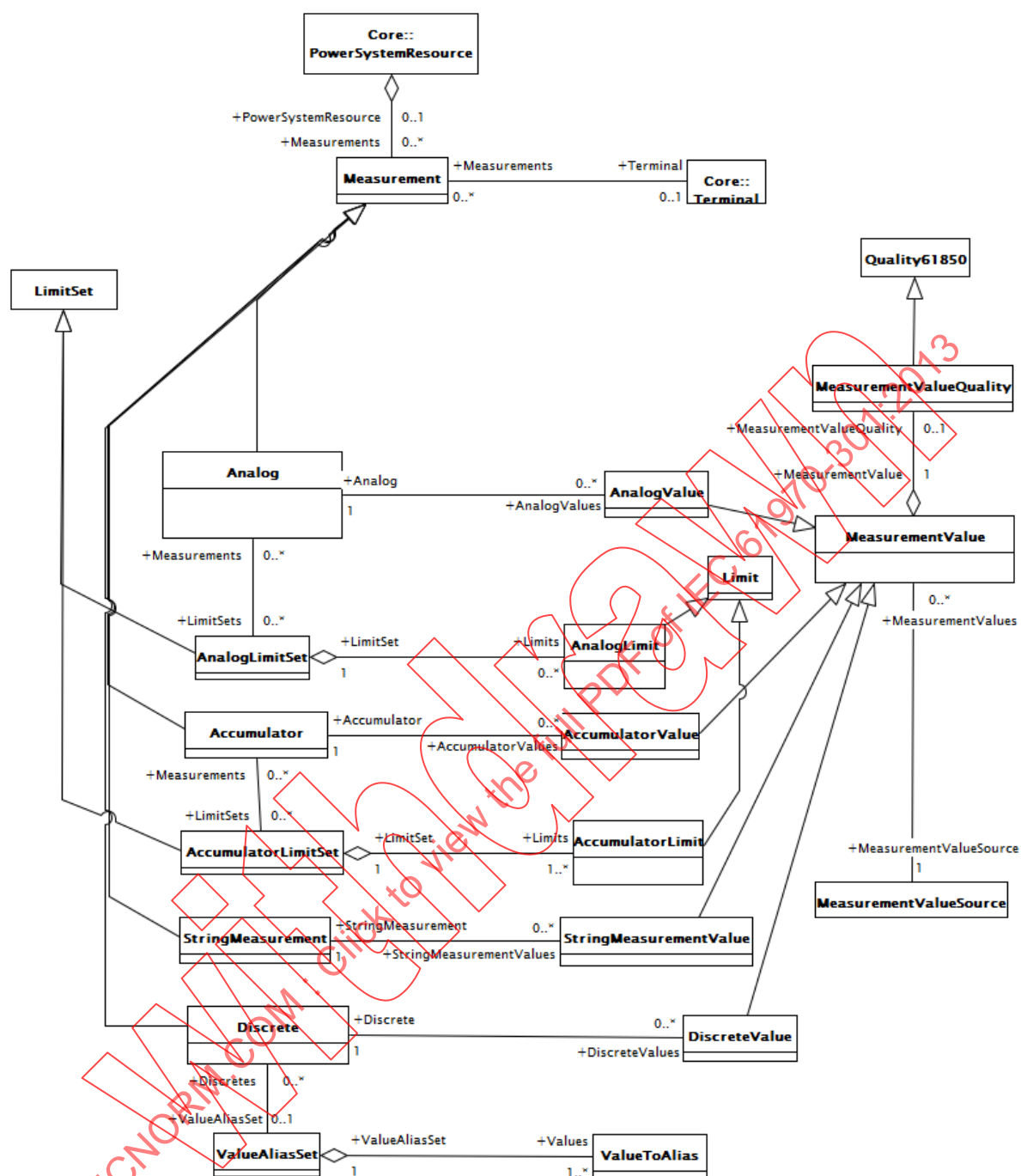


Figure 22 – Navigation de PSR à MeasurementValue

Tableau 2 – Conventions de dénomination de measurementType

measurementType	61850 Name	description
Courant	Amp	Courant (efficace) d'un circuit non triphasé
ThreePhaseCurrent	AvAmps	Courant total (efficace) dans un circuit triphasé
PhaseCurrent	A	Courant de phase mesuré
Frequency	Hz	Fréquence
PowerFactor	PwrFact	Facteur de puissance non alloué à une phase
ThreePhasePowerFactor	TotPF	Facteur de puissance moyen dans un circuit triphasé
ThreePhaseApparentPower	TotVA	Puissance apparente totale dans un circuit triphasé
ThreePhaseReactivePower	TotVAr	Puissance réactive totale dans un circuit triphasé
ThreePhaseActivePower	TotW	Puissance réelle totale dans un circuit triphasé
ApparentPower	VoltAmp	Puissance apparente dans un circuit non triphasé
ReactivePower	VoltAmpr	Puissance réactive dans un circuit non triphasé
Voltage	Vol	Tension (efficace) non allouée à une phase
ActivePower	Watt	Puissance réelle dans un circuit non triphasé
Pressure	Pres	Pression
Temperature	Tmp	Température
Angle	Ang	Déphasage entre la tension et le courant
ApparentEnergy	TotVAh	Energie apparente
ReactiveEnergy	TotVArh	Energie réactive
ActiveEnergy	TotWh	Energie réelle
Automatic	Auto	Fonctionnement automatique (non manuel)
LocalOperation	Loc	Fonctionnement local (non télécommandé)
SwitchPosition	Pos	Position de commutation [2bits= intermédiaire,ouvert,fermé,ignorer]
TapPosition	TapPos	Position de prise du transformateur de puissance ou du déphaseur
Operation Count	OperCnt	Compteur d'opérations – typiquement pour les interrupteurs
LineToNeutralVoltage		Tension phase à neutre
LineToGroundVoltage		Tension phase à terre

Le Tableau 2 décrit divers types de mesures qui sont aussi définis dans la CEI 61850. Les colonnes du Tableau 2 ont la signification suivante:

- measurementType (Measurement.measurementType) est le nom du type de mesure selon la CEI 61970.
- 61850 Name est le nom attribué à l'objet de données dans la CEI 61850. (Se reporter à l'Article 6 de la CEI 61850-7-4:2010, sémantique du nom d'objet de données).
- description de la donnée.

Il doit être noté que le Tableau 2 n'est pas une liste exhaustive et que la correspondance entre mesures telles que définies dans un centre de conduite et un poste n'est pas triviale.

**Tableau 3 – Conventions de dénomination de MeasurementValueSource
(source de valeur de mesure)**

nom	description
SCADA	Valeurs télémesurées reçues d'un système SCADA local
CCLink	Valeur reçue d'un centre de conduite distant par l'intermédiaire de TASE.2 ou autre protocole
Operator	Valeur entrée par un opérateur (toujours maintenu manuellement, le PSR n'est pas connecté à une "RTU" (unité terminale distante))
Estimated	Valeur mise à jour par un estimateur d'état
PowerFlow	Valeur mise à jour par un Powerflow (calcul de répartition)
Forecasted	Valeur qui est planifiée ou prévisionnelle.
Calculated	Valeur calculée à partir d'autres valeurs de mesure (par exemple, une somme)
Allocated	Valeur calculée par un allocateur de charge

En respectant ces conventions:

- chaque instance de Measurement représente une grandeur technique d'une PowerSystemResource;
- chaque MeasurementValue d'un Measurement représente une valeur pour la grandeur technique, fournie par une source unique;
- l'attribut source de MeasurementValueQuality indique alors si la source a vraiment fourni la valeur courante ou bien si elle est une valeur de substitution ou une valeur par défaut.

Noter qu'une nouvelle identité MeasurementValue n'est pas normalement créée pour chaque échange d'une valeur mesurée. Il est prévu que la même instance MeasurementValue puisse échanger de nouvelles valeurs dans des messages ultérieurs. C'est pourquoi l'identité d'une instance MeasurementValue peut être établie a priori pour échanger des valeurs mesurées. La modélisation d'une série chronologique de valeurs n'est pas explicitement mentionnée dans le modèle du CIM bien que le modèle permette aux systèmes de constituer en interne de tels modèles de série chronologique à l'aide d'une série d'échanges de valeurs de mesure.

4.4.10.4 Rattachement des Measurement

Comme mentionné dans 4.4.10.3 et comme montré à la Figure 22, les Measurement sont contenues par une PowerSystemResource. Cela est suffisant pour des Measurement qui ne sont pas reliées à la connectivité, par exemple la température, le poids, la taille.

Pour spécifier l'emplacement d'une Measurement dans le réseau, il est utilisé une association à Terminal. Des exemples sont les flux de puissance, les tensions et les courants. Les tensions (Voltage) n'ont pas de sens et peuvent être attachées en fonction de l'emplacement du capteur. Les flux ont un sens et doivent être attachés de façon à ce que le sens du flux soit évident d'après le placement.

La Figure 23 montre deux exemples de placement de Measurements.

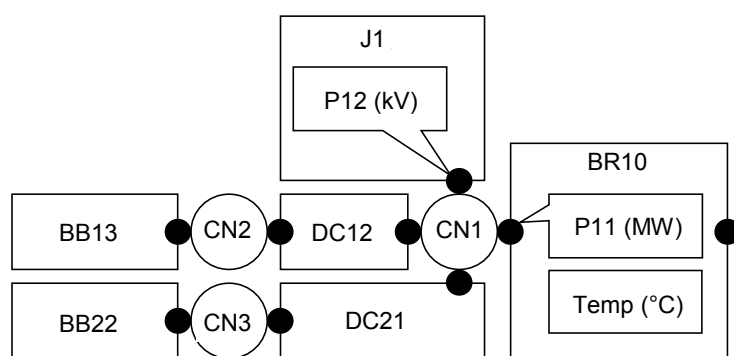


Figure 23 – Placement de Measurement

P12 est une Measurement de tension qui mesure la tension à la jonction J1. P12 est topologiquement relié au ConnectivityNode CN1 via la Terminal de la jonction J1. P11 est une Measurement qui mesure le flux traversant le disjoncteur BR10 du côté connecté au ConnectivityNode CN1. P11 est topologiquement relié au ConnectivityNode CN1 via la Terminal gauche du disjoncteur BR10. Temp est une Measurement qui mesure la température du disjoncteur. Comme la température n'est pas reliée à la connectivité, elle n'a pas de relation avec une Terminal – elle appartient seulement au disjoncteur BR10.

4.4.10.5 Mesures ICCP

Les sources de données ICCP (officiellement connues comme CEI 60870-6 TASE.2) sont spécifiées en utilisant les classes MeasurementValue (et leurs spécialisations) et la classe MeasurementValueSource. La classe MeasurementValueSource est utilisée pour définir le centre de conduite qui fournit les données ICCP. La MeasurementValueSource doit être associée à une instance du Name où l'attribut Name.name contient le nom du centre de conduite fournisseur. L'instance du NameType associée au Name du centre de conduite doit avoir "ICCP" pour attribut NameType.name.

Les classes MeasurementValue sont utilisées pour spécifier l'ICCP ID. La MeasurementValue doit être associée à une instance de Name où l'attribut Name.name contient le ICCP ID. L'instance du NameType associé à l'ICCP ID Name doit avoir "ICCP" pour attribut NameType.name. L'attribut MeasurementValue.name contient le nom de point SCADA. Chaque MeasurementValue sera associée à un Measurement. Chaque MeasurementValue qui est fournie à l'aide d'ICCP doit aussi avoir une association à une MeasurementValueSource.

4.4.11 Modèle de commande de régulation

Une commande de régulation, par exemple une commande automatique de tension sur des générateurs ou une commande de prise de tension sur des transformateurs, est modélisée en utilisant la classe RegulatingControl qui offre la possibilité de modéliser plusieurs instances d'équipement qui participent à un plan de régulation. Ces plans de régulation peuvent être physiques ou mis en œuvre manuellement dans le fonctionnement réel du système d'énergie, mais ils sont reflétés dans les modèles utilisés à des fins d'analyse des systèmes de puissance. La Figure 24 montre comment les classes RegulatingCondEq et TapChanger peuvent participer à la régulation.

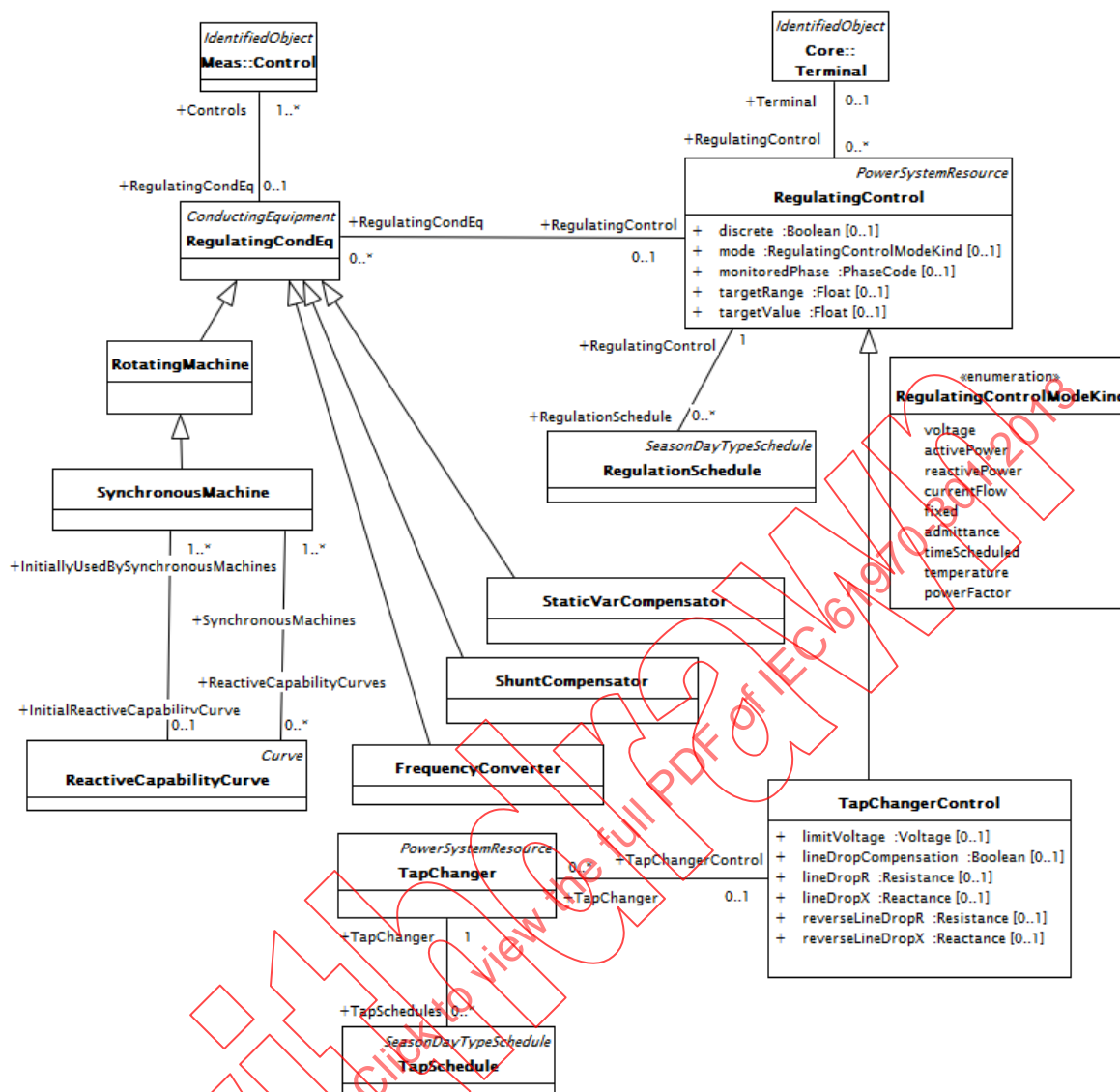


Figure 24 – Modèles de commande de régulation

4.5 Conseils de modélisation

4.5.1 Modélisation pour modifications

Les Paragraphes 4.5.2 à 4.5.5 proposent des conseils sur la façon de maintenir et étendre le CIM.

Le CIM est destiné à contenir les classes et les attributs qui seront échangés sur des interfaces publiques entre des applications principales. L'objectif est de ne conserver, autant que possible, que les caractéristiques génériques à partir desquelles une mise en œuvre détaillée peut être extraite. Il est, en général, plus facile de modifier la valeur ou le domaine d'un attribut que de modifier une définition de classe. Cela rend le modèle plus solide, car il peut supporter une plus large classe d'exigences, et plus stable, car il peut être possible de gérer de nouvelles exigences sans avoir à modifier le modèle.

4.5.2 Processus pour les amendements au CIM

Il peut être souhaitable d'amender le CIM pour réviser le modèle ou pour étendre le CIM afin de modéliser des éléments supplémentaires d'un système électrique. Le processus recommandé pour ces amendements est le suivant.

- a) Préparer un ou plusieurs cas d'utilisation et une demande de changement qui décrivent les changements souhaités. Il convient que cela inclue les changements proposés aux diagrammes de classes concernés avec illustration des classes, attributs et associations nouveaux ou révisés.
- b) Le ou les cas d'utilisation et la demande de changement sont ensuite examinés par le Groupe de Travail approprié de la CEI, qui va décider s'il convient que les changements demandés soient traités en tant que révisions de la norme CIM courante, ou s'il convient qu'ils soient traités comme des amendements privés sans nécessiter de changement pour la norme.
- c) Les propositions d'amendements acceptés par le groupe de travail seront ajoutées à la liste des sujets majeurs et le moment venu, une nouvelle version du modèle CIM sera préparée et une mise à jour de la spécification CIM appropriée de la CEI sera faite.

4.5.3 Modifications apportées au modèle UML du CIM

Du point de vue de la modélisation, lorsque le CIM est étendu, l'approche consiste à partir du modèle UML de CIM existant. Les extensions peuvent être ajoutées de plusieurs façons disponibles avec UML, mais dans tous les cas, la bonne approche est d'examiner le modèle courant et de déterminer la meilleure extension à partir des diagrammes de classes existants. Les extensions peuvent prendre une des formes suivantes, de la plus simple à la plus complexe:

- ajouter des attributs supplémentaires à des classes préexistantes;
- ajouter de nouvelles classes qui sont des spécialisations de classes préexistantes;
- ajouter de nouvelles classes à l'aide d'associations avec des classes préexistantes.

Dans le but d'étendre cette norme, le principal objectif est de réutiliser le CIM existant dans la plus grande mesure possible. Du point de vue des paquetages, il convient que les extensions soient faites aux paquetages existants chaque fois que c'est possible. Si les extensions comprennent un nouveau domaine d'application, il convient d'envisager de créer un nouveau paquetage pour les additions, mais en créant toujours les nécessaires associations au paquetage préexistant, en gardant à l'esprit que bien qu'un nouveau paquetage soit créé, le CIM continue d'être un seul modèle.

Quand il s'agit d'apporter des changements non normatifs au modèle d'information, il est recommandé d'étendre le CIM normalisé sans modifier le CIM normalisé. Il est recommandé de placer de préférence ces extensions dans de nouveaux paquetages, à la fois pour identifier que les changements ne sont pas normatifs et pour faciliter le transfert des changements vers une version de la norme mise à jour.

4.5.4 Modifications apportées aux documents normatifs du CIM

D'un point de vue documentaire, lorsque le CIM est étendu, une décision doit être prise quant à la question de savoir si les changements constituent des mises à jour de documents normatifs sur le CIM ou si une nouvelle CEI 61970-3x spécifique est nécessaire. Dans les deux cas, les extensions deviendront une partie de la norme CEI sur le CIM.

4.5.5 Eléments déconseillés

Il se peut que le modèle du CIM contienne à un moment donné des paquetages, des classes, des attributs ou des associations qui sont considérés comme déconseillés. Ces éléments seront notés "déconseillés" dans la documentation ou porteront un stéréotype de l'UML "déconseillé". Un élément déconseillé figure dans la présente version du modèle mais il est prévu de le supprimer dans les versions futures. Il n'est pas garanti qu'un élément non déconseillé soit retenu dans les versions futures mais il est recommandé d'éviter autant que possible d'utiliser un terme marqué comme déconseillé.

4.5.6 Profils CIM

Une mise en œuvre du CIM peut ne pas inclure toutes les classes, attributs et associations de la norme CIM et être tout de même conforme à la norme CIM. Des profils peuvent être définis pour spécifier quels éléments doivent être inclus (éléments obligatoires) pour une utilisation particulière du CIM, ainsi que ceux qui sont facultatifs. Ces profils sont définis dans les documents concernant les profils de la série de normes CEI 61970 ainsi que dans d'autres séries de normes du TC57 qui utilisent et étendent le CIM de base spécifié dans ce document.

Un exemple est le profil pour échanger des modèles de système de puissance. Ce profil spécifie comment le CIM doit être utilisé pour échanger des modèles de système de puissance en XML et il spécifie aussi les classes, attributs et associations, obligatoires et facultatifs, devant être pris en charge pour cette utilisation du CIM. Les profils peuvent être maintenus à l'aide de nombreux outils tels que l'outil open source (c'est-à-dire de source ouverte) CIMTool, disponible à l'adresse <http://www.CIMTool.org>.

4.6 Outils de modélisation

Cette version du CIM a été établie en utilisant le produit Sparx Systems Enterprise Architect. Le modèle UML du CIM complet existe sous la forme d'un fichier de projet Enterprise Architect¹ et peut être visualisé avec cet outil, y compris les diagrammes de classe et les descriptions des classes, des attributs, des types et des relations. La visualisation du CIM de cette façon offre une interface de navigation graphique qui permet de visualiser toutes les données de spécification du CIM par simple pointer-cliquer sur le diagramme de classe dans chaque paquetage. Une visionneuse gratuite est disponible auprès de Sparx Systems.

Idéalement, le modèle d'informations du CIM est indépendant de tout outil UML spécifique, bien que l'expérience ait montré que les échanges entre des outils différents soient souvent loin d'être parfaits. En attendant la réalité de l'interopérabilité des outils, les futures modifications apportées à la spécification du CIM, débouchant sur de nouvelles versions de la présente norme, seront incorporées en premier lieu dans la description du projet Enterprise Architect pour garantir une source unique pour les données de modèle CIM.

L'Article 6 de la présente norme a été généré automatiquement en utilisant l'outil libre, disponible à l'adresse <http://cimug.ucaiug.org/MTEI/Shared%20Documents/jCleanCim>, qui utilise les interfaces d'automatisation de Microsoft Word^{TM2}, d'Enterprise Architect ainsi que le format et les styles normalisés de la CEI.

4.7 Conventions de mise en œuvre pour les utilisateurs

4.7.1 Conventions au-delà de l'UML

Le Paragraphe 4.7.2 propose des conventions d'utilisateur recommandées pour l'utilisation du CIM dans la mise en œuvre des systèmes réels.

4.7.2 Nombre des Terminal pour les objets ConductingEquipment

Les ConductingEquipments (équipements conducteurs) suivants ont deux bornes: ACLineSegment, DCLineSegment, Switch et toutes ses spécialisations (y compris Jumper,

¹ Enterprise Architect est l'appellation commerciale d'un produit distribué par Sparx Systems. Cette information est donnée à l'intention des utilisateurs de la présente Norme internationale et ne signifie nullement que la CEI approuve ou recommande l'emploi exclusif du produit ainsi désigné. Des produits équivalents peuvent être utilisés s'il est démontré qu'ils conduisent aux mêmes résultats.

² WordTM est l'appellation commerciale d'un produit distribué par Microsoft. Cette information est donnée à l'intention des utilisateurs de la présente Norme internationale et ne signifie nullement que la CEI approuve ou recommande l'emploi exclusif du produit ainsi désigné. Des produits équivalents peuvent être utilisés s'il est démontré qu'ils conduisent aux mêmes résultats.

Fuse, Breaker, Disconnecter, LoadBreakSwitch et Cut), SeriesCompensator et EquivalentBranch. La classe PowerTransformer a typiquement deux bornes mais elle peut aussi avoir une borne ou davantage. Par exemple un transformateur mis à la terre connecté en zig-zag peut avoir une borne. On utilise souvent des transformateurs à trois bornes dans des systèmes de transport et les transformateurs pour cas spéciaux peuvent comporter quatre, cinq bornes ou davantage. Toutes les autres classes feuilles ConductingEquipment (notamment qui comprennent Clamp et BusbarSection) ont une seule borne.

4.8 Exemples de modélisation CIM

Des modèles de systèmes électriques ont été créés de plusieurs façons à partir du modèle UML du CIM. Le premier exemple est une version du schéma RDF (Resource Description Framework, c'est-à-dire cadre général pour la description des ressources) du CIM, qui utilise XML (eXtensible Markup Language) pour décrire un modèle de réseau de système de puissance. La CEI 61970-501 et d'autres normes de profil en développement sont utilisées pour spécifier le format d'échange de modèles. Des versions de schémas RDF du CIM ont été utilisées pour créer les fichiers de modèles XML de réseaux réels pour des tests d'interopérabilité. Une version de schéma RDF du CIM est générée à partir du fichier de modèle UML du CIM en utilisant des outils logiciels basés sur une spécification de schéma RDF du CIM.

Il convient de noter qu'une version de schéma RDF du CIM est du niveau métadonnées plutôt qu'une instanciation d'un réseau réel. Cependant, des fichiers de modélisation complets de réseaux avec tous les éléments du réseau ainsi que leurs connexions électriques peuvent être générés par des fournisseurs utilisant des outils d'exportation propriétaires, puis importés ensuite par d'autres systèmes avec un outil similaire qui est utilisé pour remplir une base de données technique locale sur le réseau. Des exemples de ces modèles XML du CIM sont notamment les fichiers Siemens 100 bus model, Alstom 60 bus model et ABB 40 bus model³⁾ utilisés pour des essais d'interopérabilité XML du CIM.

5 Modèle détaillé

5.1 Vue d'ensemble

La partie du Modèle d'Information Commun (CIM) dont traite la présente norme représente une vue logique complète des informations échangées parmi différents systèmes dans les services publics d'électricité. Cette définition inclut les classes et les attributs publics ainsi que leurs relations. Les paragraphes suivants décrivent la manière dont l'Article 6 est structuré. L'Article 6 est généré automatiquement à partir du modèle CIM maintenu avec les outils décrits en 4.6.

5.2 Contexte

Le CIM est fractionné en sous-paquetages. Les classes au sein des paquetages sont répertoriées dans l'ordre dans lequel elles sont définies dans le modèle UML. Les attributs natifs de la classe sont d'abord répertoriés, suivis des attributs hérités dans l'ordre de la hiérarchie d'héritage. Les associations natives sont énumérées en premier pour chaque classe, suivies des associations héritées dans l'ordre de profondeur d'héritage, puis par ordre alphabétique selon le nom de l'extrémité d'association. Les associations sont décrites selon le rôle de chaque classe participant à l'association. Les extrémités d'associations sont répertoriées pour chaque classe à chaque extrémité d'une association.

La Figure 25 montre les paquetages de haut niveau inclus dans la présente norme.

³⁾ Siemens100 bus model, Alstom 60 bus model et ABB 40 bus model sont les appellations commerciales de produits distribués par Siemens, Areva et ABB. Cette information est donnée à l'intention des utilisateurs de la présente Norme internationale et ne signifie nullement que la CEI approuve ou recommande l'emploi exclusif du produit ainsi désigné. Des produits équivalents peuvent être utilisés s'il est démontré qu'ils conduisent aux mêmes résultats.

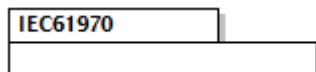


Figure 25 – Paquetages de haut niveau du CIM

Pour chaque paquetage, le modèle d'information de chaque classe est entièrement décrit. Les informations d'attribut et d'extrémité d'association pour des attributs natifs et hérités sont énumérées au Tableau 4 et au Tableau 5 en prenant une classe appelée "Class1" à titre d'exemple dans un paquetage "Package1" à titre d'exemple. Pour tous les éventuels attributs ou extrémités d'association hérités, la colonne "description" contiendra le texte indiquant que les attributs sont hérités d'une classe spécifique. Dans cet exemple, nous supposons que Class1 est hérité de la classe IdentifiedObject. La colonne "description" pour les attributs natifs et extrémités d'associations contient la description réelle. Pour des attributs hérités, la classe de base est notée dans la description et la documentation complète sur l'attribut est disponible dans cette classe de base sous forme d'attribut natif.

Tableau 4 – Attributs de Package1::Class1

nom	type	description
native1	Float	Un attribut natif virgule flottante de la classe est décrit ici.
native2	ActivePower	Documentation pour un autre attribut natif du type ActivePower.
name	String	Hérité de: IdentifiedObject

Dans les cas où un attribut est une constante, le Tableau 4 ajoute l'expression "(const)" à la colonne du nom. Dans ces cas, l'attribut a normalement une valeur initiale aussi qui est précédée du signal "égal" et accolée au nom d'attribut.

Tableau 5 – Extrémités d'association de Package1::Class1 avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] Class2	Class2	La description de l'extrémité d'association à une autre classe appelée "Class1". Noter qu'il est typique mais non nécessaire que l'extrémité d'association soit identique au type de référence. Cette extrémité d'association peut référencer zéro ou un objet. L'autre extrémité de l'association peut référencer un nombre quelconque d'instances de Class1.
[0..1]	[0..*] SecondClass2	Class2	Description de références à la classe appelée "Class2". Dans ce cas, il est possible de faire un nombre quelconque de références. La référence inverse peut référencer zéro ou une instance de Class1.
[1..1]	[0..*] Class3	Class3	Référence à un nombre quelconque d'instances de Class3. La référence inverse doit référencer une seule instance de Class1.
[0..*]	[0..*] Class4	Class4	Association plusieurs à plusieurs.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

Dans le Tableau 5, la première colonne décrit la multiplicité en cette extrémité de l'association (c'est-à-dire, comment cette classe participe à l'association). La deuxième colonne décrit l'autre extrémité d'association. Sa multiplicité est insérée entre des crochets. Le nom de l'extrémité d'association est répertorié en texte en clair. La classe à l'autre extrémité de l'association est donnée dans la troisième colonne. Une multiplicité zéro indique une association facultative. Une multiplicité "*" indique que n'importe quel nombre est autorisé. Par exemple, une multiplicité [1..*] indique qu'une plage allant de 1 à n'importe quel nombre plus grand est autorisée.

Lorsqu'une classe est une énumération appelée "Enumeration1", le tableau des attributs est remplacé par la documentation de libellés comme montre le Tableau 6, puisque les libellés d'énumération n'ont pas de type. Il n'existe pas de libellés d'énumération hérités pour une classe d'énumération.

Tableau 6 – Libellés de Package1::Enumeration1

libellé	description
native1	Il s'agit de la première valeur enumeration native.
native2	Il s'agit de la deuxième valeur enumeration native.
native3	Il n'y a typiquement pas d'attributs hérités pour les énumérations.

6 Paquetage de haut niveau (Top package) IEC61970

6.1 Généralités

Paquetage de haut niveau pour CEI 61970.

La Figure 26 montre le diagramme de classe IEC61970Dependencies.

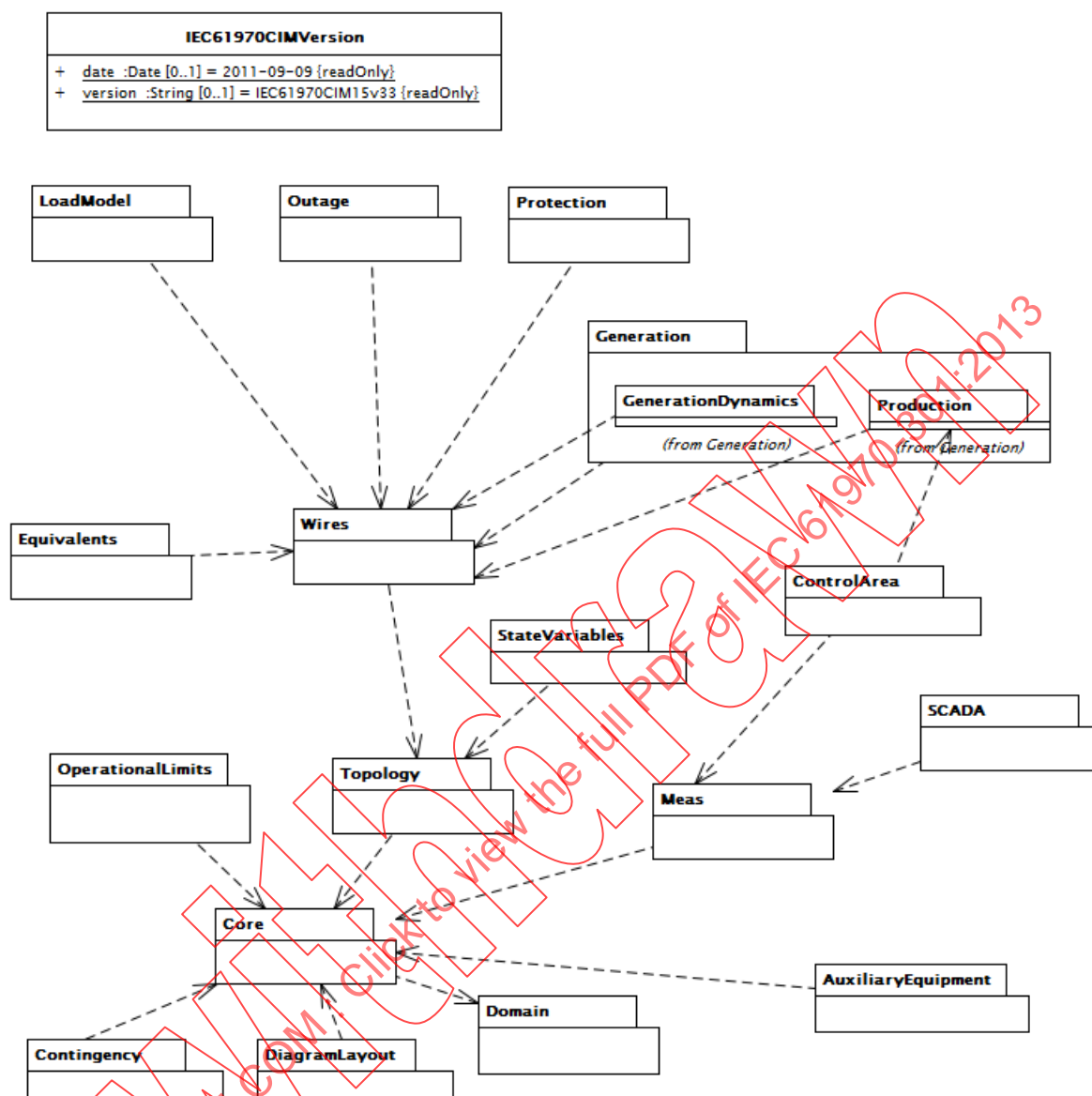


Figure 26 – Diagramme de classe IEC61970::IEC61970Dependencies

Ce diagramme montre tous les paquetages 61970 et leurs dépendances logiques.

6.2 Classe racine IEC61970CIMVersion

Il s'agit du numéro de version CIM de la CEI 61970 assigné à ce modèle UML.

Le Tableau 7 montre tous les attributs de IEC61970CIMVersion.

Tableau 7 – Attributs de IEC61970::IEC61970CIMVersion

nom	type	description
date=09/09/2011 (const)	Date	La forme est AAAA-MM-JJ par exemple le 5 janvier 2009 est 2009-01-05.
version=IEC61970CIM15v33 (const)	String	La forme est IEC61970CIMXXvYY où XX est la version principale du paquetage CIM et YY la version secondaire. Par exemple IEC61970CIM13v18.

6.3 Paquetage Domain

6.3.1 Généralités

Le paquetage Domain définit les types de données primitives qui sont utilisées par des classes dans d'autres paquetages. Des stéréotypes sont utilisés pour décrire les types de données. Sont définis les stéréotypes suivants:

<<enumeration>> liste des valeurs constantes permises.

<<Primitive>> type de données le plus simple qui est utilisé pour composer tous les autres type de données.

<<CIMDatatype>> type de données qui contient un attribut de valeur, une unité facultative de mesure et un multiplicateur d'unité. L'unité et le multiplicateur peuvent être spécifiés comme des variables statiques initialisées à la valeur autorisée.

<<Compound>> un composé de Primitive, énumération, CIMDatatype ou d'autres classes Compound, dans la mesure où les classes Compound ne sont pas inversées.

La Figure 27 montre le diagramme de classe BasicDatatypes.

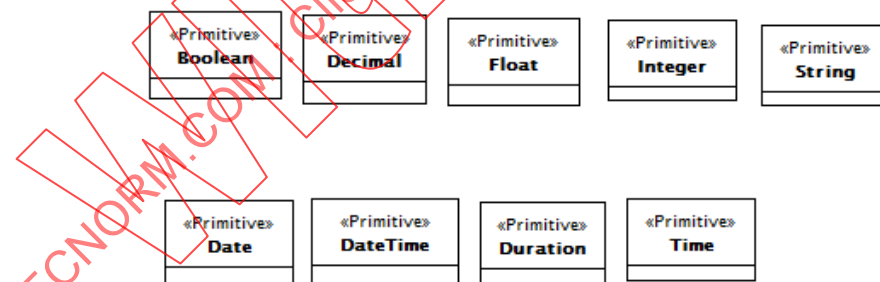


Figure 27 – Diagramme de classe Domain::BasicDatatypes

Ce diagramme montre les types de données fondamentaux qui sont la base de tous les autres types de données.

La Figure 28 montre le diagramme de classe ElectricityDatatypes.

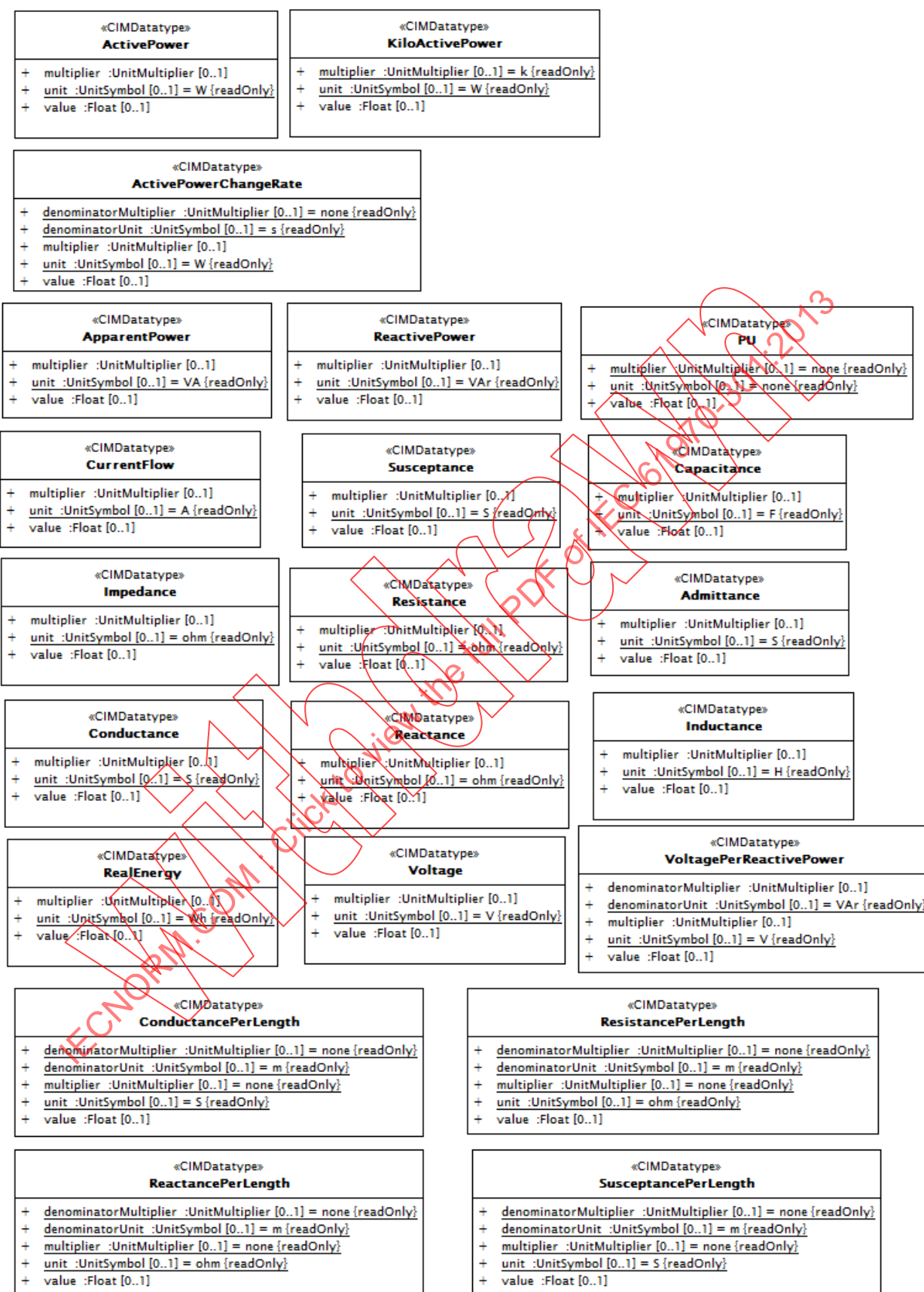


Figure 28 – Diagramme de classe Domain::ElectricityDatatypes

Ce diagramme représente les types de données liées à l'électricité.

La Figure 29 montre le diagramme de classe EnumeratedUnitDatatypes.

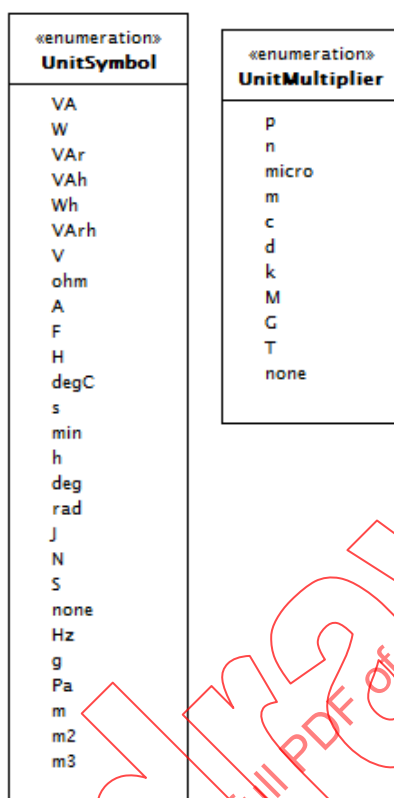


Figure 29 – Diagramme de classe Domain::EnumeratedUnitDatatype

Ce diagramme montre les types de données énumérés liés au système d'unités SI.

La Figure 30 montre le diagramme de classe GeneralDatatypes.

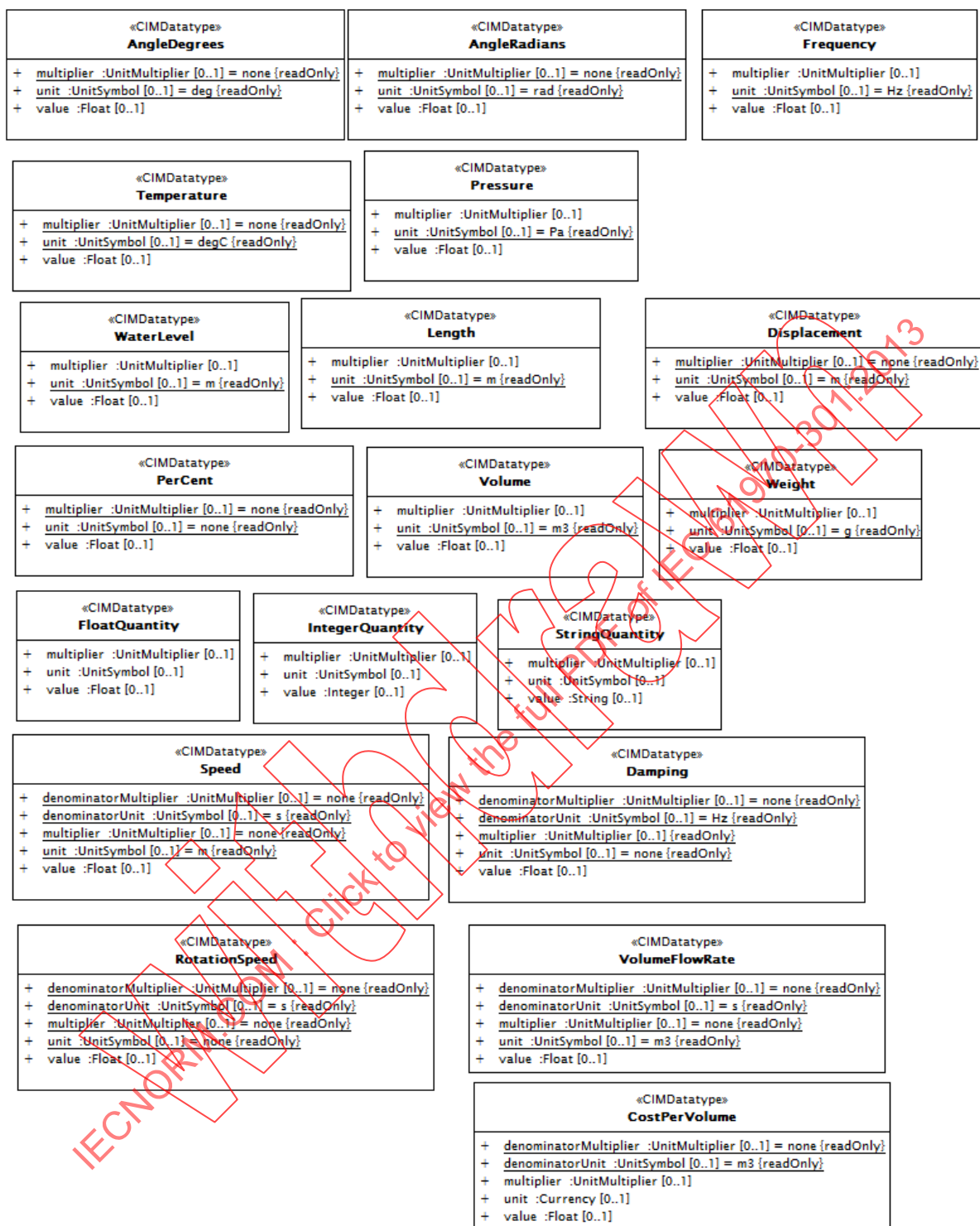


Figure 30 – Diagramme de classe Domain::GeneralDatatypes

Ce diagramme représente les types de données générales.

La Figure 31 montre le diagramme de classe MonetaryDatatypes.

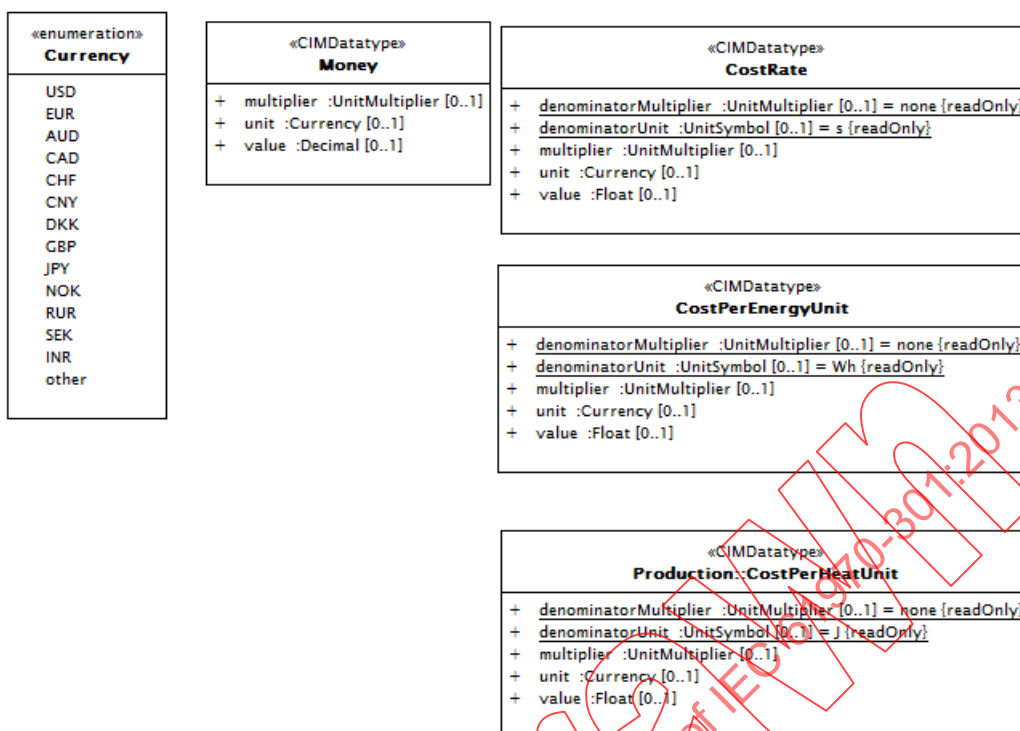


Figure 31 – Diagramme de classe Domain::MonetaryDatatypes

Ce diagramme montre les types de données liés à la monnaie.

La Figure 32 montre le diagramme de classe TimeDatatypes.

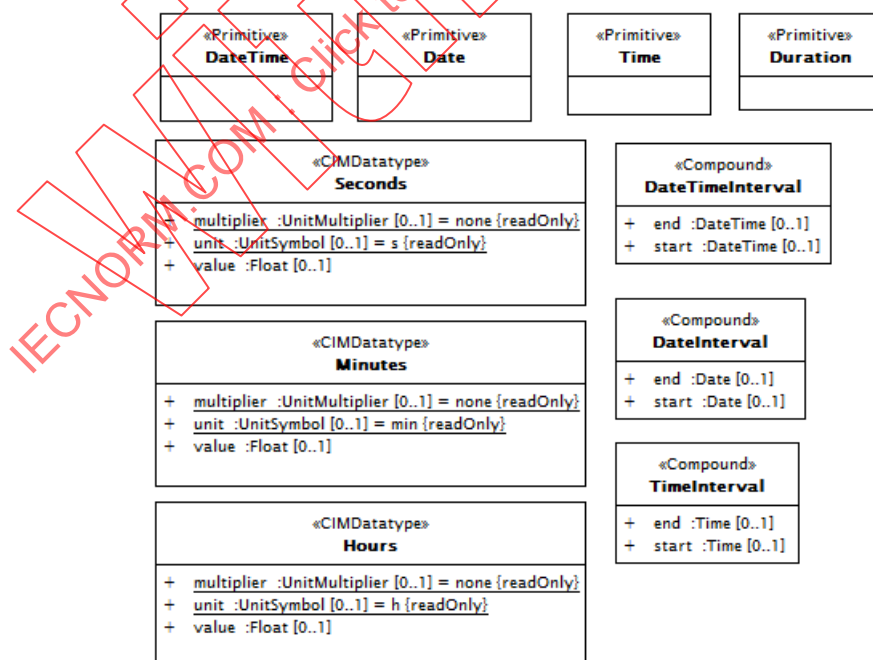


Figure 32 – Diagramme de classe Domain::TimeDatatypes

Ce diagramme représente les types de données liées au temps.

6.3.2 Type de données ActivePower

Produit de la valeur RMS (valeur efficace) de la tension et de la valeur RMS de l'élément en phase du courant.

Le Tableau 8 montre tous les attributs de ActivePower.

Tableau 8 – Attributs de Domain::ActivePower

nom	type	description
value	Float	
unit=W (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.3 Type de données ActivePowerChangeRate

Taux de changement de la puissance active par unité de temps.

Le Tableau 9 montre tous les attributs de ActivePowerChangeRate.

Tableau 9 – Attributs de Domain::ActivePowerChangeRate

nom	type	description
value	Float	
unit=W (const)	UnitSymbol	
multiplier	UnitMultiplier	
denominatorUnit=s (const)	UnitSymbol	
denominatorMultiplier=none (const)	UnitMultiplier	

6.3.4 Type de données Admittance

Rapport courant/tension.

Le Tableau 10 montre tous les attributs de Admittance.

Tableau 10 – Attributs de Domain::Admittance

nom	type	description
value	Float	
unit=S (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.5 Type de données AngleDegrees

Mesure d'angle en degrés.

Le Tableau 11 montre tous les attributs de AngleDegrees.

Tableau 11 – Attributs de Domain::AngleDegrees

nom	type	description
value	Float	
unit=deg (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	

6.3.6 Type de données AngleRadians

Angle de phase en radians.

Le Tableau 12 montre tous les attributs de AngleRadians.

Tableau 12 – Attributs de Domain::AngleRadians

nom	type	description
value	Float	
unit=rad (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	

6.3.7 Type de données ApparentPower

Produit de la valeur RMS (moyenne quadratique) de la tension et de la valeur RMS du courant.

Le Tableau 13 montre tous les attributs de ApparentPower.

Tableau 13 – Attributs de Domain::ApparentPower

nom	type	description
value	Float	
unit=VA (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.8 Primitive Booléen

Un type avec l'espace de valeurs "true" (c'est-à-dire, vrai) et "false" (c'est-à-dire, faux).

6.3.9 Type de données Capacitance

Partie capacitive de la réactance (partie imaginaire de l'impédance), à la fréquence assignée.

Le Tableau 14 montre tous les attributs de Capacitance.

Tableau 14 – Attributs de Domain::Capacitance

nom	type	description
value	Float	
unit=F (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.10 Type de données Conductance

Facteur par lequel doit être multipliée la tension pour obtenir la puissance correspondante perdue dans un circuit. Partie réelle de l'admittance.

Le Tableau 15 montre tous les attributs de Conductance.

Tableau 15 – Attributs de Domain::Conductance

nom	type	description
value	Float	
unit=S (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.11 Type de données ConductancePerLength

Partie réelle de l'admittance par unité de longueur.

Le Tableau 16 montre tous les attributs de ConductancePerLength.

Tableau 16 – Attributs de Domain::ConductancePerLength

nom	type	description
value	Float	
unit=S (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	
denominatorUnit=m (const)	UnitSymbol	
denominatorMultiplier=none (const)	UnitMultiplier	

6.3.12 Type de données CostPerEnergyUnit

Coût, en unités monétaires, par quantité d'énergie électrique produite.

Le Tableau 17 montre tous les attributs de CostPerEnergyUnit.

Tableau 17 – Attributs de Domain::CostPerEnergyUnit

nom	type	description
value	Float	
unit	Currency	
multiplier	UnitMultiplier	
denominatorUnit=Wh (const)	UnitSymbol	
denominatorMultiplier=none (const)	UnitMultiplier	

6.3.13 Type de données CostPerVolume

Coût par unité de volume.

Le Tableau 18 montre tous les attributs de CostPerVolume.

Tableau 18 – Attributs de Domain::CostPerVolume

nom	type	description
denominatorMultiplier=none (const)	UnitMultiplier	
denominatorUnit=m3 (const)	UnitSymbol	
multiplier	UnitMultiplier	
unit	Currency	
value	Float	

6.3.14 Type de données CostRate

Coût en unités monétaires, selon le temps écoulé.

Le Tableau 19 montre tous les attributs de CostRate.

Tableau 19 – Attributs de Domain::CostRate

nom	type	description
value	Float	
unit	Currency	
multiplier	UnitMultiplier	
denominatorUnit=s (const)	UnitSymbol	
denominatorMultiplier=none (const)	UnitMultiplier	

6.3.15 Enumération Currency

Devises monétaires. Veuillez noter que cette liste n'est pas exhaustive.

Le Tableau 20 montre tous les libellés de Currency.

Tableau 20 – Libellés de Domain::Currency

libellé	description
USD	Dollar US
EUR	Euro européen
AUD	Dollar australien
CAD	Dollar canadien
CHF	Franc suisse
CNY	Yuan renminbi chinois
DKK	Couronne danoise
GBP	Livre britannique
JPY	Yen japonais
NOK	Couronne norvégienne
RUR	Rouble russe
SEK	Couronne suédoise
INR	Roupie indienne
other	Autre type de devise

6.3.16 Type de données CurrentFlow

Courant électrique avec le signe conventionnel: un flux positif circule de l'équipement conducteur jusque dans le nœud de connectivité.

Le Tableau 21 montre tous les attributs de CurrentFlow.

Tableau 21 – Attributs de Domain::CurrentFlow

nom	type	description
value	Float	
unit=A (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.17 Type de données Damping

Variation de puissance active par unité avec une fréquence référencée dans la base de puissance apparente du système. Les valeurs types sont comprises entre 1,0 et 2,0.

Le Tableau 22 montre tous les attributs de Damping.

Tableau 22 – Attributs de Domain::Damping

nom	type	description
value	Float	
unit=none (const)	UnitSymbol	
multiplier	UnitMultiplier	
denominatorUnit=Hz (const)	UnitSymbol	
denominatorMultiplier=none (const)	UnitMultiplier	

6.3.18 Type de données primitif Date

Date et heure sous la forme "aaaa-mm-jj" conformément à ISO 8601. Le fuseau horaire TUC est spécifié sous forme "aaaa-mm-jjZ". Un temps TUC par rapport à un fuseau horaire local est spécifié sous la forme "aaaa-mm-jj(+/-)hh:mm".

6.3.19 Compound DateInterval

Intervalle de temps entre deux dates.

Le Tableau 23 montre tous les attributs de DateInterval.

Tableau 23 – Attributs de Domain::DateInterval

nom	type	description
start	Date	Date à laquelle commence cet intervalle de temps.
end	Date	Date à laquelle finit cet intervalle de temps.

6.3.20 Type de données primitif DateTime

Date et heure sous forme "aaaa-mm-jjThh:mm:ss.sss", conformément à ISO 8601. Le fuseau horaire TUC est spécifié sous forme "aaaa-mm-jjThh:mm:ss.sssZ". Un temps TUC par rapport à un fuseau horaire local est spécifié sous la forme "aaaa-mm-jjThh:mm:ss.sss-hh:mm".

6.3.21 Compound DateTimeInterval

Intervalle de temps entre deux dates et instants.

Le Tableau 24 montre tous les attributs de DateTimeInterval.

Tableau 24 – Attributs de Domain::DateTimeInterval

nom	type	description
start	DateTime	Date et heure auxquels commence cet intervalle de temps.
end	DateTime	Date et heure auxquels finit cet intervalle de temps.

6.3.22 Type de données primitif Decimal

Le type Decimal renvoie au système de notation décimal utilisant la base 10 pour représenter les nombres réels.

6.3.23 Type de données Déplacement

Unité de déplacement par rapport à une position de référence, peut donc être négative.

Le Tableau 25 montre tous les attributs de Déplacement.

Tableau 25 – Attributs de Domain::Displacement

nom	type	description
value	Float	
unit=m (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	

6.3.24 Type de données primitif Duration

Durée représentée sous le format "PnYnMnDTnHnMnS", conformément à l'ISO 8601, où nY exprime un nombre d'années, nM un nombre de mois, nD un nombre de jours. La lettre T sépare l'expression de la date de l'expression de l'heure; ensuite, nH identifie un nombre d'heures, nM un nombre de minutes et nS un nombre de secondes. Le nombre de secondes peut être exprimé en nombre décimal alors que tous les autres nombres sont des entiers.

6.3.25 Primitive Float

Un nombre à virgule flottante. La plage est indéterminée et n'est pas limitée.

6.3.26 Type de données FloatQuantity

Grandeur avec valeur float (c'est-à-dire flottante) et informations associées relatives aux unités.

Le Tableau 26 montre tous les attributs de FloatQuantity.

Tableau 26 – Attributs de Domain::FloatQuantity

nom	type	description
value	Float	
unit	UnitSymbol	
multiplier	UnitMultiplier	

6.3.27 Type de données Frequency

Cycles par seconde.

Le Tableau 27 montre tous les attributs de Frequency.

Tableau 27 – Attributs de Domain::Frequency

nom	type	description
value	Float	
unit=Hz (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.28 Type de données Hours

Durée spécifiée en heures.

Le Tableau 28 montre tous les attributs de Hours.

Tableau 28 – Attributs de Domain::Hours

nom	type	description
value	Float	
unit=h (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	

6.3.29 Type de données Impedance

Rapport tension/courant.

Le Tableau 29 montre tous les attributs de Impedance.

Tableau 29 – Attributs de Domain::Impedance

nom	type	description
value	Float	
unit=ohm (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.30 Type de données Inductance

Partie inductive de la réactance (partie imaginaire de l'impédance), à la fréquence assignée.

Le Tableau 30 montre tous les attributs de Inductance.

Tableau 30 – Attributs de Domain::Inductance

nom	type	description
value	Float	
unit=H (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.31 Primitive Integer

Un nombre entier. La plage est indéterminée et n'est pas limitée.

6.3.32 Type de données IntegerQuantity

Grandeur avec valeur integer (c'est-à-dire entière) et informations associées relatives aux unités.

Le Tableau 31 montre tous les attributs de IntegerQuantity.

Tableau 31 – Attributs de Domain::IntegerQuantity

nom	type	description
value	Integer	
unit	UnitSymbol	
multiplier	UnitMultiplier	

6.3.33 Type de données KiloActivePower

Puissance active, en kilowatts.

Le Tableau 32 montre tous les attributs de KiloActivePower.

Tableau 32 – Attributs de Domain::KiloActivePower

nom	type	description
value	Float	
unit=W (const)	UnitSymbol	
multiplier=k (const)	UnitMultiplier	

6.3.34 Type de données Length

Unité de longueur. Jamais négative.

Le Tableau 33 montre tous les attributs de Length.

Tableau 33 – Attributs de Domain::Length

nom	type	description
multiplier	UnitMultiplier	
unit=m (const)	UnitSymbol	
value	Float	

6.3.35 Type de données Minutes

Temps en minutes.

Le Tableau 34 montre tous les attributs de Minutes.

Tableau 34 – Attributs de Domain::Minutes

nom	type	description
value	Float	
unit=min (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	

6.3.36 Type de données Money

Quantité d'argent.

Le Tableau 35 montre tous les attributs de Money.

Tableau 35 – Attributs de Domain::Money

nom	type	description
multiplier	UnitMultiplier	
unit	Currency	
value	Decimal	

6.3.37 Type de données PU

Par unité – valeur négative ou positive se référant à une base définie. Valeurs généralement comprises entre –10 et +10.

Le Tableau 36 montre tous les attributs de PU.

Tableau 36 – Attributs de Domain::PU

nom	type	description
value	Float	
unit=none (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	

6.3.38 Type de données PerCent

Pourcentage sur une base définie. Spécifier par exemple 100 pour indiquer la base définie.

Le Tableau 37 montre tous les attributs de PerCent.

Tableau 37 – Attributs de Domain::PerCent

nom	type	description
value	Float	Normalement 0 à 100 sur une base définie
unit=none (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	

6.3.39 Type de données Pressure

Pression en pascals.

Le Tableau 38 montre tous les attributs de Pressure.

Tableau 38 – Attributs de Domain::Pressure

nom	type	description
value	Float	
unit=Pa (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.40 Type de données Reactance

Réactance (partie imaginaire de l'impédance), à une fréquence assignée.

Le Tableau 39 montre tous les attributs de Reactance.

Tableau 39 – Attributs de Domain::Reactance

nom	type	description
value	Float	
unit=ohm (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.41 Type de données ReactancePerLength

Réactance (partie imaginaire de l'impédance) par unité de longueur, à la fréquence assignée.

Le Tableau 40 montre tous les attributs de ReactancePerLength.

Tableau 40 – Attributs de Domain::ReactancePerLength

nom	type	description
value	Float	
unit=ohm (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	
denominatorUnit=m (const)	UnitSymbol	
denominatorMultiplier=none (const)	UnitMultiplier	

6.3.42 Type de données ReactivePower

Produit de la valeur RMS (valeur efficace) de la tension et de la valeur RMS de la composante en quadrature du courant.

Le Tableau 41 montre tous les attributs de ReactivePower.

Tableau 41 – Attributs de Domain::ReactivePower

nom	type	description
value	Float	
unit=VAR (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.43 Type de données RealEnergy

Énergie électrique réelle.

Le Tableau 42 montre tous les attributs de RealEnergy.

Tableau 42 – Attributs de Domain::RealEnergy

nom	type	description
value	Float	
unit=Wh (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.44 Type de données Resistance

Résistance (partie réelle de l'impédance).

Le Tableau 43 montre tous les attributs de Resistance.

Tableau 43 – Attributs de Domain::Resistance

nom	type	description
value	Float	
unit=ohm (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.45 Type de données ResistancePerLength

Résistance (partie réelle de l'impédance) par unité de longueur.

Le Tableau 44 montre tous les attributs de ResistancePerLength.

Tableau 44 – Attributs de Domain::ResistancePerLength

nom	type	description
value	Float	
unit=ohm (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	
denominatorUnit=m (const)	UnitSymbol	
denominatorMultiplier=none (const)	UnitMultiplier	

6.3.46 Type de données RotationSpeed

Nombre de tours par seconde.

Le Tableau 45 montre tous les attributs de RotationSpeed.

Tableau 45 – Attributs de Domain::RotationSpeed

nom	type	description
value	Float	
unit=none (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	
denominatorUnit=s (const)	UnitSymbol	
denominatorMultiplier=none (const)	UnitMultiplier	

6.3.47 Type de données Seconds

Temps, en secondes.

Le Tableau 46 montre tous les attributs de Seconds.

Tableau 46 – Attributs de Domain::Seconds

nom	type	description
value	Float	Temps, en secondes
unit=s (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	

6.3.48 Type de données Speed

Distance par unité de temps.

Le Tableau 47 montre tous les attributs de Speed.

Tableau 47 – Attributs de Domain::Speed

nom	type	description
value	Float	
unit=m (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	
denominatorUnit=s (const)	UnitSymbol	
denominatorMultiplier=none (const)	UnitMultiplier	

6.3.49 Primitive String

Une chaîne constituée d'une séquence de caractères 8 bits. Le codage de caractères est UTF-8. La longueur de la chaîne n'est ni spécifiée ni limitée.

6.3.50 Type de données StringQuantity

Grandeur avec valeur chaîne de caractères (lorsqu'il importe peu qu'il s'agisse d'un entier ou d'un nombre à virgule flottante) et les informations associées relatives aux unités.

Le Tableau 48 montre tous les attributs de StringQuantity.

Tableau 48 – Attributs de Domain::StringQuantity

nom	type	description
value	String	
unit	UnitSymbol	
multiplier	UnitMultiplier	

6.3.51 Type de données Susceptance

Partie imaginaire de l'admittance.

Le Tableau 49 montre tous les attributs de Susceptance.

Tableau 49 – Attributs de Domain::Susceptance

nom	type	description
value	Float	
unit=S (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.52 Type de données SusceptancePerLength

Partie imaginaire de l'admittance par unité de longueur.

Le Tableau 50 montre tous les attributs de SusceptancePerLength.

Tableau 50 – Attributs de Domain::SusceptancePerLength

nom	type	description
value	Float	
unit=S (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	
denominatorUnit=m (const)	UnitSymbol	
denominatorMultiplier=none (const)	UnitMultiplier	

6.3.53 Type de données Temperature

Valeur de température en degrés Celsius.

Le Tableau 51 montre tous les attributs de Temperature.

Tableau 51 – Attributs de Domain::Temperature

nom	type	description
multiplier=none (const)	UnitMultiplier	
unit=degC (const)	UnitSymbol	
value	Float	

6.3.54 Type de données primitif Time

Heure sous la forme "hh:mm:ss.sss", conformément à ISO 8601. Le fuseau horaire TUC est spécifié sous forme "hh:mm:ss.sssZ". Un temps TUC par rapport à un fuseau horaire local est spécifié sous la forme "hh:mm:ss.sss±hh:mm".

6.3.55 Composé TimeInterval

Intervalle de temps entre deux instants.

Le Tableau 52 montre tous les attributs de TimeInterval.

Tableau 52 – Attributs de Domain::TimeInterval

nom	type	description
start	Durée	Instant auquel commence cet intervalle de temps.
end	Durée	Instant auquel finit cet intervalle de temps.

6.3.56 Enumération UnitMultiplier

Les multiplicateurs d'unité définis pour le CIM.

Le Tableau 53 montre tous les libellés de UnitMultiplier.

Tableau 53 – Libellés de Domain::UnitMultiplier

libellé	description
p	Pico 10** ⁻¹² .
n	Nano 10** ⁻⁹ .
micro	Micro 10** ⁻⁶ .
m	Milli 10** ⁻³ .
c	Centi 10** ⁻² .
d	Déci 10** ⁻¹ .
k	Kilo 10** ³ .
M	Méga 10** ⁶ .
G	Giga 10** ⁹ .
T	Téra 10** ¹² .
néant	Pas de multiplicateur ou, ce qui revient au même, multiplier par 1.

6.3.57 Enumération UnitSymbol

Les unités définies en vue de leur utilisation dans le CIM.

Le Tableau 54 montre tous les libellés de UnitSymbol.

Tableau 54 – Libellés de Domain::UnitSymbol

libellé	description
VA	Puissance apparente en volts ampères.
W	Puissance active en watts.
VA _r	Puissance réactive en volts ampères réactifs.
VA _h	Energie apparente en heures volt-ampère.
Wh	Energie réelle en wattheures.
VA _r h	Energie réactive en heures réactives volt-ampère.
V	Tension en volts.
ohm	Résistance en ohms.
A	Courant en ampères.
F	Capacité en farads.
H	Inductance en henrys.
degC	Température relative en degrés Celsius. Le symbole est °C dans le système international d'unités (SI). La charge électrique est mesurée en coulombs qui a pour symbole C. Pour distinguer les degrés Celsius et le coulomb, le symbole utilisé dans l'UML est degC. La raison pour laquelle on n'utilise pas °C tient à ce qu'il est difficile de gérer le caractère spécial ° dans le logiciel.
s	Temps en secondes.
min	Temps en minutes.
h	Temps en heures.
deg	Angle plan en degrés.
rad	Angle plan en radians.
J	Energie en joules.
N	Force en newtons.
S	Conductance en siemens.
néant	Grandeur sans dimension, par exemple: compte, par unité, etc.
Hz	Fréquence en hertz.
g	Masse en grammes.
Pa	Pression en pascals (N/m ²).
m	Longueur en mètres.
m ²	Surface en mètres carrés.
m ³	Volume en mètres cubes.

6.3.58 Type de données Voltage

Tension électrique.

Le Tableau 55 montre tous les attributs de Voltage.

Tableau 55 – Attributs de Domain::Voltage

nom	type	description
value	Float	
unit=V (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.59 Type de données VoltagePerReactivePower

Variation de la tension avec la puissance réactive.

Le Tableau 56 montre tous les attributs de VoltagePerReactivePower.

Tableau 56 – Attributs de Domain::VoltagePerReactivePower

nom	type	description
value	Float	
unit=V (const)	UnitSymbol	
multiplier	UnitMultiplier	
denominatorUnit=VA _r (const)	UnitSymbol	
denominatorMultiplier	UnitMultiplier	

6.3.60 Type de données Volume

Volume.

Le Tableau 57 montre tous les attributs de Volume.

Tableau 57 – Attributs de Domain::Volume

nom	type	description
value	Float	
unit=m ³ (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.61 Type de données VolumeFlowRate

Volume par unité de temps.

Le Tableau 58 montre tous les attributs de VolumeFlowRate.

Tableau 58 – Attributs de Domain::VolumeFlowRate

nom	type	description
denominatorMultiplier=none (const)	UnitMultiplier	
denominatorUnit=s (const)	UnitSymbol	
multiplier=none (const)	UnitMultiplier	
unit=m ³ (const)	UnitSymbol	
value	Float	

6.3.62 Type de données WaterLevel

Niveau d'eau présente dans le réservoir par rapport à une référence de type niveau moyen de la mer.

Le Tableau 59 montre tous les attributs de WaterLevel.

Tableau 59 – Attributs de Domain::WaterLevel

nom	type	description
value	Float	
unit=m (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.3.63 Type de données Weight

Le poids d'un objet.

Le Tableau 60 montre tous les attributs de Weight.

Tableau 60 – Attributs de Domain::Weight

nom	type	description
value	Float	
unit=g (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.4 Paquetage Core

6.4.1 Généralités

Contient les entités centrales PowerSystemResource (ressource du réseau électrique) et ConductingEquipment (équipement conducteur) partagées par toutes les applications plus les collections communes de ces entités. Toutes les applications ne nécessitent pas toutes les entités Core. Ce paquetage ne dépend d'aucun autre paquetage (à l'exception du paquetage Domain), mais en revanche, la plupart des autres paquetages ont des associations et des généralisations qui dépendent de ce paquetage.

La Figure 33 montre le diagramme de classe Main.

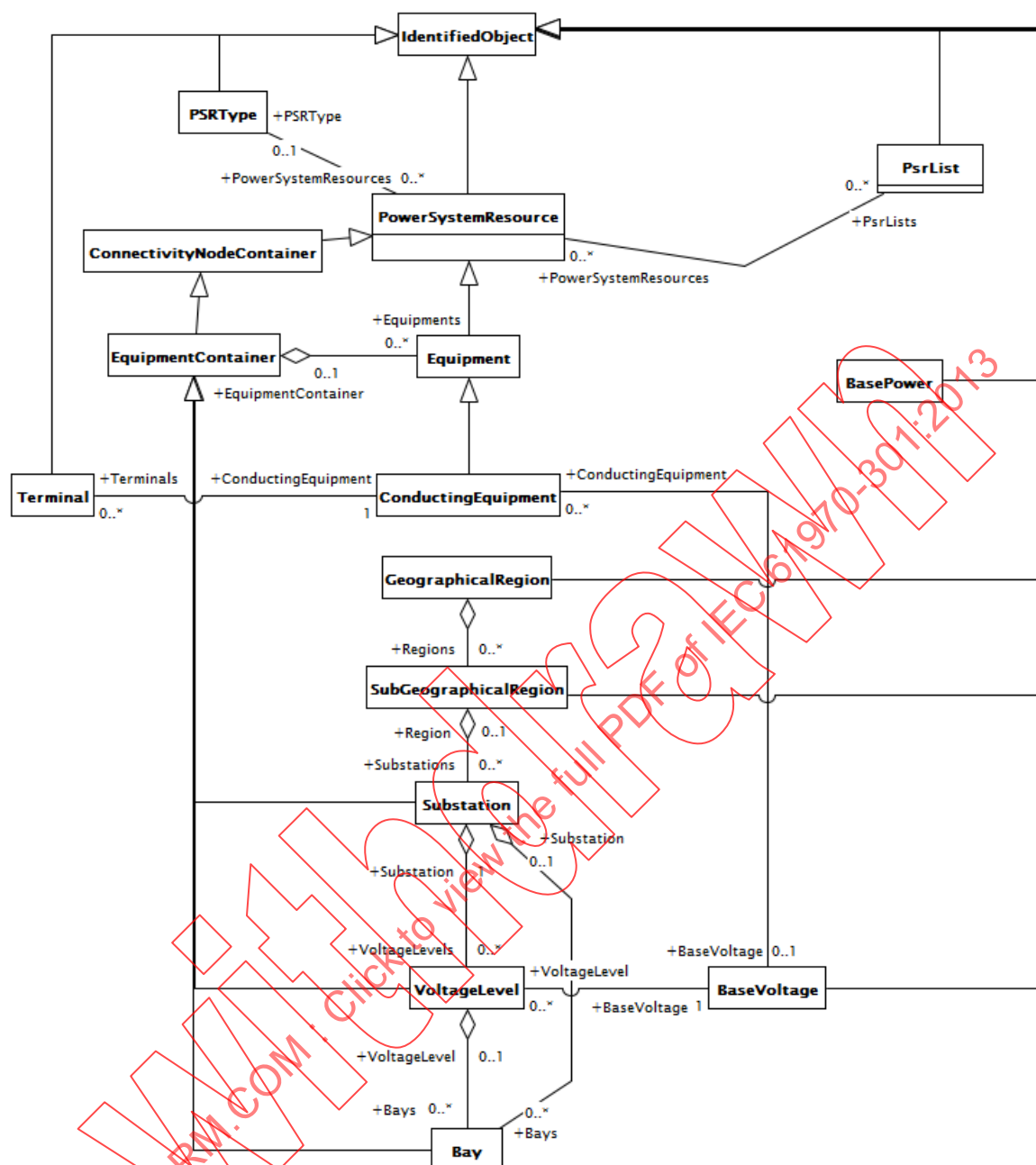


Figure 33 – Diagramme de classe Core::Main

Ce diagramme montre les principales classes nécessaires à toute application du CIM pour modéliser des systèmes de transport ou de production.

La Figure 34 montre le diagramme de classe Names.

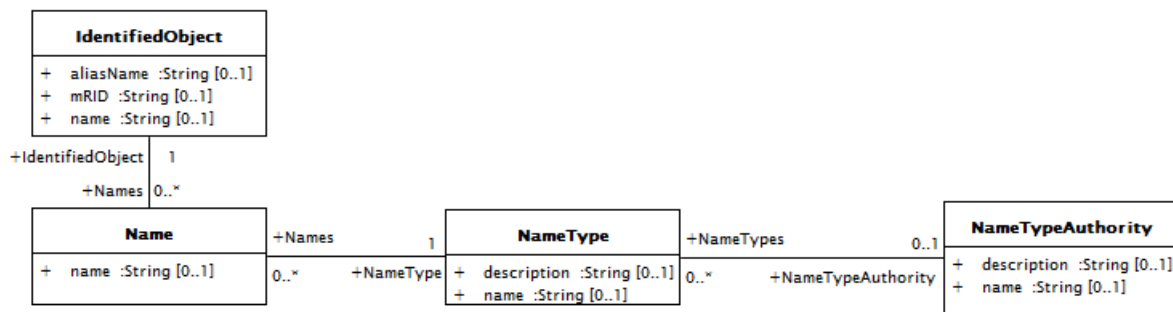


Figure 34 – Diagramme de classe Core::Names

Ce diagramme montre les classes utilisées pour attribuer des noms aux objets identifiés.

La Figure 35 montre le diagramme de classe CurveSchedule.

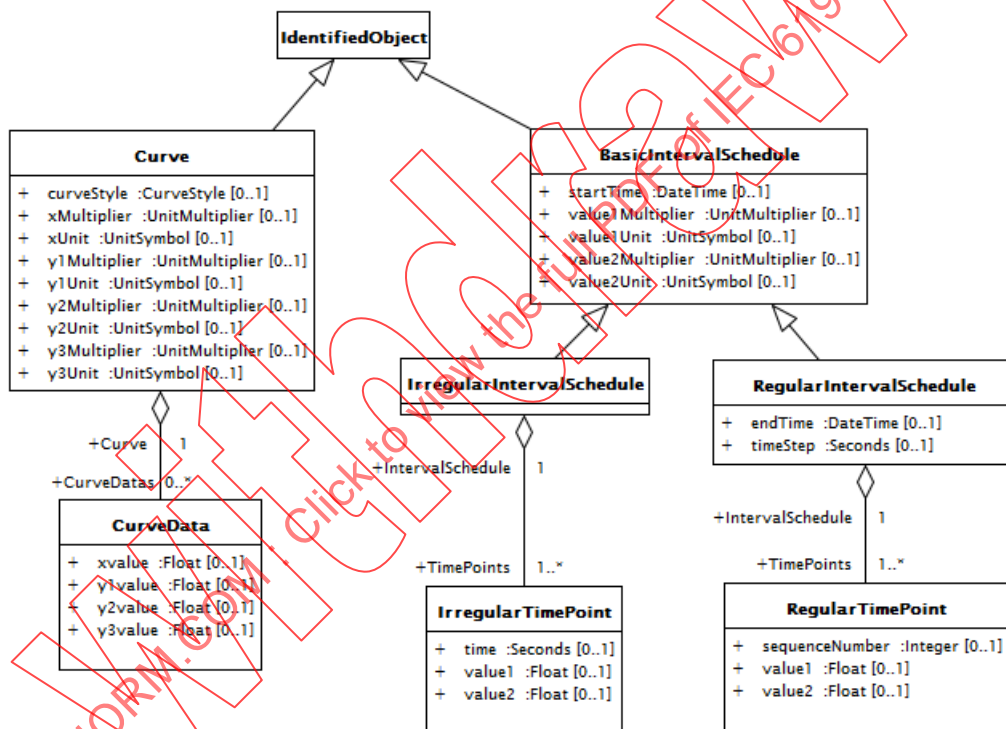


Figure 35 – Diagramme de classe Core::CurveSchedule

Ce diagramme montre le modèle commun pour des courbes et des programmes liés au temps.

La Figure 36 montre le diagramme de classe Datatypes.

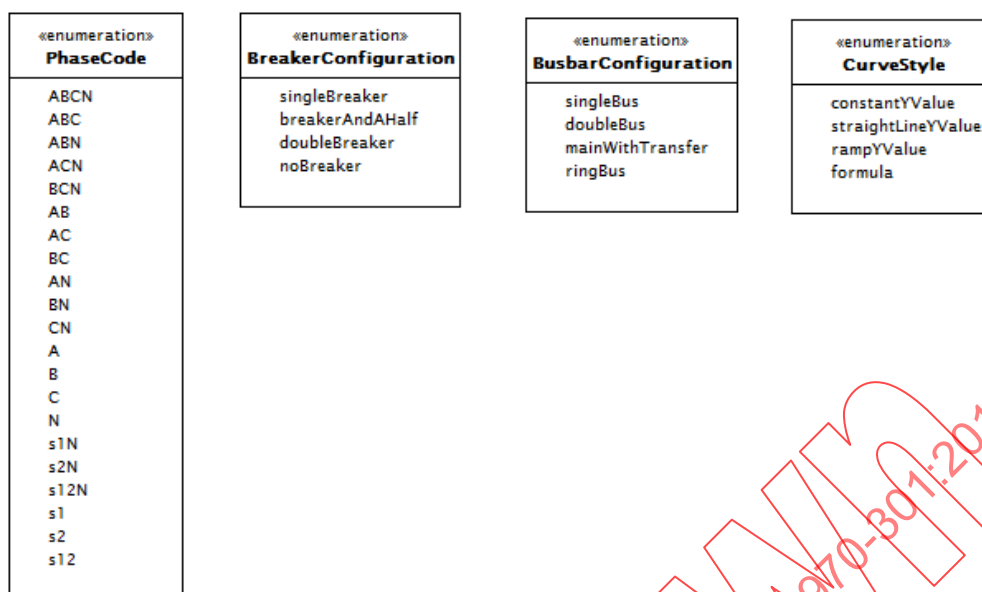


Figure 36 – Diagramme de classe Core::Datatypes

Ce diagramme montre les types de données spécifiques au paquetage Core.

La Figure 37 montre le diagramme de classe Reporting.

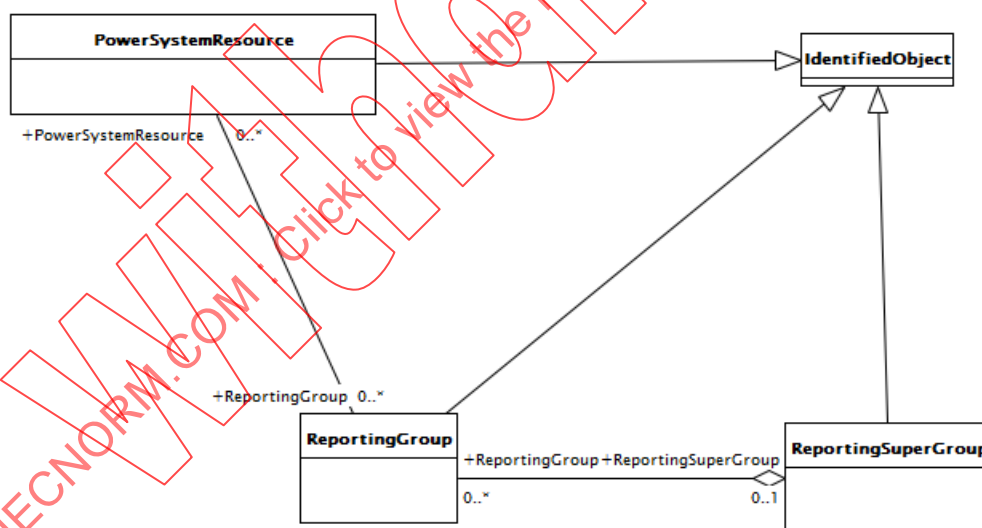


Figure 37 – Diagramme de classe Core::Reporting

Ce diagramme montre le modèle commun pour le reporting ad hoc.

La Figure 38 montre le diagramme de classe OperatingShare.

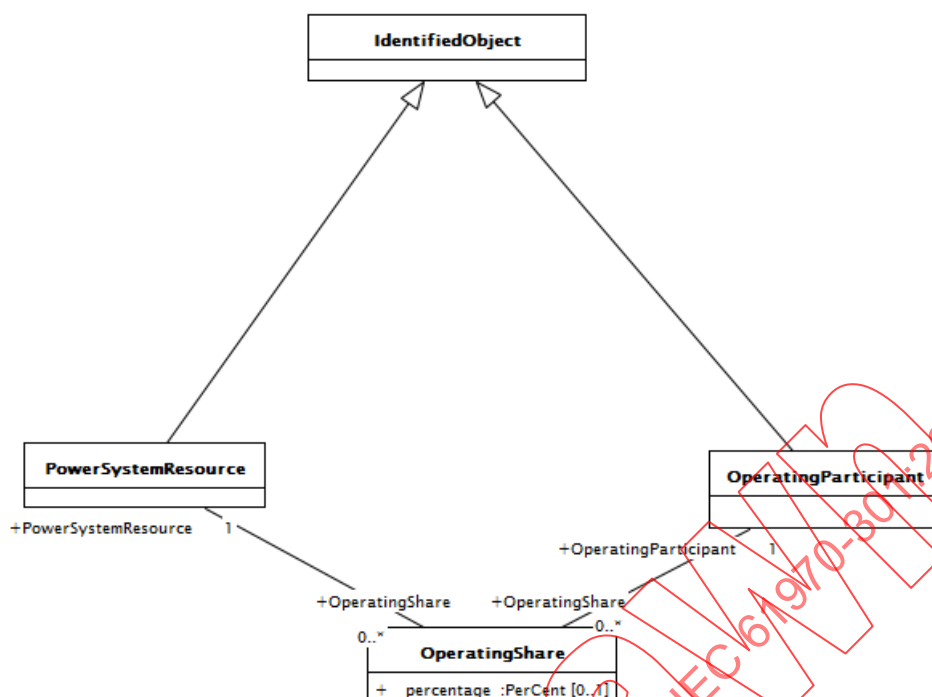


Figure 38 – Diagramme de classe Core::OperatingShare

Ce diagramme montre le modèle commun pour la quote-part d'exploitation.

6.4.2 BasePower

La classe BasePower définit la puissance de base dans les calculs par unité.

Le Tableau 61 montre tous les attributs de BasePower.

Tableau 61 – Attributs de Core::BasePower

nom	type	description
basePower	ApparentPower	Valeur utilisée comme puissance de base.
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 62 montre toutes les extrémités d'association de BasePower avec d'autres classes.

Tableau 62 – Extrémités d'association de Core::BasePower avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.4.3 BaseVoltage

Définit une tension de base du système qui est référencée.

Le Tableau 63 montre tous les attributs de BaseVoltage.

Tableau 63 – Attributs de Core::BaseVoltage

nom	type	description
isDC	Boolean	Si true (c'est-à-dire vrai), il s'agit d'une tension de base en courant continu et les éléments affectés à cette tension de base sont aussi associés à des fonctionnalités en courant continu. False (c'est-à-dire faux) indique le courant alternatif.
nominalVoltage	Voltage	Tension de base des composants du réseau électrique.
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 64 montre toutes les extrémités d'association de BaseVoltage avec d'autres classes.

Tableau 64 – Extrémités d'association de Core::BaseVoltage avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[0..*] ConductingEquipment	ConductingEquipment	Tout équipement conducteur qui a cette tension de base. A n'utiliser que quand aucun conteneur de niveau de tension n'est utilisé et que s'applique une seule tension de base. Par exemple, n'est pas utilisé pour des transformateurs.
[1..1]	[0..*] VoltageLevel	VoltageLevel	Niveaux de tension ayant cette tension de base.
[0..1]	[0..*] TopologicalNode	TopologicalNode	Les nœuds topologiques à la tension de base.
[0..1]	[0..*] TransformerEnds	TransformerEnd	Extrémités de transformateur à la tension de base. Primordial pour le calcul de PU.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.4.4 BasicIntervalSchedule

Programme de valeurs à des instants donnés.

Le Tableau 65 montre tous les attributs de BasicIntervalSchedule.

Tableau 65 – Attributs de Core::BasicIntervalSchedule

nom	type	description
startTime	DateTime	L'heure pour le premier TimePoint.
value1Unit	UnitSymbol	Unités de mesure pour value1.
value1Multiplier	UnitMultiplier	Multiplicateur pour value1.
value2Unit	UnitSymbol	Unités de mesure pour value2.
value2Multiplier	UnitMultiplier	Multiplicateur pour value2.
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 66 montre toutes les extrémités d'association de BasicIntervalSchedule avec d'autres classes.

Tableau 66 – Extrémités d'association de Core::BasicIntervalSchedule avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.4.5 Bay

Collection de ressources du réseau électrique (au sein d'un poste donné) comprenant équipement conducteur, relais de protection, mesures et télémétrie. Une cellule représente typiquement un groupement physique lié à la modularisation de l'équipement.

Le Tableau 67 montre tous les attributs de Bay.

Tableau 67 – Attributs de Core::Bay

nom	type	description
bayEnergyMeasFlag	Boolean	Indique la présence/l'absence de mesures d'énergie.
bayPowerMeasFlag	Boolean	Indique la présence/l'absence de mesures de puissance active/réactive.
breakerConfiguration	BreakerConfiguration	Configuration de l'organe de coupure.
busBarConfiguration	BusbarConfiguration	Configuration du jeu de barres.
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 68 montre toutes les extrémités d'association de Bay avec d'autres classes.

Tableau 68 – Extrémités d'association de Core::Bay avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] Substation	Substation	Poste qui contient la cellule.
[0..*]	[0..1] VoltageLevel	VoltageLevel	Niveau de tension qui contient cette cellule.
[0..1]	[0..*] Equipments	Equipement	hérité de: EquipmentContainer
[1..1]	[0..*] ConnectivityNodes	ConnectivityNode	hérité de: ConnectivityNodeContainer
[0..1]	[0..*] TopologicalNode	TopologicalNode	hérité de: ConnectivityNodeContainer
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.4.6 Enumération BreakerConfiguration

Disposition de commutation pour Bay.

Le Tableau 69 montre tous les libellés de BreakerConfiguration.

Tableau 69 – Libellés de Core::BreakerConfiguration

libellé	description
singleBreaker	Un seul organe de coupure.
breakerAndAHalf	Un organe de coupure et demi.
doubleBreaker	Double organe de coupure.
noBreaker	Aucun organe de coupure.

6.4.7 Enumération BusbarConfiguration

Disposition du jeu de barres pour Bay.

Le Tableau 70 montre tous les libellés de BusbarConfiguration.

Tableau 70 – Libellés de Core::BusbarConfiguration

libellé	description
singleBus	Une seule barre.
doubleBus	Double barre.
mainWithTransfer	Barre principale avec barre de transfert.
ringBus	Barre en anneau.

6.4.8 ConductingEquipment

Parties du réseau électrique conçues pour transporter du courant ou connectées par des bornes de façon à assurer la conduction.

Le Tableau 71 montre tous les attributs de ConductingEquipment.

Tableau 71 – Attributs de Core::ConductingEquipment

nom	type	description
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 72 montre toutes les extrémités d'association de ConductingEquipment avec d'autres classes.

Tableau 72 – Extrémités d'association de Core::ConductingEquipment avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] BaseVoltage	BaseVoltage	Tension de base pour cet équipement conducteur. A n'utiliser que quand aucun conteneur de niveau de tension n'est utilisé et que s'applique une seule tension de base. Par exemple, n'est pas utilisé pour des transformateurs.
[1..1]	[0..*] Terminals	Terminal	L'équipement conducteur a des bornes qui peuvent être connectées à d'autres bornes d'un équipement conducteur par des nœuds de connectivité ou des nœuds topologiques.
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	Dispositif de protection utilisé pour protéger un équipement conducteur spécifique.
[1..1]	[0..*] ClearanceTags	ClearanceTag	Les marqueurs de consignation pour un champ autorisé fonctionnent sur cet équipement conducteur.
[1..1]	[0..1] SvStatus	SvStatus	Variable d'état statut associée à cet équipement conducteur.
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.4.9 ConnectivityNode

Les Connectivity nodes (nœuds de connectivité) sont des points où les bornes d'un équipement conducteur sont connectées avec une impédance nulle.

Le Tableau 73 montre tous les attributs de ConnectivityNode.

Tableau 73 – Attributs de Core::ConnectivityNode

nom	type	description
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 74 montre toutes les extrémités d'association de ConnectivityNode avec d'autres classes.

Tableau 74 – Extrémités d'association de Core::ConnectivityNode avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[1..1] ConnectivityNodeContainer	ConnectivityNodeContainer	Conteneur de ce nœud de connectivité
[0..*]	[0..1] TopologicalNode	TopologicalNode	Nœud topologique auquel est affecté ce nœud de connectivité. Peut dépendre de l'état courant de commutateurs dans le réseau.
[0..1]	[0..*] Terminals	Terminal	Bornes interconnectées à impédance zéro à ce nœud de connectivité.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.4.10 ConnectivityNodeContainer

Classe de base pour tous les objets qui peuvent contenir des nœuds de connectivité ou des nœuds topologiques.

Le Tableau 75 montre tous les attributs de ConnectivityNodeContainer.

Tableau 75 – Attributs de Core::ConnectivityNodeContainer

nom	type	description
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 76 montre toutes les extrémités d'association de ConnectivityNodeContainer avec d'autres classes.

Tableau 76 – Extrémités d'association de Core::ConnectivityNodeContainer avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[1..1]	[0..*] ConnectivityNodes	ConnectivityNode	Les nœuds de connectivité qui appartiennent à ce conteneur de nœuds de connectivité.
[0..1]	[0..*] TopologicalNode	TopologicalNode	Les nœuds topologiques qui appartiennent à ce conteneur de nœuds de connectivité.
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.4.11 Curve

Courbe d'usage général ou relation fonctionnelle entre une variable indépendante (axe X) et des variables dépendantes (axe Y).

Le Tableau 77 montre tous les attributs de Curve.

Tableau 77 – Attributs de Core::Curve

nom	type	description
curveStyle	CurveStyle	Style ou forme de la courbe.
xMultiplier	UnitMultiplier	Multiplicateur pour l'axe X.
xUnit	UnitSymbol	Unités de mesure de l'axe X.
y1Multiplier	UnitMultiplier	Multiplicateur pour l'axe Y1.
y1Unit	UnitSymbol	Unités de mesure de l'axe Y1.
y2Multiplier	UnitMultiplier	Multiplicateur pour l'axe Y2.
y2Unit	UnitSymbol	Unités de mesure de l'axe Y2.
y3Multiplier	UnitMultiplier	Multiplicateur pour l'axe Y3.
y3Unit	UnitSymbol	Unités de mesure de l'axe Y3.
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 78 montre toutes les extrémités d'association de Curve avec d'autres classes.

Tableau 78 – Extrémités d'association de Core::Curve avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[1..1]	[0..*] CurveDatas	CurveData	Valeurs de données du point définissant cette courbe
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.4.12 Classe racine CurveData

Points de données d'usage général pour définir une courbe. L'utilisation de cette classe générique est déconseillée si une classe plus spécifique peut être utilisée pour spécifier les valeurs des axes x et y ainsi que leurs types de données spécifiques.

Le Tableau 79 montre tous les attributs de CurveData.

Tableau 79 – Attributs de Core::CurveData

nom	type	description
xvalue	Float	La valeur de données de la variable de l'axe X, en fonction des unités de l'axe X.
y1value	Float	La valeur de données de la première variable de l'axe Y, en fonction des unités de l'axe Y.
y2value	Float	La valeur de données de la deuxième variable de l'axe Y (s'il y en a une), en fonction des unités de l'axe Y.
y3value	Float	La valeur de données de la troisième variable de l'axe Y (s'il y en a une), en fonction des unités de l'axe Y.

Le Tableau 80 montre toutes les extrémités d'association de CurveData avec d'autres classes.

Tableau 80 – Extrémités d'association de Core::CurveData avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[1..1] Curve	Curve	Courbe de ce point de données de courbe.

6.4.13 Enumération CurveStyle

Style ou forme de la courbe

Le Tableau 81 montre tous les libellés de CurveStyle.

Tableau 81 – Libellés de Core::CurveStyle

libellé	description
constantYValue	Les valeurs de l'axe Y sont supposées constantes jusqu'au point suivant de courbe et avant le premier point de courbe.
straightLineYValues	Les valeurs de l'axe Y sont supposées former une ligne droite entre les valeurs. Aussi appelé interpolation linéaire.
rampYValue	Les valeurs de l'axe Y sont supposées présenter une pente entre les points.
formula	Une formule non spécifiée est supposée calculer la valeur de l'axe Y entre les points.

6.4.14 Equipement

Les éléments d'une ressource de réseau électrique qui sont des dispositifs matériels, électroniques ou mécaniques.

Le Tableau 82 montre tous les attributs de Equipment.

Tableau 82 – Attributs de Core::Equipment

nom	type	description
aggregate	Boolean	L'instance unique de l'équipement représente plusieurs éléments d'équipement qui ont été modélisés ensemble comme un agrégat. On peut donner pour exemple de cela des transformateurs de puissance ou des machines synchrones fonctionnant en parallèle, modélisées comme un seul agrégat de transformateur de puissance ou agrégat de machine synchrone. Cela ne doit pas être utilisé pour indiquer un équipement qui est partie intégrante d'un groupe d'équipements interdépendants produits par un programme de production de réseau.
normallyInService	Boolean	Si true (c'est-à-dire vrai), l'équipement est normalement en service.
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 83 montre toutes les extrémités d'association de Equipment avec d'autres classes.

Tableau 83 – Extrémités d'association de Core::Equipment avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	Conteneur de cet équipement.
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	Equipements de contingence auxquels cet équipement participe
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	Jeux de limites opérationnelles associés à cet équipement.
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.4.15 EquipmentContainer

Un choix de modélisation pour proposer une classe racine qui regroupe des équipements.

Le Tableau 84 montre tous les attributs de EquipmentContainer.

Tableau 84 – Attributs de Core::EquipmentContainer

nom	type	description
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 85 montre toutes les extrémités d'association de EquipmentContainer avec d'autres classes.

Tableau 85 – Extrémités d'association de Core::EquipmentContainer avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[0..*] Equipments	Equipement	Equipement contenu.
[1..1]	[0..*] ConnectivityNodes	ConnectivityNode	hérité de: ConnectivityNodeContainer
[0..1]	[0..*] TopologicalNode	TopologicalNode	hérité de: ConnectivityNodeContainer
[0..*]	[0..1] PSRTYPE	PSRTYPE	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.4.16 GeographicalRegion

Une région géographique d'un modèle de réseau de système de puissance.

Le Tableau 86 montre tous les attributs de GeographicalRegion.

Tableau 86 – Attributs de Core::GeographicalRegion

nom	type	description
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 87 montre toutes les extrémités d'association de GeographicalRegion avec d'autres classes.

Tableau 87 – Extrémités d'association de Core::GeographicalRegion avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[0..*] Regions	SubGeographicalRegion	Toutes les sous-régions géographiques de cette région géographique.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.4.17 Classe racine IdentifiedObject

Il s'agit d'une classe racine qui fournit une identification commune pour toutes les classes nécessitant des attributs d'identification et de dénomination.

Le Tableau 88 montre tous les attributs de IdentifiedObject.

Tableau 88 – Attributs de Core::IdentifiedObject

nom	type	description
aliasName	String	L'aliasName est un nom de l'objet en texte libre lisible par l'homme en remplacement de IdentifiedObject.name. Il peut ne pas être unique et peut ne pas se corréler à une hiérarchie de dénomination. L'attribut aliasName a été retenu parce qu'il est compatible avec les précédentes versions du CIM. Il est cependant recommandé de remplacer aliasName par la classe Nom puisqu'il est prévu de supprimer aliasName à l'avenir.
mRID	String	Identificateur Master resource publié par une autorité de modèle. Le mRID (identificateur Master resource) est globalement unique dans un contexte d'échange. Il est facile d'obtenir une unicité globale en utilisant un UUID pour le mRID. Il est fortement recommandé de le faire. Pour des fichiers de données CIMXML en syntaxe RDF, le mRID est cartographié en attributs rdf:ID ou rdf:about qui identifient les éléments d'objet du CIM.
name	String	Le nom est un texte, librement lisible par l'homme et si possible non unique, qui nomme l'objet.

Le Tableau 89 montre toutes les extrémités d'association de IdentifiedObject avec d'autres classes.

Tableau 89 – Extrémités d'association de Core::IdentifiedObject avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[0..*] DiagramObjects	DiagramObject	Objets de diagramme qui sont associés à l'objet de domaine.
[1..1]	[0..*] Names	Name	Tous les noms de cet objet identifié.

6.4.18 IrregularIntervalSchedule

Le programme comporte plusieurs instants entre lesquels le temps varie.

Le Tableau 90 montre tous les attributs de IrregularIntervalSchedule.

Tableau 90 – Attributs de Core::IrregularIntervalSchedule

nom	type	description
startTime	DateTime	hérité de: BasicIntervalSchedule
value1Unit	UnitSymbol	hérité de: BasicIntervalSchedule
value1Multiplier	UnitMultiplier	hérité de: BasicIntervalSchedule
value2Unit	UnitSymbol	hérité de: BasicIntervalSchedule
value2Multiplier	UnitMultiplier	hérité de: BasicIntervalSchedule
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 91 montre toutes les extrémités d'association de IrregularIntervalSchedule avec d'autres classes.

Tableau 91 – Extrémités d'association de Core::IrregularIntervalSchedule avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[1..1]	[1..*] TimePoints	IrregularTimePoint	Valeurs de données du point définissant une courbe.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.4.19 Classe racine IrregularTimePoint

Des TimePoints pour un programme où le temps entre les points varie.

Le Tableau 92 montre tous les attributs de IrregularTimePoint.

Tableau 92 – Attributs de Core::IrregularTimePoint

nom	type	description
temps	Seconds	Le temps est lié à l'instant où commence le programme.
value1	Float	La première valeur au temps indiqué. La signification de cette valeur est définie par le type dérivé du programme associé.
value2	Float	La deuxième valeur au temps indiqué. La signification de cette valeur est définie par le type dérivé du programme associé.

Le Tableau 93 montre toutes les extrémités d'association de IrregularTimePoint avec d'autres classes.

Tableau 93 – Extrémités d'association de Core::IrregularTimePoint avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[1..*]	[1..1] IntervalSchedule	IrregularIntervalSchedule	Un IrregularTimePoint appartient à un IrregularIntervalSchedule.

6.4.20 Classe racine Name

La classe Name donne les moyens de définir un nombre quelconque de noms, lisibles par l'homme, pour un objet Un nom ne doit pas être utilisé pour définir des relations entre objets. Utiliser plutôt l'identification d'objet 'mRID' pour les relations entre objets.

Le Tableau 94 montre tous les attributs de Name.

Tableau 94 – Attributs de Core::Name

nom	type	description
name	String	Tout texte libre qui désigne l'objet.

Le Tableau 95 montre toutes les extrémités d'association de Name avec d'autres classes.

Tableau 95 – Extrémités d'association de Core::Name avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[1..1] IdentifiedObject	IdentifiedObject	Objet identifié que désigne ce nom.
[0..*]	[1..1] NameType	NameType	Type de ce nom.

6.4.21 Classe racine NameType

Type de nom. Les valeurs possibles pour l'attribut 'name' dépendent de la mise en œuvre mais des profils normatifs peuvent spécifier des types. Une entreprise peut avoir plusieurs systèmes informatiques dont chacun a son propre nom local pour le même objet, par exemple un système de planification peut utiliser d'autres noms qu'un système de gestion de l'énergie (EMS). Un objet peut aussi avoir des noms différents dans le même système informatique, par exemple localName tel que le définit le CIM version 14. La définition du CIM14 est:

Le localName (c'est-à-dire nom local) est un nom de l'objet lisible par l'homme. C'est un nom en texte libre se trouvant à un nœud dans une hiérarchie de dénomination similaire à une structure de répertoire de fichiers. Une hiérarchie de dénomination relative à système électrique peut être la suivante: Substation, VoltageLevel, Equipment etc. Dans une telle hiérarchie, les enfants du même parent ont des noms qui sont typiquement uniques les uns par rapport aux autres.

Le Tableau 96 montre tous les attributs de NameType.

Tableau 96 – Attributs de Core::NameType

nom	type	description
name	String	Nom du type de nom.
description	String	Description du type de nom.

Le Tableau 97 montre toutes les extrémités d'association de NameType avec d'autres classes.

Tableau 97 – Extrémités d'association de Core::NameType avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] NameTypeAuthority	NameTypeAuthority	Autorité en charge de la gestion des noms de ce type.
[1..1]	[0..*] Names	Name	Tous les noms de ce type.

6.4.22 Classe racine NameTypeAuthority

Autorité responsable de créer et de gérer les noms d'un type donné; typiquement une organisation ou un système d'entreprises.

Le Tableau 98 montre tous les attributs de NameTypeAuthority.

Tableau 98 – Attributs de Core::NameTypeAuthority

nom	type	description
name	String	Nom de l'autorité pour le type de nom.
description	String	Description de l'autorité pour le type de nom.

Le Tableau 99 montre toutes les extrémités d'association de NameTypeAuthority avec d'autres classes.

Tableau 99 – Extrémités d'association de Core::NameTypeAuthority avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[0..*] NameTypes	NameType	Tous les types de noms gérés par cette autorité.

6.4.23 OperatingParticipant

Opérateur de plusieurs objets de ressource de réseau électrique. Noter que plusieurs participants opérateurs peuvent exploiter le même objet de ressource de réseau électrique. Cela peut être utilisé pour modéliser des unités détenues en copropriété où chaque propriétaire agit selon une quote-part contractuelle.

Le Tableau 100 montre tous les attributs de OperatingParticipant.

Tableau 100 – Attributs de Core::OperatingParticipant

nom	type	description
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 101 montre toutes les extrémités d'association de OperatingParticipant avec d'autres classes.

Tableau 101 – Extrémités d'association de Core::OperatingParticipant avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[1..1]	[0..*] OperatingShare	OperatingShare	Les quotes-parts d'exploitation d'un participant opérateur. Un participant opérateur peut être réutilisé pour un nombre quelconque de composants du réseau électrique.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.4.24 Classe racine OperatingShare

Spécifie les relations contractuelles d'exploitation entre une ressource de réseau électrique et un participant au contrat.

Le Tableau 102 montre tous les attributs de OperatingShare.

Tableau 102 – Attributs de Core::OperatingShare

nom	type	description
percentage	PerCent	Pourcentage de possession opérationnelle de la paire (composant de réseau électrique et participant opérateur) associée à cette quote-part. Il convient que la possession totale en pourcentage d'un composant de réseau électrique soit égale à 100%.

Le Tableau 103 montre toutes les extrémités d'association de OperatingShare avec d'autres classes.

Tableau 103 – Extrémités d'association de Core::OperatingShare avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[1..1] OperatingParticipant	OperatingParticipant	Participant opérateur qui a cette quote-part avec le composant de réseau électrique associé.
[0..*]	[1..1] PowerSystemResource	PowerSystemResource	Ressource de réseau électrique à laquelle s'applique cette quote-part.

6.4.25 Énumération PhaseCode

Énumération des identificateurs de phase. Permet de désigner des phases pour l'équipement, les circuits et les charges à la fois de transport et de distribution.

Les charges résidentielles et celles de petits commerces sont souvent fournies à partir de circuits secondaires monophasés, ou en phase auxiliaire. En prenant pour exemple s12N, les phases 1 et 2 se réfèrent à des fils chauds qui sont en décalage de phase de 180 degrés tandis que N se réfère au fil neutre. Par des connexions monophasées de transformateur, ces circuits secondaires peuvent être alimentés par une ou deux des phases primaires A, B et C. Pour des charges triphasées, utiliser les codes de phase A, B, C au lieu de s12N.

Le Tableau 104 montre tous les libellés de PhaseCode.

Tableau 104 – Libellés de Core::PhaseCode

libellé	description
ABCN	Phases A, B, C et N.
ABC	Phases A, B et C.
ABN	Phases A, B et neutre.
ACN	Phases A, C et neutre.
BCN	Phases B, C et neutre.
AB	Phases A et B.
AC	Phases A et C.
BC	Phases B et C.
AN	Phases A et neutre.
BN	Phases B et neutre.
CN	Phases C et neutre.
A	Phase A.
B	Phase B.
C	Phase C.
N	Phase neutre.
s1N	Phase secondaire 1 et neutre.
s2N	Phase secondaire 2 et neutre.
s12N	Phases secondaires 1, 2 et neutre.
s1	Phase secondaire 1.
s2	Phase secondaire 2.
s12	Phases secondaires 1 et 2.

6.4.26 PowerSystemResource

Une ressource du réseau électrique peut être un élément d'équipement tel qu'un organe de coupure, un conteneur d'équipements contenant beaucoup d'éléments d'équipements individuels tel qu'un poste ou une entité opérationnelle telle qu'une sous-zone de réglage. Les composants de réseau électrique peuvent comporter des mesures associées.

Le Tableau 105 montre tous les attributs de PowerSystemResource.

Tableau 105 – Attributs de Core::PowerSystemResource

nom	type	description
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 106 montre toutes les extrémités d'association de PowerSystemResource avec d'autres classes.

Tableau 106 – Extrémités d'association de Core::PowerSystemResource avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] PSRType	PSRType	Classification personnalisée pour ce composant de réseau électrique.
[0..1]	[0..*] Measurements	Measurement	Les mesurages associés avec ce composant de réseau électrique.
[1..1]	[0..*] OperatingShare	OperatingShare	Quotes-parts d'exploitation de ce composant de réseau électrique.
[0..*]	[0..*] PsrLists	PsrList	Listes de ceux auxquels appartient ce composant de réseau électrique.
[1..1]	[0..1] OutageSchedule	OutageSchedule	Un composant de réseau électrique peut avoir un programme d'indisponibilité.
[0..*]	[0..*] ReportingGroup	ReportingGroup	Groupes de reporting auxquels appartient ce composant de réseau électrique.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.4.27 PsrList

Liste arbitraire de composants de réseau électrique. Peut être utilisée à diverses fins, y compris le regroupement pour la génération de rapports.

Le Tableau 107 montre tous les attributs de PsrList.

Tableau 107 – Attributs de Core::PsrList

nom	type	description
typePSRList	String	Type de ressources de réseau électrique (Power System Ressource) contenues dans cette liste.
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 108 montre toutes les extrémités d'association de PsrList avec d'autres classes.

Tableau 108 – Extrémités d'association de Core::PsrList avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..*] PowerSystemResources	PowerSystemResource	Ressources de réseau électrique (Power System Ressource) contenues dans cette liste.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.4.28 PSRType

Instances de classification de la même classe, par exemple ACLineSegments (segments de ligne en courant alternatif) aériens et souterrains. Ce mécanisme de classification est destiné à procurer de la flexibilité au-delà du domaine de la présente norme, c'est-à-dire à apporter une personnalisation qui n'est pas dans la norme.

Le Tableau 109 montre tous les attributs de PSRType.

Tableau 109 – Attributs de Core::PSRType

nom	type	description
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 110 montre toutes les extrémités d'association de PSRType avec d'autres classes.

Tableau 110 – Extrémités d'association de Core::PSRType avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[0..*] PowerSystemResources	PowerSystemResource	Ressources de réseau électrique classées avec ce type de ressource de réseau électrique.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.4.29 RegularIntervalSchedule

Le programme comporte plusieurs instants entre lesquels le temps est constant.

Le Tableau 111 montre tous les attributs de RegularIntervalSchedule.

Tableau 111 – Attributs de Core::RegularIntervalSchedule

nom	type	description
endTime	DateTime	L'heure pour le dernier instant (Time Point).
timeStep	Seconds	La durée entre chaque paire d'instant réguliers ultérieurs, en ordre séquentiel.
startTime	DateTime	hérité de: BasicIntervalSchedule
value1Unit	UnitSymbol	hérité de: BasicIntervalSchedule
value1Multiplier	UnitMultiplier	hérité de: BasicIntervalSchedule
value2Unit	UnitSymbol	hérité de: BasicIntervalSchedule
value2Multiplier	UnitMultiplier	hérité de: BasicIntervalSchedule
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 112 montre toutes les extrémités d'association de RegularIntervalSchedule avec d'autres classes.

Tableau 112 – Extrémités d'association de Core::RegularIntervalSchedule avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[1..1]	[1..*] TimePoints	RegularTimePoint	Valeurs de données du point définissant ce programme.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.4.30 Classe racine RegularTimePoint

Instant pour un programme, l'intervalle de temps entre les instants consécutifs étant constant.

Le Tableau 113 montre tous les attributs de RegularTimePoint.

Tableau 113 – Attributs de Core::RegularTimePoint

nom	type	description
sequenceNumber	Integer	Position du RegularTimePoint dans cette séquence. Noter que les Time Point ne doivent pas nécessairement être séquentiels, c'est-à-dire que des Time Point peuvent être omis. La durée réelle pour un RegularTimePoint est calculée en multipliant RegularIntervalSchedule.timeStep associé par le numéro de sélection de RegularTimePoint.sequenceNumber et en ajoutant l'heure de départ des programmes associés.
value1	Float	La première valeur au temps indiqué. La signification de cette valeur est définie par le type dérivé du programme associé.
value2	Float	La deuxième valeur au temps indiqué. La signification de cette valeur est définie par le type dérivé du programme associé.

Le Tableau 114 montre toutes les extrémités d'association de RegularTimePoint avec d'autres classes.

Tableau 114 – Extrémités d'association de Core::RegularTimePoint avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[1..*]	[1..1] IntervalSchedule	RegularIntervalSchedule	Programme d'intervalle régulier contenant cet instant.

6.4.31 ReportingGroup

Un groupe de reporting est utilisé pour divers groupements ad hoc utilisés pour la production de rapports.

Le Tableau 115 montre tous les attributs de ReportingGroup.

Tableau 115 – Attributs de Core::ReportingGroup

nom	type	description
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 116 montre toutes les extrémités d'association de ReportingGroup avec d'autres classes.

Tableau 116 – Extrémités d'association de Core::ReportingGroup avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..*] PowerSystemResource	PowerSystemResource	Ressources de réseau électrique qui appartiennent au groupe de reporting.
[0..*]	[0..1] ReportingSuperGroup	ReportingSuperGroup	Supergroupe de reporting auquel ce groupe de reporting appartient.
[0..1]	[0..*] BusNameMarker	BusNameMarker	Les marqueurs de nom de barres qui appartiennent à ce groupe de reporting.
[0..1]	[0..*] TopologicalNode	TopologicalNode	Les nœuds topologiques qui appartiennent à ce groupe de reporting.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.4.32 ReportingSuperGroup

Un supergroupe de reporting regroupe des groupes de reporting pour un rapport de niveau plus haut.

Le Tableau 117 montre tous les attributs de ReportingSuperGroup.

Tableau 117 – Attributs de Core::ReportingSuperGroup

nom	type	description
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 118 montre toutes les extrémités d'association de ReportingSuperGroup avec d'autres classes.

Tableau 118 – Extrémités d'association de Core::ReportingSuperGroup avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[0..*] ReportingGroup	ReportingGroup	Groupes de reporting qui sont regroupés sous ce supergroupe.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.4.33 SubGeographicalRegion

Un sous-ensemble d'une région géographique d'un modèle de réseau de système de puissance.

Le Tableau 119 montre tous les attributs de SubGeographicalRegion.

Tableau 119 – Attributs de Core::SubGeographicalRegion

nom	type	description
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 120 montre toutes les extrémités d'association de SubGeographicalRegion avec d'autres classes.

Tableau 120 – Extrémités d'association de Core::SubGeographicalRegion avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[0..*] Substations	Substation	Les postes dans cette sous-région géographique.
[0..1]	[0..*] Lines	Line	Les lignes dans cette sous-région géographique.
[0..*]	[0..1] Region	GeographicalRegion	La région géographique dans laquelle se trouve cette sous-région géographique.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.4.34 Substation

Collection d'équipements dans d'autres buts que la production ou l'utilisation, à travers lesquels passe de l'énergie électrique en vrac dans le but de commuter ou de modifier ses caractéristiques.

Le Tableau 121 montre tous les attributs de Substation.

Tableau 121 – Attributs de Core::Substation

nom	type	description
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 122 montre toutes les extrémités d'association de Substation avec d'autres classes.

Tableau 122 – Extrémités d'association de Core::Substation avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[1..1]	[0..*] VoltageLevels	VoltageLevel	Les niveaux de tension à l'intérieur de ce poste.
[0..1]	[0..*] Bays	Bay	Cellules contenues dans ce poste.
[0..*]	[0..1] Region	SubGeographicalRegion	La sous-région géographique où se trouve ce poste.
[0..1]	[0..*] Equipments	Equipement	hérité de: EquipmentContainer
[1..1]	[0..*] ConnectivityNodes	ConnectivityNode	hérité de: ConnectivityNodeContainer
[0..1]	[0..*] TopologicalNode	TopologicalNode	hérité de: ConnectivityNodeContainer
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.4.35 Terminal

Point de connexion électrique à une partie d'un équipement conducteur. Les Terminals (bornes) sont connectées sur des points de connexion physiques appelés nœuds de connectivité (ConnectivityNode).

Le Tableau 123 montre tous les attributs de Terminal.

Tableau 123 – Attributs de Core::Terminal

nom	type	description
connected	Boolean	Le statut "connected" (connecté) est relatif à un modèle de topologie nodale et à la relation nœud topologique-borne. "True" (vrai) implique que la borne est connectée au nœud topologique connexe et "false" (faux) implique qu'elle ne l'est pas. Dans un modèle de topologie nodale, le statut "connected" (connecté) est utilisé pour indiquer si l'équipement est déconnecté sans devoir changer la connectivité décrite par la relation nœud topologique-borne. Un cas valide est que l'équipement conducteur peut être connecté à une extrémité et ouvert à l'autre. C'est un cas approprié, en particulier pour un segment de ligne de CA où la charge de ligne réactive peut être significative.
phases	PhaseCode	Représente la condition de phase normale du réseau.
sequenceNumber	Integer	L'orientation des connexions de bornes pour un équipement conducteur à plusieurs bornes. La numérotation séquentielle commence à 1 et il convient que les bornes supplémentaires suivent l'ordre croissant. La première borne est le "point de départ" d'une branche à deux bornes.
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 124 montre toutes les extrémités d'association de Terminal avec d'autres classes.

Tableau 124 – Extrémités d'association de Core::Terminal avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[1..1] ConductingEquipment	ConductingEquipment	L'équipement conducteur de la borne. L'équipement conducteur a des bornes qui peuvent être connectées à d'autres bornes d'un équipement conducteur à l'aide de nœuds de connectivité ou de nœuds topologiques.
[0..*]	[0..1] ConnectivityNode	ConnectivityNode	Le nœud de connectivité auquel cette borne est connectée à impédance zéro.
[0..1]	[0..*] RegulatingControl	RegulatingControl	Les commandes qui régulent cette borne.
[0..*]	[0..1] TopologicalNode	TopologicalNode	Le nœud topologique associé à la borne. Il peut être utilisé comme remplacement d'un chemin de nœuds de connectivité allant jusqu'au nœud topologique, éliminant donc la nécessité de modéliser les nœuds de connectivité dans certains cas. Remarque que si les nœuds de connectivité sont présents dans le modèle, cette association ne serait probablement pas utilisée comme spécification d'entrée.
[1..*]	[0..1] BusNameMarker	BusNameMarker	Le marqueur de nom du bus utilisé pour nommer le bus (nœud topologique).
[1..1]	[0..*] BranchGroupTerminal	BranchGroupTerminal	Les bornes orientées de groupe de branches pour lesquelles cette borne est surveillée.

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[0..*] Measurements	Measurement	Les mesures associées à cette borne définissent l'endroit où la mesure est placée dans la topologie de réseau. On peut par exemple les utiliser pour représenter la position d'un capteur, comme un transformateur de tension (PT: Voltage transformer) sur un jeu de barres ou un transformateur de courant (CT: Current transformer) sur une barre entre un disjoncteur et un sectionneur.
[1..1]	[0..2] TieFlow	TieFlow	Les flux d'interconnexion de zone de commande auxquels cette borne est associée.
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	Les ensembles de limites opérationnelles qui s'appliquent spécifiquement à cette borne. D'autres ensembles de limites opérationnelles peuvent s'appliquer à cette borne par le truchement d'une association à l'équipement.
[1..1]	[0..*] HasSecond_MutualCoupling	MutualCoupling	Couplages mutuels avec la branche associée comme première branche.
[1..1]	[0..*] HasFirst_MutualCoupling	MutualCoupling	Couplages mutuels associés avec la branche comme première branche.
[1..1]	[0..1] SvPowerFlow	SvPowerFlow	La variable d'état de flux de puissance associée à la borne.
[1..1]	[0..*] AuxiliaryEquipment	AuxiliaryEquipment	L'équipement auxiliaire connecté à la borne.
[0..1]	[0..*] TransformerEnd	TransformerEnd	Toutes les extrémités de transformateur connectées à cette borne.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.4.36 VoltageLevel

Ensemble d'équipements à un même niveau de tension, constituant un ensemble de commutation. L'équipement consiste typiquement en disjoncteurs, jeux de barre, dispositifs d'instrumentation, de contrôle, de régulation et de protection, de même que des regroupements de ce type de constituants.

Le Tableau 125 montre tous les attributs de VoltageLevel.

Tableau 125 – Attributs de Core::VoltageLevel

nom	type	description
highVoltageLimit	Voltage	La tension limite haute du jeu de barres
lowVoltageLimit	Voltage	La tension limite basse du jeu de barres
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 126 montre toutes les extrémités d'association de VoltageLevel avec d'autres classes.

Tableau 126 – Extrémités d'association de Core::VoltageLevel avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[1..1] BaseVoltage	BaseVoltage	La tension de base utilisée pour tout l'équipement dans le niveau de tension.
[0..1]	[0..*] Bays	Bay	Cellules qui se trouvent à ce niveau de tension.
[0..*]	[1..1] Substation	Substation	Le poste du niveau de tension.
[0..1]	[0..*] Equipments	Equipement	hérité de: EquipmentContainer
[1..1]	[0..*] ConnectivityNodes	ConnectivityNode	hérité de: ConnectivityNodeContainer
[0..1]	[0..*] TopologicalNode	TopologicalNode	hérité de: ConnectivityNodeContainer
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.5 Package DiagramLayout

6.5.1 Généralités

Ce paquetage décrit la présentation du diagramme. Il décrit la manière dont les objets sont disposés dans un système de coordonnées plutôt que la manière dont ils sont rendus.

La Figure 39 montre le diagramme de classe DiagramLayout.

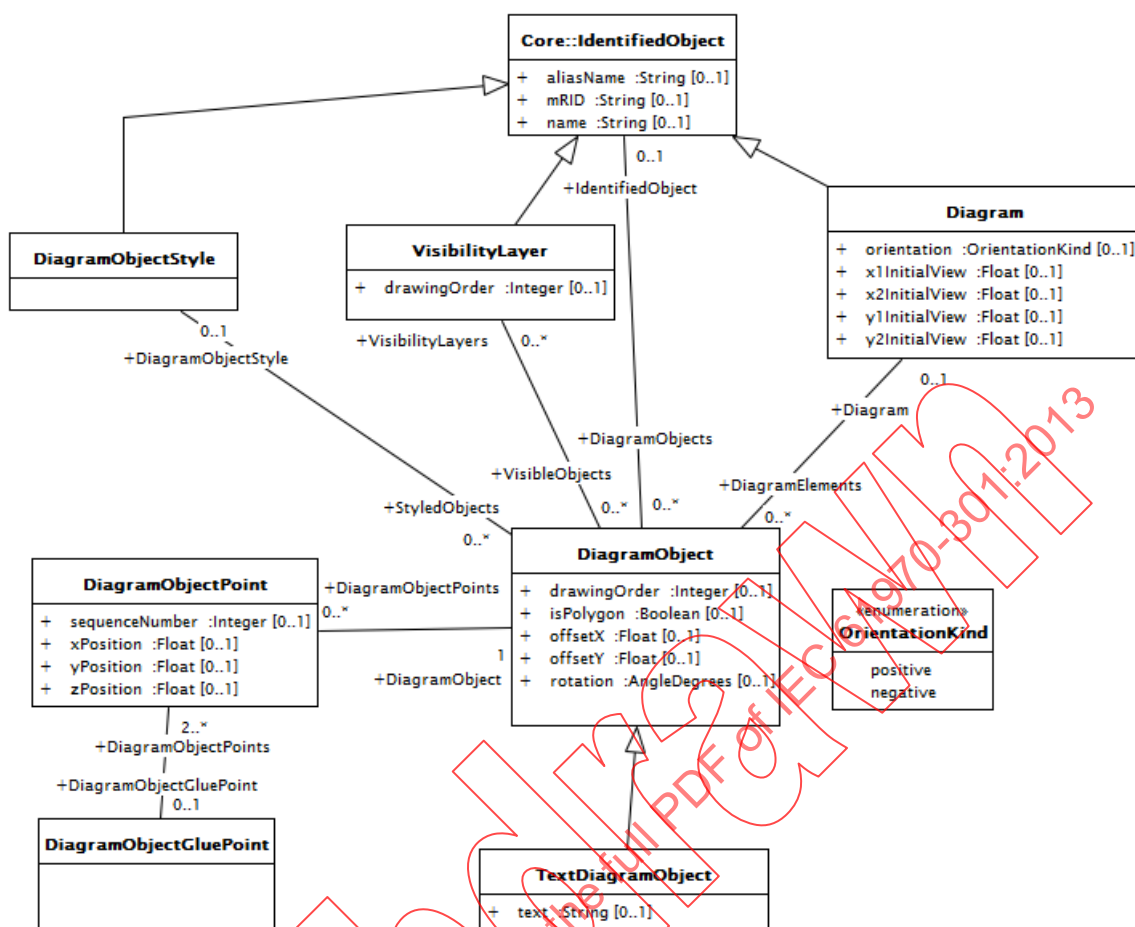


Figure 39 – Diagramme de classe DiagramLayout::DiagramLayout

Le diagramme montre le modèle de présentation du diagramme.

6.5.2 Diagram (diagramme)

Le diagramme échange. Le système de coordonnées est un système standard de coordonnées cartésiennes et l'attribut Orientation définit l'orientation.

Le Tableau 127 montre tous les attributs de Diagram.

Tableau 127 – Attributs de DiagramLayout::Diagram

nom	type	description
orientation	OrientationKind	Orientation du système de coordonnées pour le diagramme.
x1InitialView	Float	Coordonnées X du premier coin de la vue initiale.
x2InitialView	Float	Coordonnées X du deuxième coin de la vue initiale.
y1InitialView	Float	Coordonnées Y du premier coin de la vue initiale.
y2InitialView	Float	Coordonnées Y du deuxième coin de la vue initiale.
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 128 montre toutes les extrémités d'association de Diagram avec d'autres classes.

Tableau 128 – Extrémités d'association de DiagramLayout::Diagram avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[0..*] DiagramElements	DiagramObject	Un diagramme est constitué de plusieurs objets de diagramme.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.5.3 DiagramObject

Un objet qui définit un ou plusieurs points dans un espace donné. Cet objet peut être associé à tout ce qui spécialise IdentifiedObject. Pour des schémas unifilaires, ces objets comprennent typiquement des éléments comme des valeurs analogiques, des disjoncteurs, des sectionneurs, des transformateurs de puissance et des lignes de transport.

Le Tableau 129 montre tous les attributs de DiagramObject.

Tableau 129 – Attributs de DiagramLayout::DiagramObject

nom	type	description
drawingOrder	Integer	L'ordre de dessin dans cet élément. Plus le nombre est élevé, plus l'élément est dessiné tard dans la séquence. Il est utilisé pour s'assurer que les éléments qui se chevauchent sont rendus dans le bon ordre.
isPolygon	Boolean	Définit si les points des objets du diagramme définissent ou non les limites d'un polygone ou le tracé d'une polyligne. Si cette valeur est vraie, il est recommandé que l'application de réception prenne en compte le premier et le dernier point à connecter.
offsetX	Float	Le décalage en direction X. Il sert à définir le décalage depuis le centre pour le rendu d'une icône (par défaut un point unique spécifie le centre de l'icône). Le décalage est exprimé par unité, 0 indique qu'il n'y a aucun décalage à partir du centre horizontal de l'icône. -0,5 indique que le décalage est de 50 % vers la gauche et 0,5 indique un décalage de 50 % vers la droite.
offsetY	Float	Le décalage en direction Y. Il sert à définir le décalage depuis le centre pour le rendu d'une icône (par défaut un point unique spécifie le centre de l'icône). Le décalage est exprimé par unité, 0 indique qu'il n'y a aucun décalage à partir du centre vertical de l'icône. La direction du décalage dépend de l'orientation du diagramme, -0,5 et 0,5 indiquent un décalage de +/- 50 % sur l'axe vertical.
rotation	AngleDegrees	Définit l'angle de rotation de l'objet de diagramme dans le sens des aiguilles d'une montre à partir de la normale
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 130 montre toutes les extrémités d'association de DiagramObject avec d'autres classes.

Tableau 130 – Extrémités d'association de DiagramLayout::DiagramObject avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] Diagram	Diagram	Un objet de diagramme fait partie d'un diagramme.
[0..*]	[0..*] VisibilityLayers	VisibilityLayer	Un objet de diagramme peut faire partie de plusieurs couches de visibilité.
[1..1]	[0..*] DiagramObjectPoints	DiagramObjectPoint	Un objet de diagramme peut avoir 0 ou plusieurs points pour refléter la position de sa présentation, son tracé (pour des polygones) ou sa limite (pour des polygones).
[0..*]	[0..1] DiagramObjectStyle	DiagramObjectStyle	Un objet de diagramme a un style associé qui fournit une référence pour le style utilisé dans le système d'origine.
[0..*]	[0..1] IdentifiedObject	IdentifiedObject	L'objet de domaine auquel est associé cet objet de diagramme.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.5.4 Classe racine DiagramObjectGluePoint

Celle-ci est utilisée pour regrouper des points d'un objet de diagramme à partir de différents objets de diagramme qui sont considérés comme indissociables l'un de l'autre ("collés") dans un diagramme, même s'ils ne se trouvent pas exactement aux mêmes coordonnées.

Le Tableau 131 montre toutes les extrémités d'association de DiagramObjectGluePoint avec d'autres classes.

Tableau 131 – Extrémités d'association de DiagramLayout::DiagramObjectGluePoint avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[2..*] DiagramObjectPoints	DiagramObjectPoint	Un point de collage d'un objet de diagramme est associé à 2 points d'objet ou davantage qui sont considérés comme collés l'un à l'autre.

6.5.5 Classe racine DiagramObjectPoint

Un point dans un espace donné défini par 3 coordonnées et associé à un objet de diagramme. Les coordonnées peuvent être positives ou négatives puisque l'origine ne se trouve pas nécessairement dans le coin d'un diagramme.

Le Tableau 132 montre tous les attributs de DiagramObjectPoint.

Tableau 132 – Attributs de DiagramLayout::DiagramObjectPoint

nom	type	description
sequenceNumber	Integer	La position séquentielle du point, utilisée pour définir l'ordre des points pour des objets de diagramme qui forment une polygône ou un polygone ayant plus d'un point.
xPosition	Float	La coordonnée X de ce point.
yPosition	Float	La coordonnée Y de ce point.
zPosition	Float	La coordonnée Z de ce point.

Le Tableau 133 montre toutes les extrémités d'association de DiagramObjectPoint avec d'autres classes.

Tableau 133 – Extrémités d'association de DiagramLayout::DiagramObjectPoint avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[2..*]	[0..1] DiagramObjectGluePoint	DiagramObjectGluePoint	Le point de collage auquel est associé ce point.
[0..*]	[1..1] DiagramObject	DiagramObject	L'objet de diagramme auquel sont associés les points.

6.5.6 DiagramObjectStyle

Une référence à un style utilisé par le système d'origine pour un objet de diagramme. Un style d'objet de diagramme décrit des informations comme l'épaisseur de la ligne, la forme comme un cercle ou un rectangle, etc., et la couleur.

Le Tableau 134 montre tous les attributs de DiagramObjectStyle.

Tableau 134 – Attributs de DiagramLayout::DiagramObjectStyle

nom	type	description
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 135 montre toutes les extrémités d'association de DiagramObjectStyle avec d'autres classes.

Tableau 135 – Extrémités d'association de DiagramLayout::DiagramObjectStyle avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[0..*] StyledObjects	DiagramObject	Un style peut être attribué à plusieurs objets de diagramme.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.5.7 Enumération OrientationKind

L'orientation du système de coordonnées par rapport au sommet, à la gauche et au système de numérotation des coordonnées.

Le Tableau 136 montre tous les libellés de OrientationKind.

Tableau 136 – Libellés de DiagramLayout::OrientationKind

libellé	description
positive	Pour des diagrammes en 2D, une orientation positive implique que les valeurs X croissent de la gauche vers la droite et les valeurs Y croissent du bas vers le haut. On l'appelle aussi orientation à droite.
negative	Pour des diagrammes en 2D, une orientation négative donne une orientation à gauche (tel que se présente généralement l'affichage graphique sur ordinateur), les valeurs X croissent de la gauche vers la droite et les valeurs Y croissent du haut vers le bas. On l'appelle aussi orientation à gauche.

6.5.8 TextDiagramObject

Un objet de diagramme pour placer un texte libre ou un texte dérivé d'un objet de domaine associé.

Le Tableau 137 montre tous les attributs de TextDiagramObject.

Tableau 137 – Attributs de DiagramLayout::TextDiagramObject

nom	type	description
text	String	Le texte qui est affiché par cet objet de diagramme de texte.
drawingOrder	Integer	hérité de: DiagramObject
isPolygon	Boolean	hérité de: DiagramObject
offsetX	Float	hérité de: DiagramObject
offsetY	Float	hérité de: DiagramObject
rotation	AngleDegrees	hérité de: DiagramObject
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 138 montre toutes les extrémités d'association de TextDiagramObject avec d'autres classes.

Tableau 138 – Extrémités d'association de DiagramLayout::TextDiagramObject avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] Diagram	Diagram	hérité de: DiagramObject
[0..*]	[0..*] VisibilityLayers	VisibilityLayer	hérité de: DiagramObject
[1..1]	[0..*] DiagramObjectPoints	DiagramObjectPoint	hérité de: DiagramObject
[0..*]	[0..1] DiagramObjectStyle	DiagramObjectStyle	hérité de: DiagramObject
[0..*]	[0..1] IdentifiedObject	IdentifiedObject	hérité de: DiagramObject
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.5.9 VisibilityLayer

Les couches sont typiquement utilisées pour regrouper des objets de diagramme d'après des thèmes et des échelles. Les thèmes sont utilisés pour afficher ou masquer certaines informations (par exemple des lacs, des frontières) tandis que les échelles sont utilisées pour masquer ou afficher des informations qui dépendent du niveau réel de zoom (masquer le texte quand il est trop petit pour être lisible ou quand il dépasse la taille de l'écran). On l'appelle aussi mise au net.

L'échange de graphique basé sur le CIM prend en charge une relation m:n entre les objets de diagramme et les couches. Il est du ressort du système d'importation de convertir un cas m:n en représentation 1:n appropriée si le système d'importation ne prend pas en charge m:n.

Le Tableau 139 montre tous les attributs de VisibilityLayer.

Tableau 139 – Attributs de DiagramLayout::VisibilityLayer

nom	type	description
drawingOrder	Integer	L'ordre de dessin pour cette couche. Plus le nombre est grand, plus la couche et les objets qui s'y trouvent sont rendus tard.
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le

Tableau 140 montre toutes les extrémités d'association de VisibilityLayer avec d'autres classes.

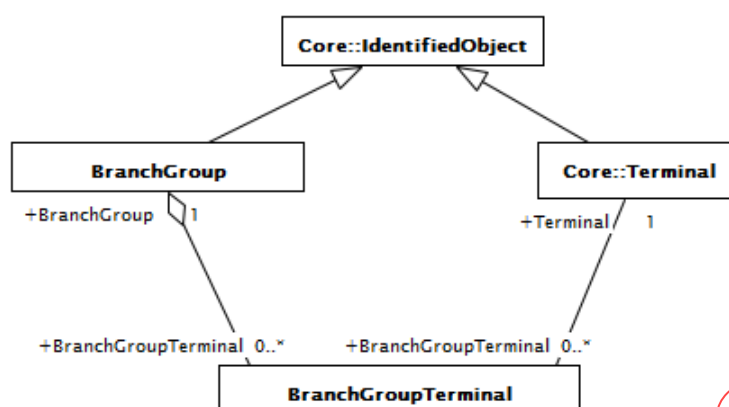


Figure 41 – Diagramme de classe OperationalLimits::BranchGroup

Ce diagramme montre le modèle pour contrôler des flux combinés sur des groupes de branches.

6.6.2 ActivePowerLimit

Limite sur le flux de puissance active.

Le Tableau 141 montre tous les attributs de ActivePowerLimit.

Tableau 141 – Attributs de OperationalLimits::ActivePowerLimit

nom	type	description
value	ActivePower	Valeur de la limite de puissance active.
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 142 montre toutes les extrémités d'association de ActivePowerLimit avec d'autres classes.

Tableau 142 – Extrémités d'association de OperationalLimits::ActivePowerLimit avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[1..1] OperationalLimitSet	OperationalLimitSet	hérité de: OperationalLimit
[0..*]	[0..1] OperationalLimitType	OperationalLimitType	hérité de: OperationalLimit
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.6.3 ApparentPowerLimit

Limite de puissance apparente.

Le Tableau 143 montre tous les attributs de ApparentPowerLimit.

Tableau 143 – Attributs de OperationalLimits::ApparentPowerLimit

nom	type	description
valeur	ApparentPower	La limite de puissance apparente.
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 144 montre toutes les extrémités d'association de ApparentPowerLimit avec d'autres classes.

Tableau 144 – Extrémités d'association de OperationalLimits::ApparentPowerLimit avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[1..1] OperationalLimitSet	OperationalLimitSet	hérité de: OperationalLimit
[0..*]	[0..1] OperationalLimitType	OperationalLimitType	hérité de: OperationalLimit
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.6.4 BranchGroup

Un groupe de bornes de branches dont la sommation des flux orientés doit être surveillée. Un groupe de branches peut ne pas former un ensemble de coupure du réseau.

Le Tableau 145 montre tous les attributs de BranchGroup.

Tableau 145 – Attributs de OperationalLimits::BranchGroup

nom	type	description
maximumActivePower	ActivePower	Le flux maximal de puissance active.
maximumReactivePower	ReactivePower	Le flux maximal de puissance réactive.
minimumActivePower	ActivePower	Le flux minimal de puissance active.
minimumReactivePower	ReactivePower	Le flux minimal de puissance réactive.
monitorActivePower	Boolean	Surveiller le flux de puissance active.
monitorReactivePower	Boolean	Surveiller le flux de puissance réactive.
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 146 montre toutes les extrémités d'association de BranchGroup avec d'autres classes.

Tableau 146 – Extrémités d'association de OperationalLimits::BranchGroup avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[1..1]	[0..*] BranchGroupTerminal	BranchGroupTerminal	Les bornes de BranchGroups orientées devant être sommées.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.6.5 Classe racine BranchGroupTerminal

Un flux spécifique de bornes orienté pour un groupe de branches.

Le Tableau 147 montre tous les attributs de BranchGroupTerminal.

Tableau 147 – Attributs de OperationalLimits::BranchGroupTerminal

nom	type	description
positiveFlowIn	Boolean	S'il est mis à true (c'est-à-dire vrai), le flux pénétrant dans la borne est sommé. S'il est mis à false (c'est-à-dire faux), le flux sortant de la borne est sommé.

Le Tableau 148 montre toutes les extrémités d'association de BranchGroupTerminal avec d'autres classes.

Tableau 148 – Extrémités d'association de OperationalLimits::BranchGroupTerminal avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[1..1] Terminal	Terminal	La borne devant être sommée.
[0..*]	[1..1] BranchGroup	BranchGroup	Le BranchGroup auquel appartiennent les bornes de groupe de branches orientées.

6.6.6 CurrentLimit

Limite opérationnelle sur le courant.

Le Tableau 149 montre tous les attributs de CurrentLimit.

Tableau 149 – Attributs de OperationalLimits::CurrentLimit

nom	type	description
value	CurrentFlow	Limite sur le flux de courant.
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 150 montre toutes les extrémités d'association de CurrentLimit avec d'autres classes.

Tableau 150 – Extrémités d'association de OperationalLimits::CurrentLimit avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[1..1] OperationalLimitSet	OperationalLimitSet	hérité de: OperationalLimit
[0..*]	[0..1] OperationalLimitType	OperationalLimitType	hérité de: OperationalLimit
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.6.7 OperationalLimit

Une valeur associée à une sorte spécifique de limite.

Le Tableau 151 montre tous les attributs de OperationalLimit.

Tableau 151 – Attributs de OperationalLimits::OperationalLimit

nom	type	description
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 152 montre toutes les extrémités d'association de OperationalLimit avec d'autres classes.

Tableau 152 – Extrémités d'association de OperationalLimits::OperationalLimit avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[1..1] OperationalLimitSet	OperationalLimitSet	Jeu de limites auquel les valeurs limites appartiennent.
[0..*]	[0..1] OperationalLimitType	OperationalLimitType	Le type de limite associé à cette limite.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.6.8 Enumération OperationalLimitDirectionKind

Le sens d'une limite opérationnelle.

Le Tableau 153 montre tous les libellés de OperationalLimitDirectionKind.

Tableau 153 – Libellés de OperationalLimits::OperationalLimitDirectionKind

libellé	description
high	La limite est une limite haute. Si elle est appliquée au flux d'une borne, le sens positif est celui pénétrant dans la borne.
low	La limite est une limite basse. Si elle est appliquée au flux d'une borne, le sens positif est celui pénétrant dans la borne.
absoluteValue	Si la valeur absolue de la valeur surveillée est supérieure à la valeur limite, la limite est violée. En fait, la limite est à la fois une limite haute et sa valeur négative une limite basse.

6.6.9 OperationalLimitSet

Jeu de limites associées à un équipement. Les jeux de limites pourraient s'appliquer à une température spécifique, ou à une saison par exemple. Un jeu de limites peut contenir des sévérités différentes pour les niveaux limites qui s'appliqueraient au même équipement. Le jeu peut contenir des limites de types différents telles que limites de puissance apparente et limites de courant ou bien limites de tension haute et basse qui sont logiquement appliquées ensemble comme un jeu.

Le Tableau 154 montre tous les attributs de OperationalLimitSet.

Tableau 154 – Attributs de OperationalLimits::OperationalLimitSet

nom	type	description
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 155 montre toutes les extrémités d'association de OperationalLimitSet avec d'autres classes.

Tableau 155 – Extrémités d'association de OperationalLimits::OperationalLimitSet avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] Equipment	Equipment	L'équipement auquel le jeu de limites s'applique.
[0..*]	[0..1] Terminal	Terminal	La borne spécifiquement associée à ce jeu de limites opérationnelles. Si aucune borne n'est associée, toutes les bornes de l'équipement sont implicites.
[1..1]	[0..*] OperationalLimitValue	OperationalLimit	Valeurs des limites opérationnelles.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.6.10 OperationalLimitType

La signification opérationnelle d'une catégorie de limites.

Le Tableau 156 montre tous les attributs de OperationalLimitType.

Tableau 156 – Attributs de OperationalLimits::OperationalLimitType

nom	type	description
acceptableDuration	Seconds	La durée acceptable nominale de la limite. Les limites sont communément exprimées en termes de limites temporelles pour lesquelles la limite est normalement acceptable. La durée acceptable réelle d'une limite spécifique dépend d'autres facteurs locaux tels que la température ou la vitesse du vent.
direction	OperationalLimitDirection Kind	Le sens de la limite.
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 157 montre toutes les extrémités d'association de OperationalLimitType avec d'autres classes.

Tableau 157 – Extrémités d'association de OperationalLimits::OperationalLimitType avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[0..*] OperationalLimit	OperationalLimit	Les limites opérationnelles associées à ce type de limite.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.6.11 VoltageLimit

La limite opérationnelle appliquée à la tension.

Le Tableau 158 montre tous les attributs de VoltageLimit.

Tableau 158 – Attributs de OperationalLimits::VoltageLimit

nom	type	description
value	Voltage	Limite sur la tension. La nature basse ou haute de la limite dépend des propriétés du type de limite opérationnelle.
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 159 montre toutes les extrémités d'association de VoltageLimit avec d'autres classes.

Tableau 159 – Extrémités d'association de OperationalLimits::VoltageLimit avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[1..1] OperationalLimitSet	OperationalLimitSet	hérité de: OperationalLimit
[0..*]	[0..1] OperationalLimitType	OperationalLimitType	hérité de: OperationalLimit
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.7 Paquetage Topology

6.7.1 Généralités

Extension du paquetage Core qui, associé à la classe Terminal (c'est-à-dire borne), modélise la Connectivity (connectivité), à savoir la définition physique de la façon dont les équipements sont connectés ensemble. De plus, il modélise la Topology, à savoir la définition logique de la façon dont un équipement est connecté via des organes de coupure fermés. La définition de Topology est indépendante des autres caractéristiques électriques.

La Figure 42 montre le diagramme de classe TopologicalNodeTerminal.

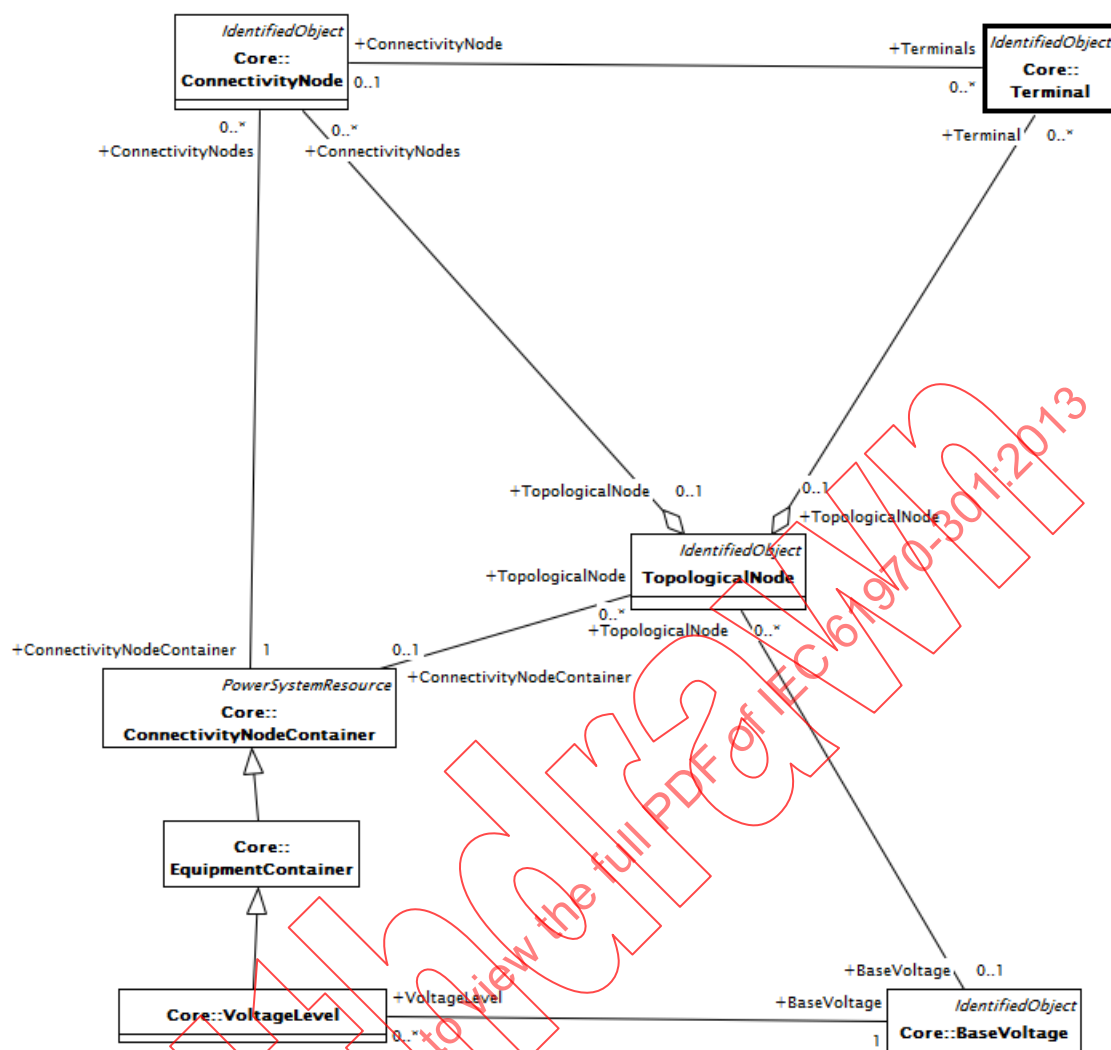


Figure 42 – Diagramme de classe Topology::TopologicalNodeTerminal

Montre les relations entre les nœuds topologiques et les bornes.

La Figure 43 montre le diagramme de classe TopologyMeasRelations.

Figure 44 – Diagramme de classe Topology::TopologyReporting

Montre les relations de reporting de topologie.

La Figure 45 montre le diagramme de classe Main.

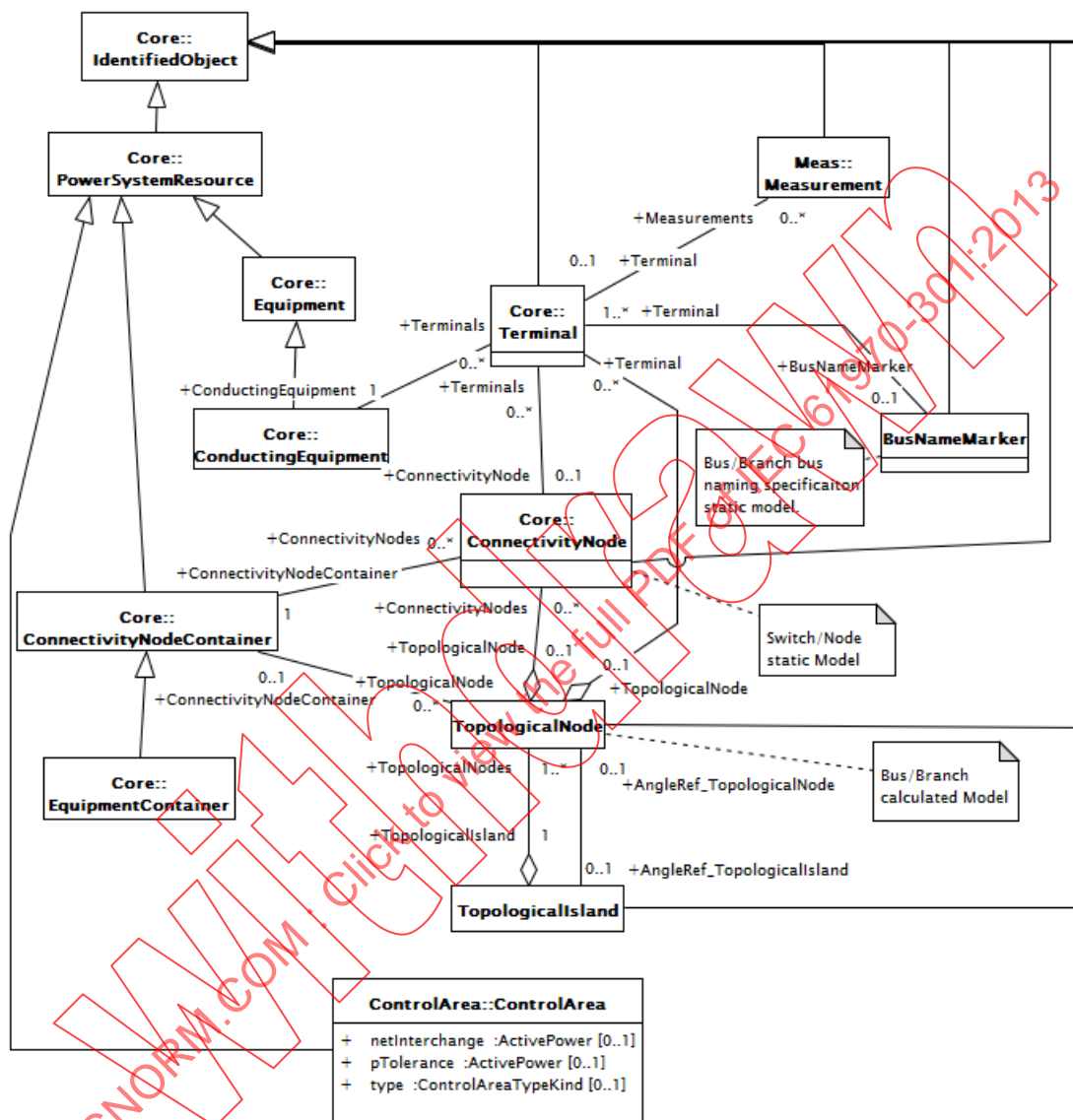


Figure 45 – Diagramme de classe Topology::Main

Ce diagramme montre toutes les classes incluses dans le paquetage Topology de même que les classes-clés externes, qui ont des associations avec des classes Topology.

6.7.2 BusNameMarker

Utilisé pour appliquer les noms normalisés d'utilisateur à des bus de topologie. Typiquement utilisé pour la génération de cas "topologie nodale". Associé à un ou plusieurs **ConnectivityNode** qui sont normalement une partie du nom du bus. Les **ConnectivityNode** associés doivent être reliés par des commutateurs sans retenue. Pour une configuration de poste en réseau bouclé, tous les nœuds de connectivité de jeux de barres dans l'anneau sont typiquement associés. Pour un disjoncteur et un demi-schéma, les deux jeux de barres seraient associés. Pour un réseau bouclé, tous les jeux de barres seraient associés. Pour une

configuration "rectiligne" de jeux de barres, seul le nœud de connectivité au niveau du jeu de barres serait associé.

Le Tableau 160 montre tous les attributs de BusNameMarker.

Tableau 160 – Attributs de Topology::BusNameMarker

nom	type	description
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 161 montre toutes les extrémités d'association de BusNameMarker avec d'autres classes.

Tableau 161 – Extrémités d'association de Topology::BusNameMarker avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] ReportingGroup	ReportingGroup	Le groupe de reporting auquel appartient ce marqueurs de nom de bus.
[0..1]	[1..*] Terminal	Terminal	Les bornes associées à ce marqueur de nom de bus.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.7.3 TopologicalIsland

Sous-ensemble du réseau électriquement connecté. Les îles topologiques peuvent changer lorsque l'état courant du réseau change, par exemple parce que

- la déconnexion de commutateurs ou de disjoncteurs change l'état dans un SCADA/EMS
- il y a création manuelle, changement ou suppression de nœuds topologiques dans un outil de planification.

Le Tableau 162 montre tous les attributs de TopologicalIsland.

Tableau 162 – Attributs de Topology::TopologicalIsland

nom	type	description
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 163 montre toutes les extrémités d'association de TopologicalIsland avec d'autres classes.

Tableau 163 – Extrémités d'association de Topology::TopologicalIsland avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[0..1] AngleRef_TopologicalNode	TopologicalNode	La référence angulaire de l'îlot. Normalement, il y a un TopologicalNode qui est sélectionné comme référence angulaire pour chaque îlot. D'autres schémas de référence existent et, donc, l'association est typiquement facultative.
[1..1]	[1..*] TopologicalNodes	TopologicalNode	Un TopologicalNode (nœud topologique) appartient à unTopologicalIsland (îlot) topologique.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.7.4 TopologicalNode

Pour un modèle de poste détaillé, un nœud topologique est un ensemble de nœuds de connectivité qui, dans l'état courant du réseau, sont reliés ensemble par n'importe quelle sorte de commutateurs fermés, y compris des cavaliers. Les nœuds topologiques changent en même temps que l'état du réseau change (c'est-à-dire que des organes de coupure, des disjoncteurs, etc., changent l'état).

Pour un modèle de planification, les états d'organe de coupure ne sont pas utilisés pour former des nœuds topologiques. Au contraire, ils sont créés ou supprimés manuellement dans un outil constructeur de modèle. Les nœuds topologiques maintenus de cette manière sont également appelés "bus".

Le Tableau 164 montre tous les attributs de TopologicalNode.

Tableau 164 – Attributs de Topology::TopologicalNode

nom	type	description
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 165 montre toutes les extrémités d'association de TopologicalNode avec d'autres classes.

Tableau 165 – Extrémités d'association de Topology::TopologicalNode avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] BaseVoltage	BaseVoltage	La tension de base du nœud topologique.
[0..*]	[0..1] ConnectivityNodeContainer	ConnectivityNodeContainer	Le conteneur de nœuds de connectivité auquel le nœud topologique appartient.
[0..*]	[0..1] ReportingGroup	ReportingGroup	Le groupe de reporting auquel le nœud topologique appartient.
[1..*]	[1..1] TopologicalIsland	TopologicalIsland	Un TopologicalNode (nœud topologique) appartient à un TopologicalIsland (îlot) topologique.
[0..1]	[0..*] ConnectivityNodes	ConnectivityNode	Les nœuds de connectivité se combinent les uns aux autres pour former ce nœud topologique. Peut dépendre de l'état courant des commutateurs dans le réseau.
[0..1]	[0..*] Terminal	Terminal	Les bornes associées au nœud topologique. Il peut être utilisé à la place d'un chemin de nœuds de connectivité allant jusqu'à la borne, éliminant donc la nécessité de modéliser les nœuds de connectivité dans certains cas. Remarque que si les nœuds de connectivité sont présents dans le modèle, cette association ne serait probablement pas utilisée en tant que spécification d'entrée.
[0..1]	[0..1] AngleRef_TopologicalIsland	TopologicalIsland	L'îlot pour lequel le nœud est une référence angulaire. Normalement, il y a un nœud référence angulaire pour chaque îlot.
[1..1]	[0..1] SvVoltage	SvVoltage	L'état de tension associé au nœud topologique.
[1..1]	[0..1] SvInjection	SvInjection	Les variables d'état du flux d'injection associées au nœud topologique.
[1..1]	[0..1] SvShortCircuit	SvShortCircuit	Les variables d'état pour court-circuit associées au nœud topologique.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8 Paquetage Wires

6.8.1 Généralités

Extension des paquetages Core et Topology qui modélise les informations des caractéristiques électriques des réseaux Transport et Distribution. Ce paquetage est utilisé par des applications de réseau telles que State Estimation, Load Flow et Optimal Power Flow.

La Figure 46 montre le diagramme de classe CutsAndJumpers.

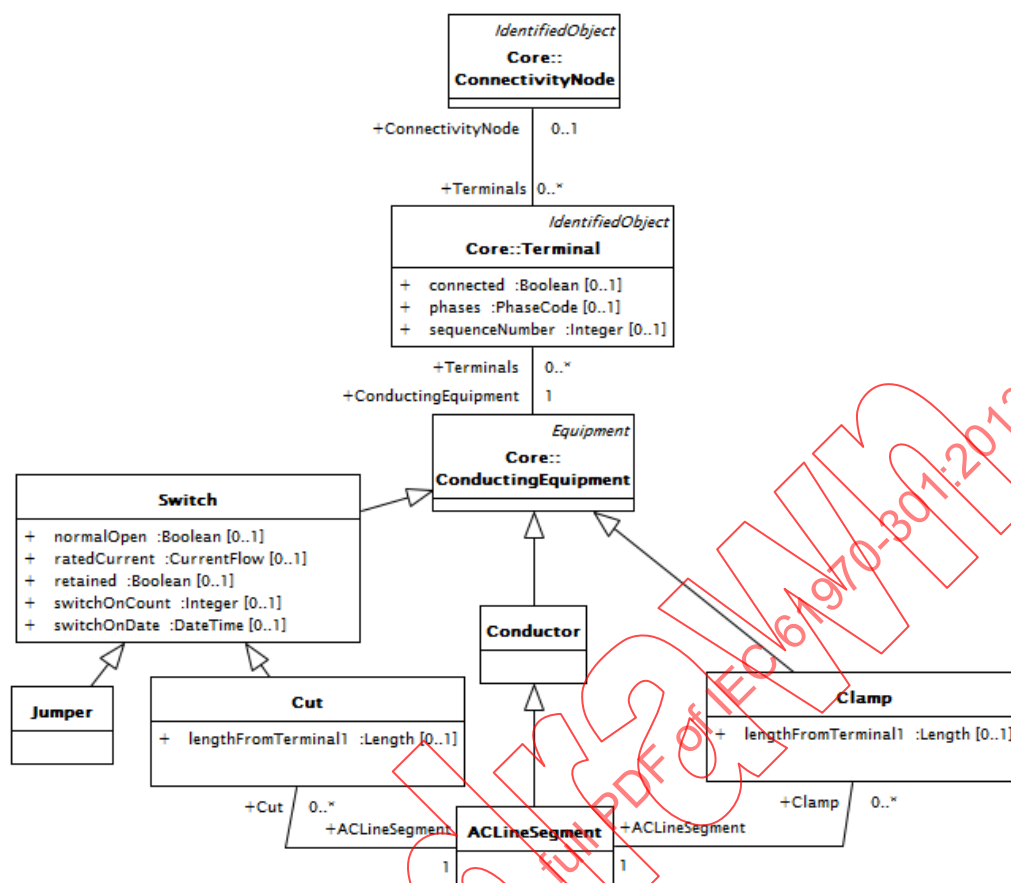


Figure 46 – Diagramme de classe Wires::CutsAndJumpers

Le diagramme montre le modèle de données des cuts et des cavaliers.

La Figure 47 montre le diagramme de classe MutualCoupling.

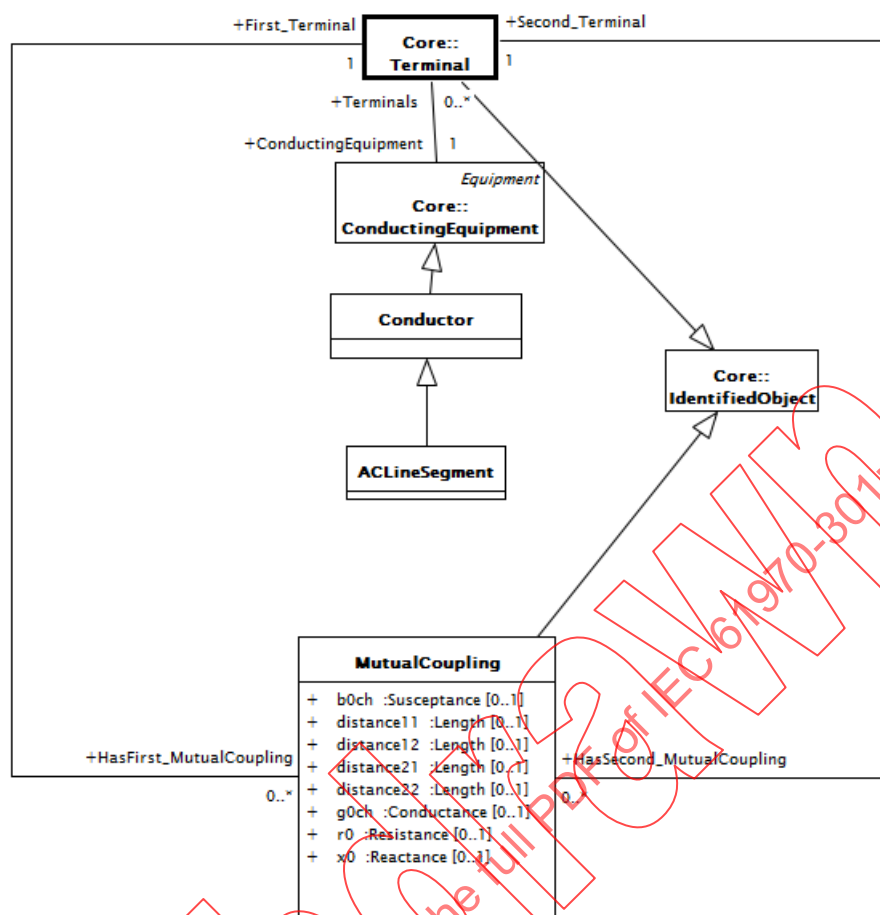


Figure 47 – Diagramme de classe Wires::MutualCoupling

Montre les modèles de couplage mutuel.

La Figure 48 montre le diagramme de classe Schedules.

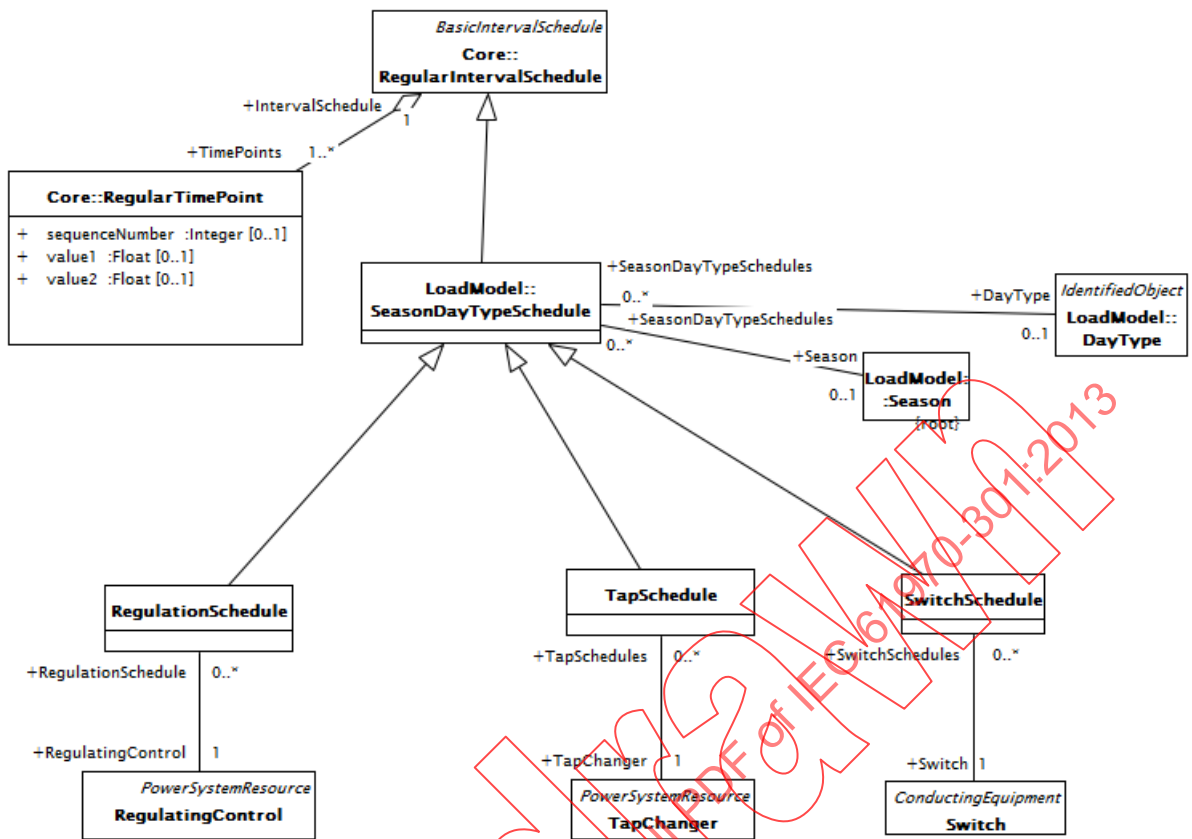


Figure 48 – Diagramme de classe Wires::Schedules

Montre les programmes du modèle de Wires.

La Figure 49 montre le diagramme de classe SwitchingEquipment.

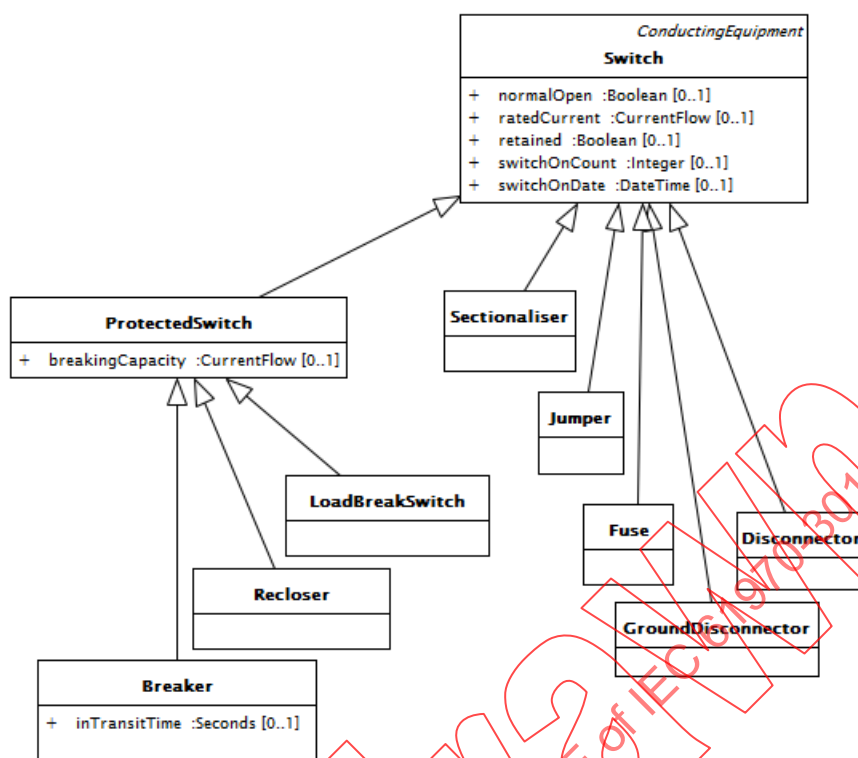


Figure 49 – Diagramme de classe Wires::SwitchingEquipment

Montre la structure d'héritage de SwitchingEquipment.

La Figure 50 montre le diagramme de classe WiresPhaseModel.

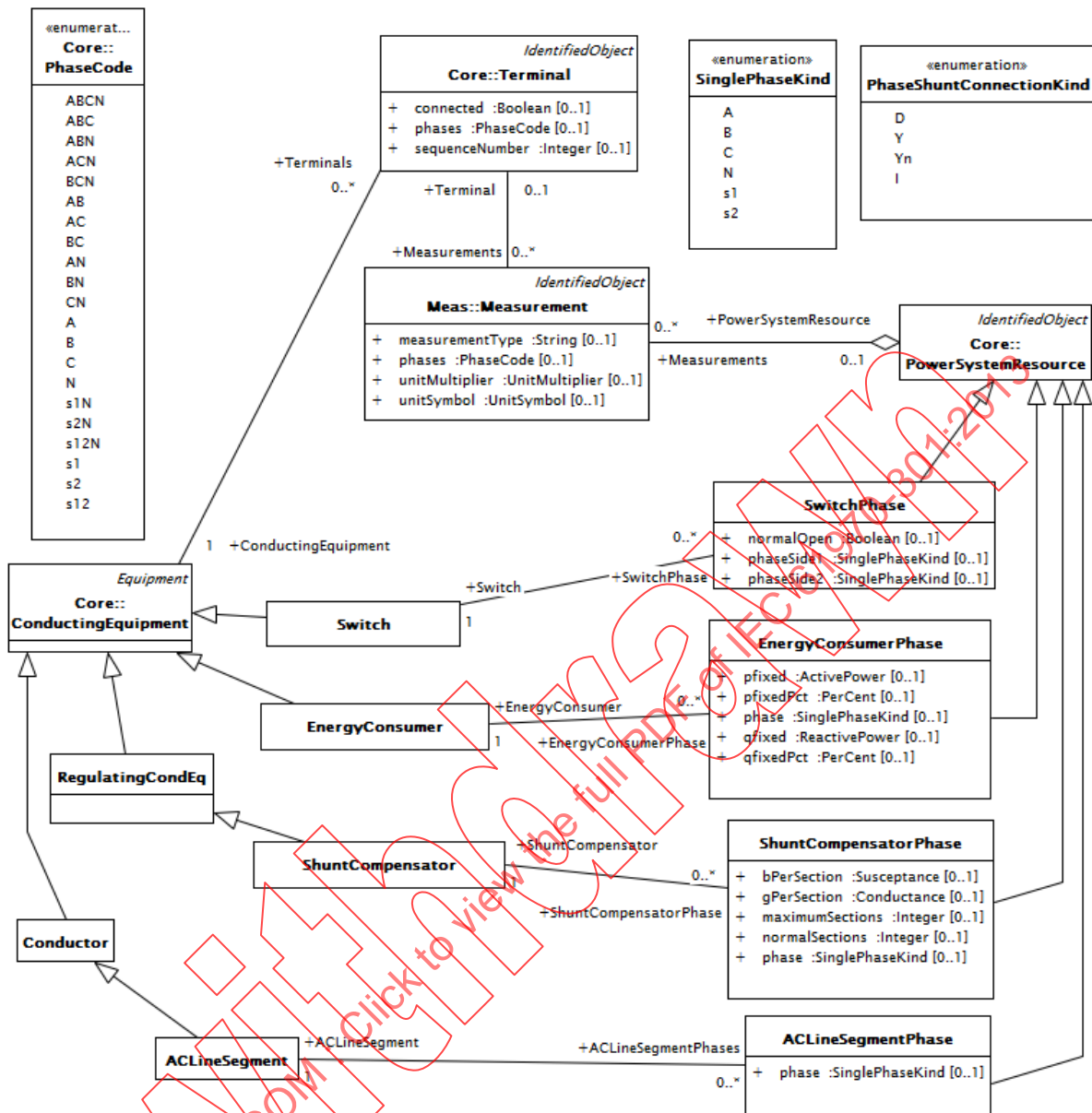


Figure 50 – Diagramme de classe Wires::WiresPhaseModel

Montre en détail la connexion de phase quand les appareils sont modélisés jusqu'aux composants de phase à l'intérieur d'un équipement conducteur.

La Figure 51 montre le diagramme de classe Datatypes.

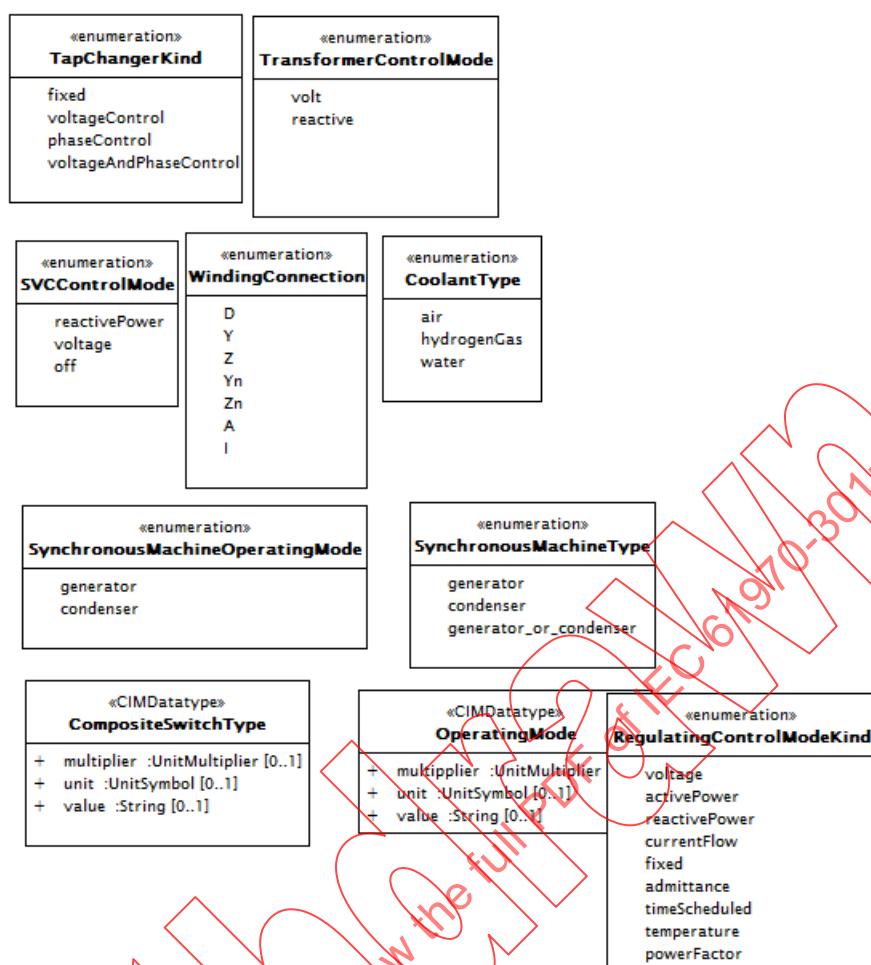


Figure 51 – Diagramme de classe Wires::Datatypes

Ce diagramme montre les types de données spécifiques au paquetage Wires.

La Figure 52 montre le diagramme de classe InheritanceHierarchy.

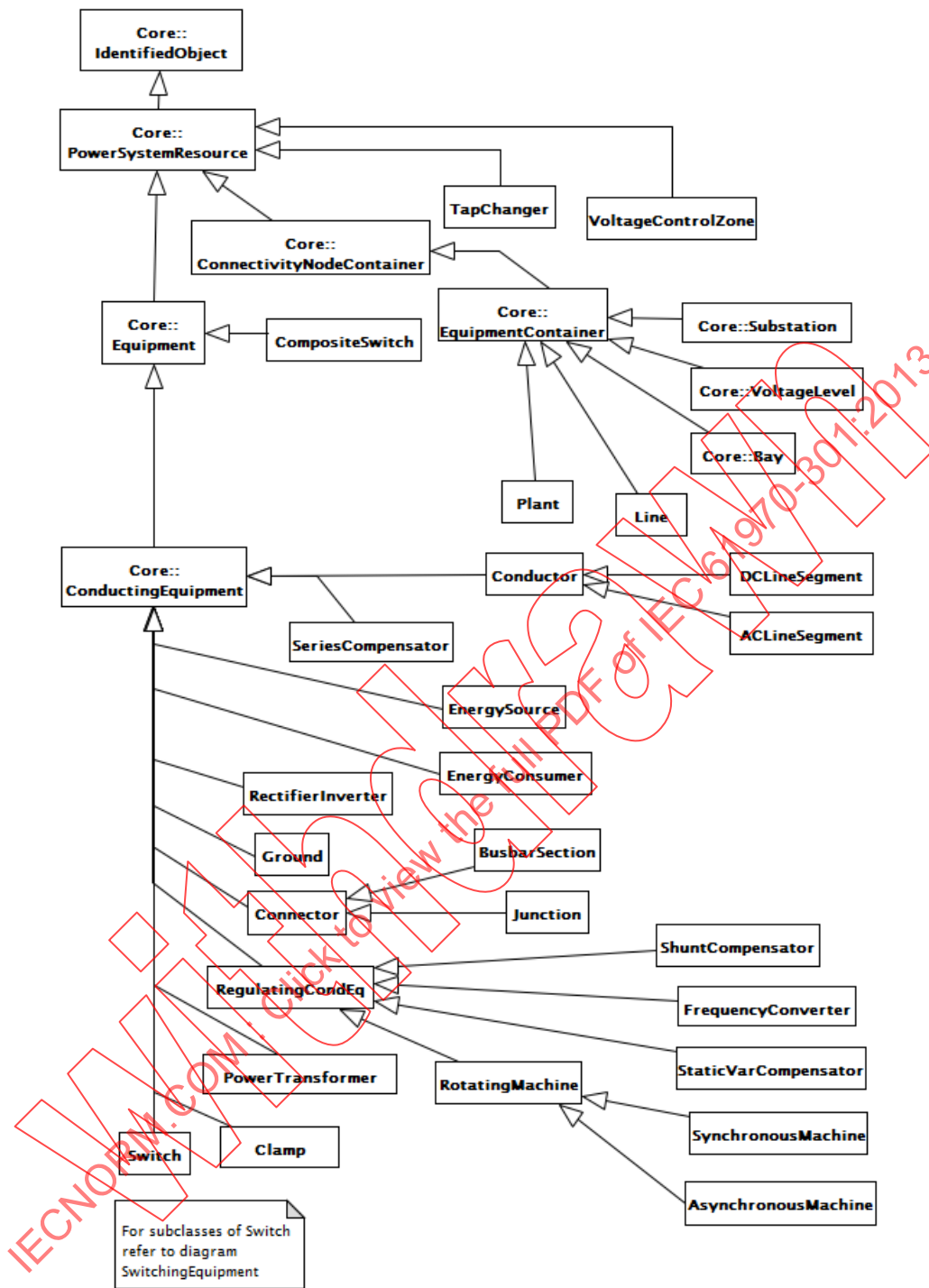


Figure 52 – Diagramme de classe Wires::InheritanceHierarchy

Ce diagramme décrit les relations d'héritage entre des classes de Wire ou liées à celui-ci.

La Figure 53 montre le diagramme de classe LineModel.

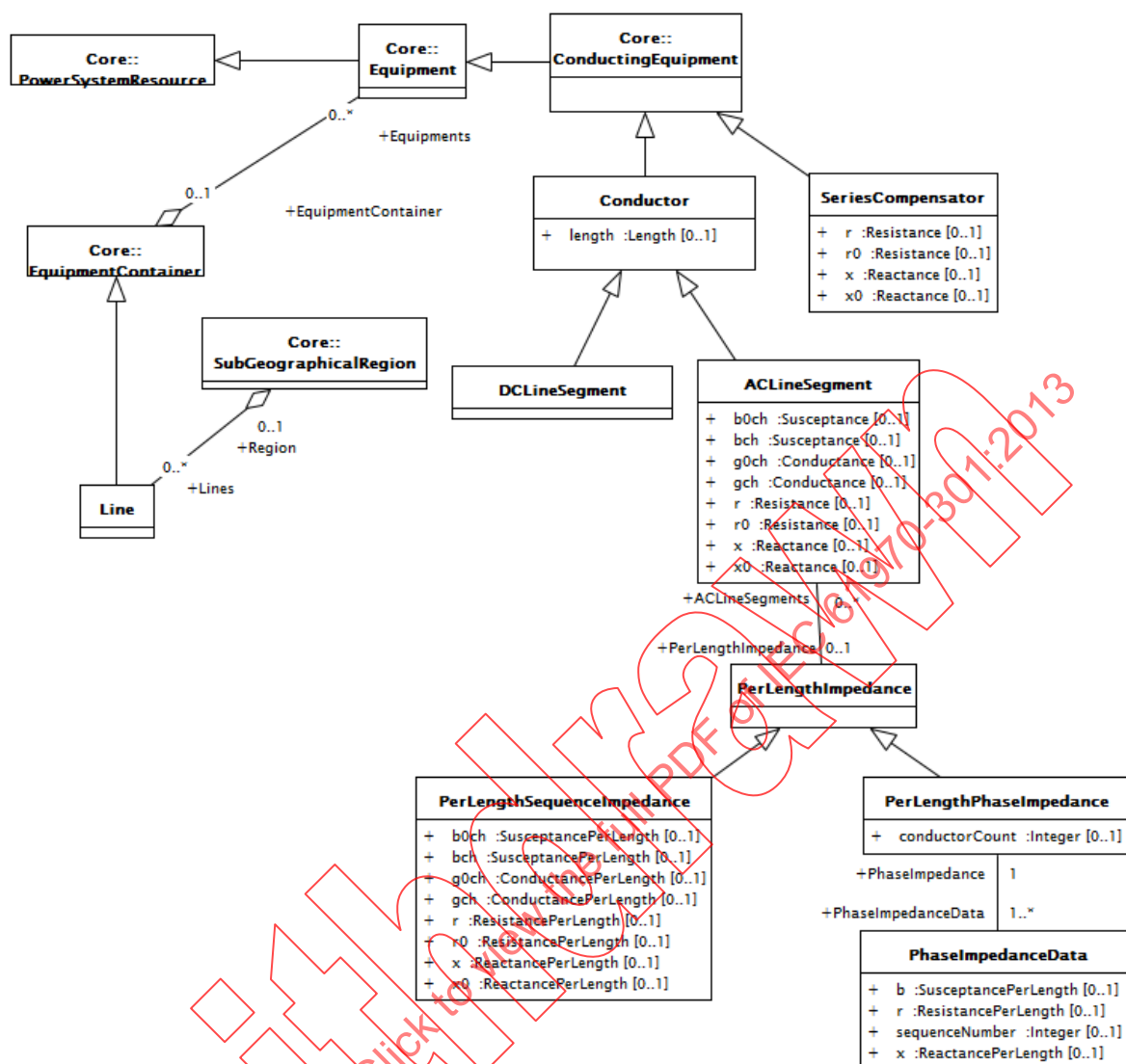


Figure 53 – Diagramme de classe Wires::LineModel

Ce diagramme montre toutes les classes liées au modèle de lignes de transport.

La Figure 54 montre le diagramme de classe NamingHierarchyPart1.

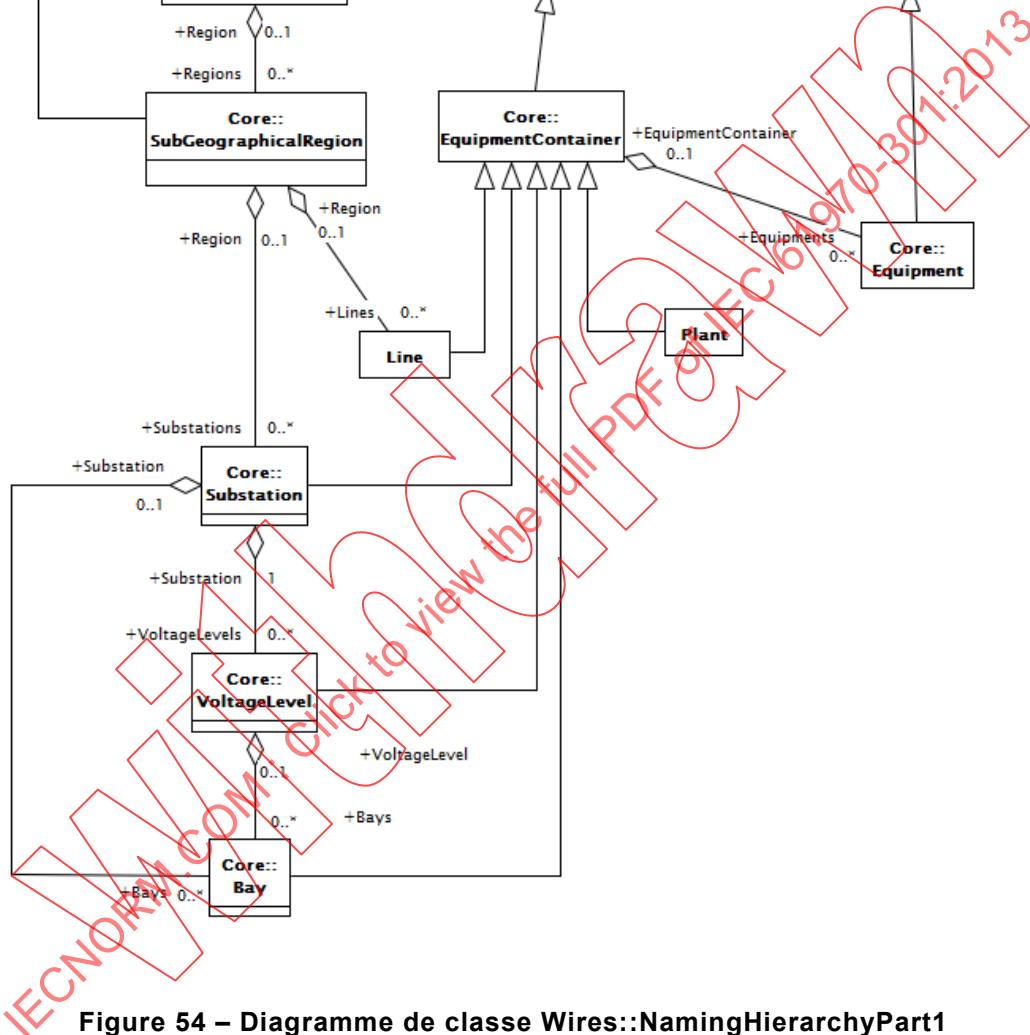


Figure 54 – Diagramme de classe Wires::NamingHierarchyPart1

Le diagramme montre la partie supérieure de la hiérarchie de dénomination. La hiérarchie est une façon d'organiser et de nommer l'équipement. Par conséquent, elle est intimement liée à l'IdentifiedObject et à ses attributs. La classe Equipment est en outre spécialisée en de nombreux sous-types qui sont tous contenus par les sous-classes de EquipmentContainer.

La Figure 55 montre le diagramme de classe NamingHierarchyPart2.

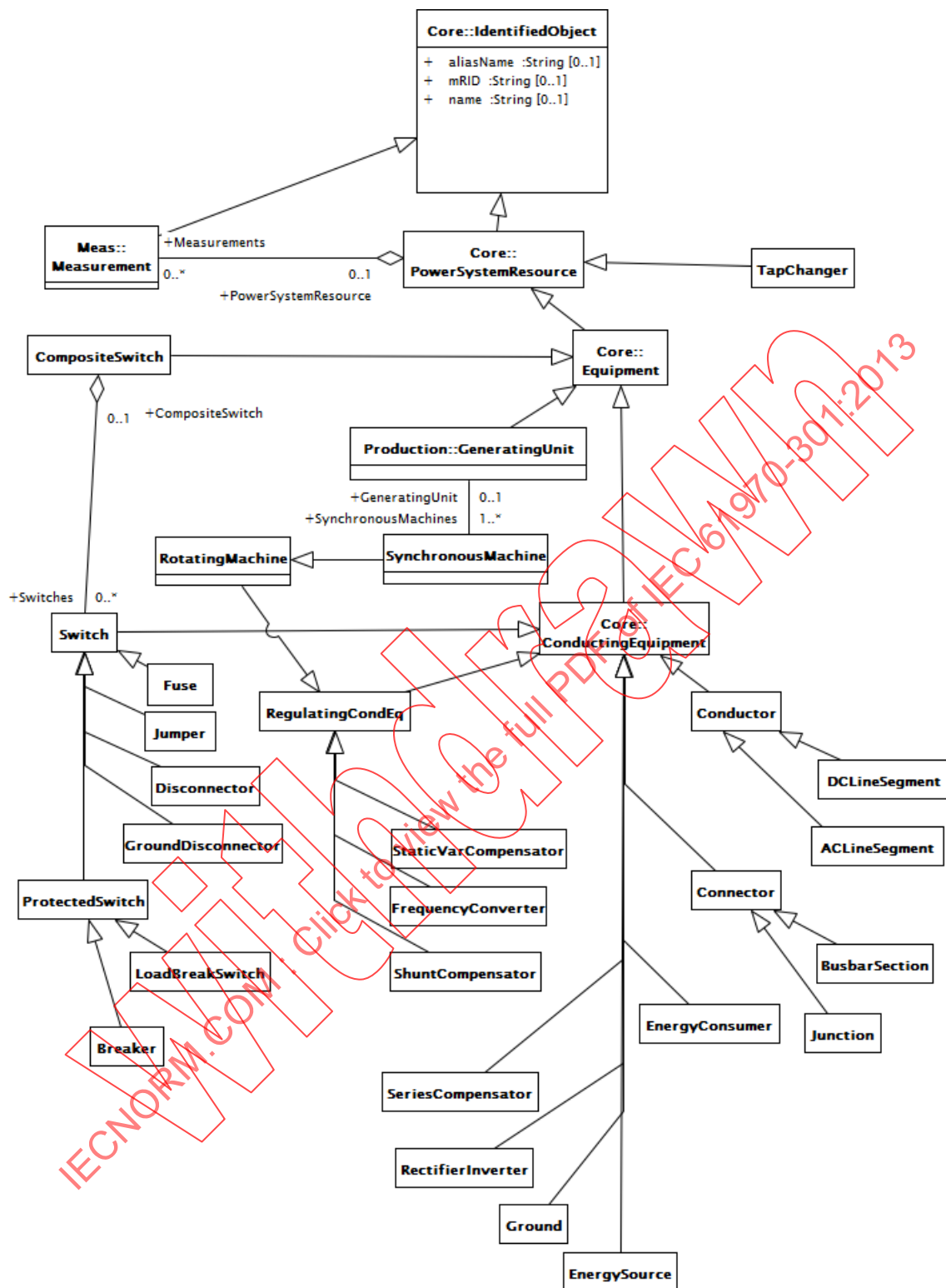


Figure 55 – Diagramme de classe Wires::NamingHierarchyPart2

Le diagramme montre la partie inférieure de la hiérarchie de dénomination. Il montre comment la classe Equipment est en outre spécialisée en de nombreux sous-types qui sont tous contenus par les sous-classes de EquipmentContainer. Pour les sous-classes du conteneur d'équipement, se référer à la partie 1 du diagramme.

La Figure 56 montre le diagramme de classe RegulatingEquipment.

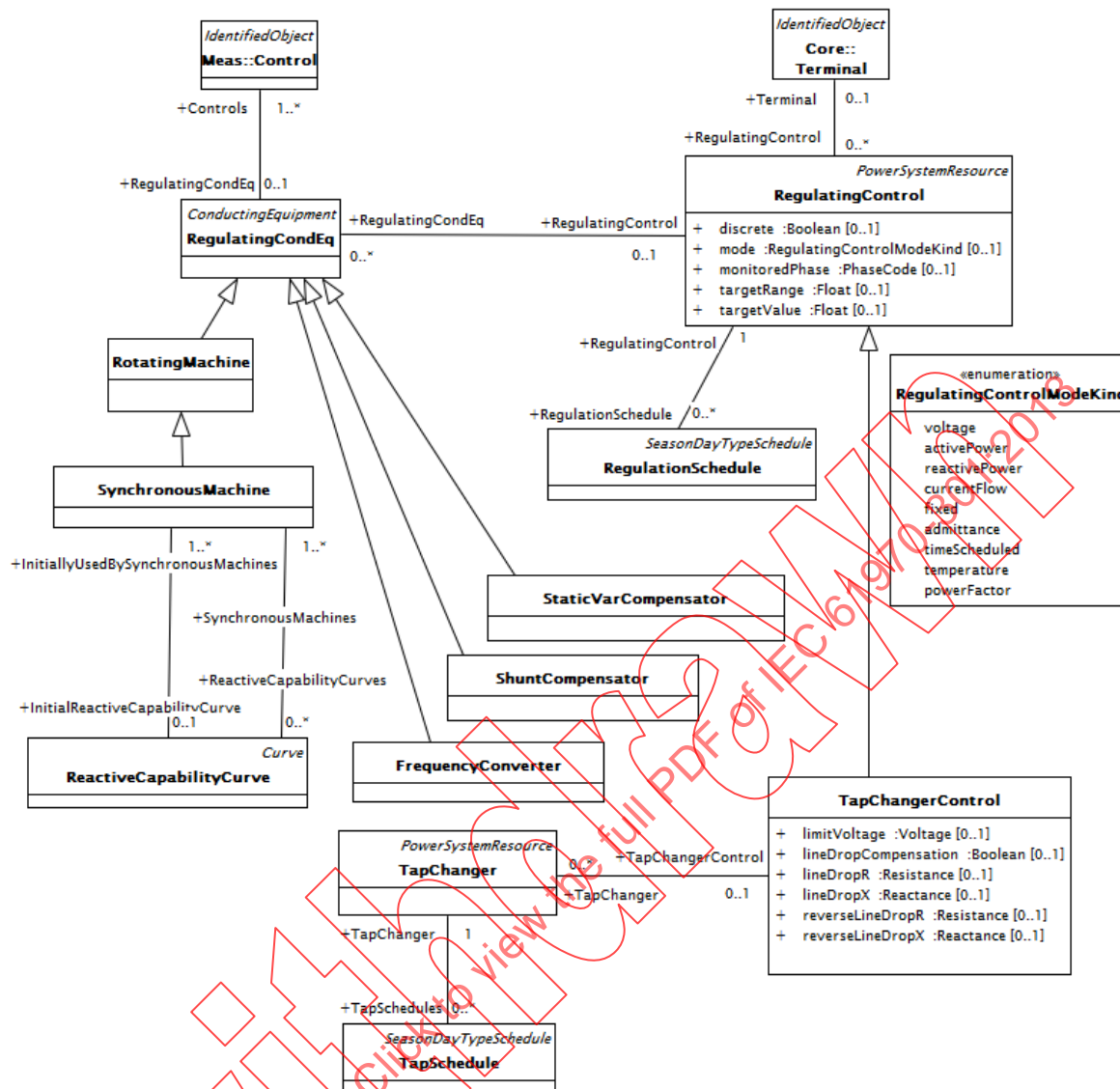


Figure 56 – Diagramme de classe Wires::RegulatingEquipment

Le diagramme montre toutes les classes liées à l'équipement de régulation et à la compensation de puissance réactive.

La Figure 57 montre le diagramme de classe TapChanger.

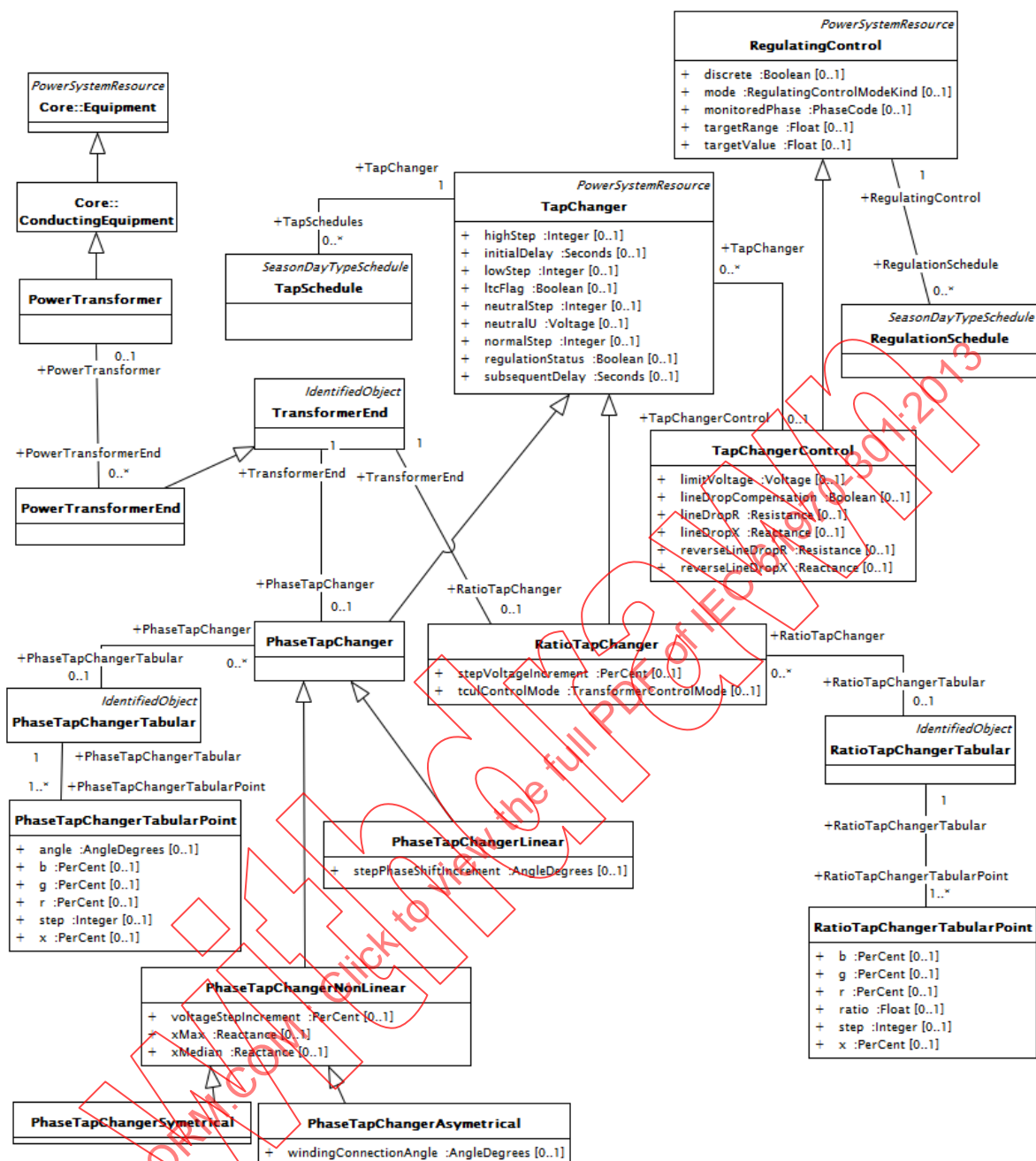


Figure 57 – Diagramme de classe Wires::TapChanger

Ce diagramme montre toutes les classes liées au modèle de prise de transformateur.

La Figure 58 montre le diagramme de classe VoltageControl.

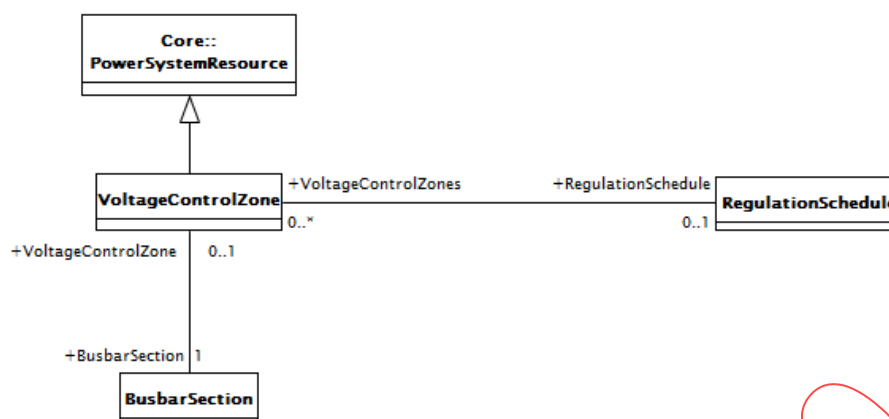


Figure 58 – Diagramme de classe Wires::VoltageControl

Ce diagramme montre toutes les classes liées à la régulation de tension de zone.

La Figure 59 montre le diagramme de classe Transformer.

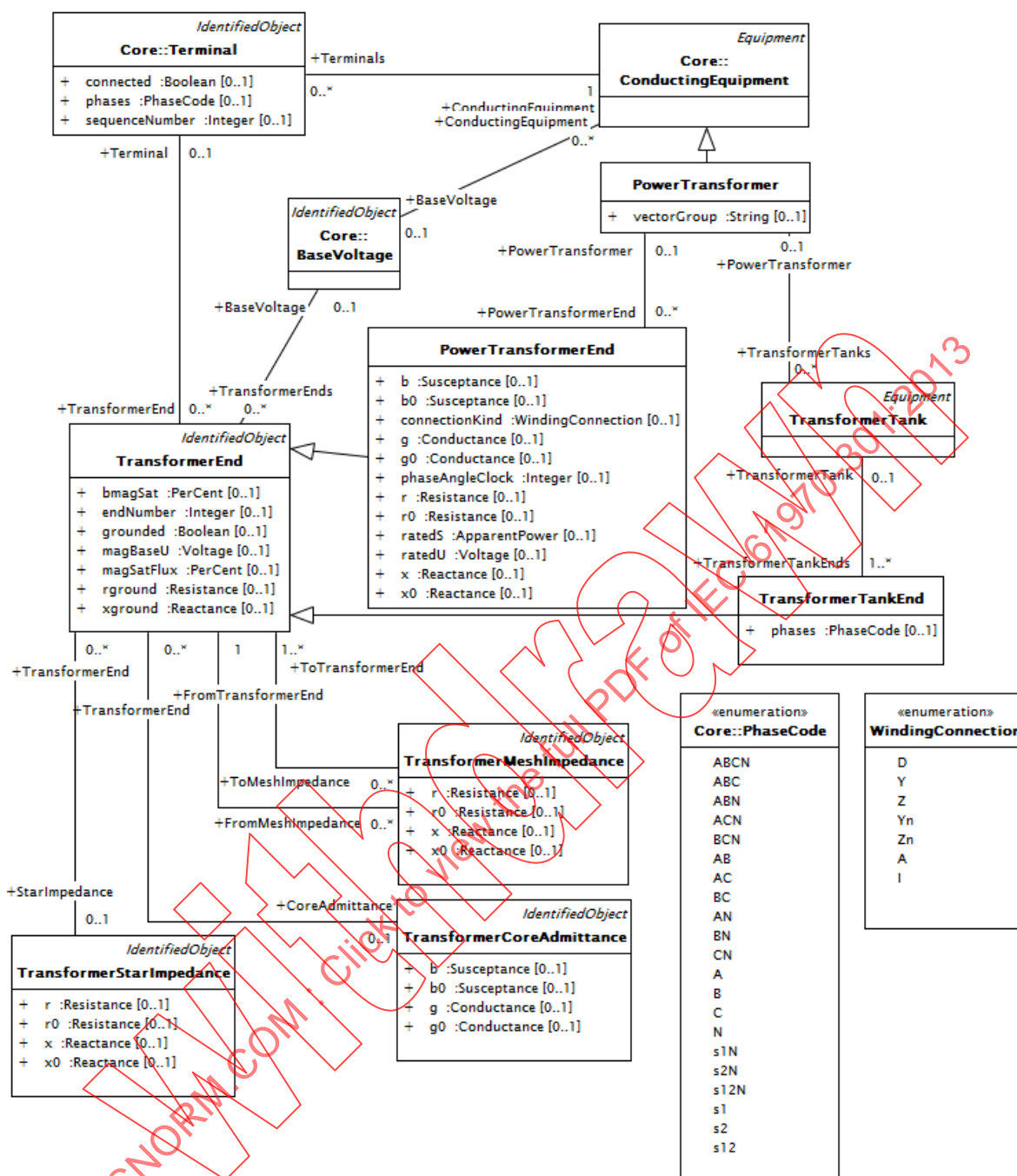


Figure 59 – Diagramme de classe Wires::Transformer

Ce diagramme montre les classes relatives au modèle central de transformateur mais ne montre pas en détail le changeur de prise.

6.8.2 ACLineSegment

Un fil ou une combinaison de fils ayant des caractéristiques électriques cohérentes, et formant un seul système électrique, utilisé pour transporter du courant alternatif entre des points du réseau.

Pour les lignes triphasées transposées symétriques, il suffit d'utiliser les attributs du segment de ligne qui décrivent les impédances et les admittances pour la longueur totale du segment.

Les impédances peuvent en outre être calculées en utilisant la longueur et les impédances associées par unité de longueur.

Le Tableau 166 montre tous les attributs de ACLineSegment.

Tableau 166 – Attributs de Wires::ACLineSegment

nom	type	description
b0ch	Susceptance	Susceptance shunt homopolaire (de chargement), uniformément répartie, de la section de ligne complète.
bch	Susceptance	Susceptance shunt directe (de chargement), uniformément répartie, de la section de ligne complète. Cette valeur représente le chargement total sur la longueur totale de la ligne.
g0ch	Conductance	Conductance shunt homopolaire (de chargement), uniformément répartie, de la section de ligne complète.
gch	Conductance	Conductance shunt directe (de chargement), uniformément répartie, de la section de ligne complète.
r	Resistance	Résistance série directe de la section de ligne complète.
r0	Resistance	Résistance série homopolaire de la section de ligne complète.
x	Reactance	Réactance série directe de la section de ligne complète.
x0	Reactance	Réactance série homopolaire de la section de ligne complète.
length	Longueur	hérité de: Conducteur
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 167 montre toutes les extrémités d'association de ACLineSegment avec d'autres classes.

Tableau 167 – Extrémités d'association de Wires::ACLineSegment avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[1..1]	[0..*] Clamp	Clamp	Les serres-fils connectées au segment de ligne.
[0..*]	[0..1] PerLengthImpedance	PerLengthImpedance	Impédance par unité de longueur pour ce segment de ligne.
[1..1]	[0..*] Cut	Cut	Cuts appliqués au segment de ligne.
[1..1]	[0..*] ACLineSegmentPhases	ACLineSegmentPhase	Les phases de segment de ligne qui appartiennent au segment de ligne.
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.3 ACLineSegmentPhase

Représente un fil unique sur un segment de ligne à courant alternatif.

Le Tableau 168 montre tous les attributs de ACLineSegmentPhase.

Tableau 168 – Attributs de Wires::ACLineSegmentPhase

nom	type	description
phase	SinglePhaseKind	La connexion de phase du fil aux deux extrémités.
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 169 montre toutes les extrémités d'association de ACLineSegmentPhase avec d'autres classes.

Tableau 169 – Extrémités d'association de Wires::ACLineSegmentPhase avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[1..1] ACLineSegment	ACLineSegment	Le segment de ligne auquel appartient la phase.
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.4 AsynchronousMachine

Une machine tournante dont l'arbre tourne de manière asynchrone par rapport au champ électrique. Aussi appelée machine à induction sans connexion externe aux enroulements du rotor, par exemple une machine à induction à cage d'écureuil.

Le Tableau 170 montre tous les attributs de AsynchronousMachine.

Tableau 170 – Attributs de Wires::AsynchronousMachine

nom	type	description
rr1	Resistance	Résistance de l'enroulement amortisseur 1 (Damper 1).
rr2	Resistance	Résistance de l'enroulement amortisseur 2 (Damper 1).
tpo	Seconds	Constante de temps du rotor transitoire (supérieure à tppo).
tppo	Seconds	Constante de temps du rotor sous-transitoire (supérieure à 0).
xlr1	Reactance	Réactance de fuite de l'enroulement amortisseur 1 (Damper 1).
xlr2	Reactance	Réactance de fuite de l'enroulement amortisseur 2 (Damper 1).
xm	Reactance	Réactance magnétisante.
xp	Reactance	Réactance transitoire (non saturée) (supérieure ou égale à xxp).
xpp	Reactance	Réactance sous-transitoire (non saturée) (supérieure à XI).
xs	Reactance	Réactance synchrone (supérieure à xp).
damping	Float	hérité de: RotatingMachine
inertia	Seconds	hérité de: RotatingMachine
ratedS	ApparentPower	hérité de: RotatingMachine
saturationFactor	Float	hérité de: RotatingMachine
saturationFactor120	Float	hérité de: RotatingMachine
statorLeakageReactance	Reactance	hérité de: RotatingMachine
statorResistance	Resistance	hérité de: RotatingMachine
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 171 montre toutes les extrémités d'association de AsynchronousMachine avec d'autres classes.

Tableau 171 – Extrémités d'association de Wires::AsynchronousMachine avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] RegulatingControl	RegulatingControl	hérité de: RegulatingCondEq
[0..1]	[1..*] Controls	Control	hérité de: RegulatingCondEq
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.5 Breaker

Dispositif de commutation mécanique capable d'établir, de transporter et d'interrompre des courants dans des conditions normales de circuit et aussi d'établir, de transporter pendant une durée déterminée et d'interrompre des courants dans des conditions anormales spécifiées de circuit, par exemple celles d'un court-circuit.

Le Tableau 172 montre tous les attributs de Breaker.

Tableau 172 – Attributs de Wires::Breaker

nom	type	description
inTransitTime	Seconds	Le délai de transition de l'état ouvert à l'état fermé.
breakingCapacity	CurrentFlow	hérité de: ProtectedSwitch
normalOpen	Boolean	hérité de: Switch
ratedCurrent	CurrentFlow	hérité de: Switch
switchOnCount	Integer	hérité de: Switch
switchOnDate	DateTime	hérité de: Switch
retained	Boolean	hérité de: Switch
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 173 montre toutes les extrémités d'association de Breaker avec d'autres classes.

Tableau 173 – Extrémités d'association de Wires::Breaker avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..*] OperatedByProtectionEquipment	ProtectionEquipment	hérité de: ProtectedSwitch
[1..1]	[0..*] RecloseSequences	RecloseSequence	hérité de: ProtectedSwitch
[1..1]	[0..*] SwitchSchedules	SwitchSchedule	hérité de: Switch
[1..1]	[0..*] SwitchPhase	SwitchPhase	hérité de: Switch
[0..*]	[0..1] CompositeSwitch	CompositeSwitch	hérité de: Switch
[0..*]	[0..*] SwitchingOperations	SwitchingOperation	hérité de: Switch
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.6 BusbarSection

Conducteur, ou groupe de conducteurs, ayant une impédance négligeable, servant à connecter un autre équipement conducteur au sein d'un seul poste.

Les mesures de tension sont habituellement obtenues à partir de VoltageTransformers (transformateurs de tension) connectés à des sections de jeu de barres. Un tronçon de jeu de barres peut avoir plusieurs bornes physiques mais, pour l'analyse, elle est modélisée avec une seule borne logique.

Le Tableau 174 montre tous les attributs de BusbarSection.

Tableau 174 – Attributs de Wires::BusbarSection

nom	type	description
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 175 montre toutes les extrémités d'association de BusbarSection avec d'autres classes.

Tableau 175 – Extrémités d'association de Wires::BusbarSection avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[1..1]	[0..1] VoltageControlZone	VoltageControlZone	Une VoltageControlZone (zone de contrôle de la tension) est contrôlée par une BusbarSection (section de jeu de barres) déterminée.
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.7 Clamp

Un serre-fil (Clamp) est une connexion galvanique à un segment de ligne où est connecté l'autre équipement. Un serre-fil ne coupe pas le segment de ligne.

Un Clamp est un ConductingEquipment et a une borne (Terminal) comportant un ConnectivityNode. Tout autre ConductingEquipment peut être connecté au Clamp ConnectivityNode.

Le Tableau 176 montre tous les attributs de Clamp.

Tableau 176 – Attributs de Wires::Clamp

nom	type	description
lengthFromTerminal1	Longueur	La longueur jusqu'à l'endroit où se trouve la pince à partir du premier côté du segment de ligne, c'est-à-dire la borne du segment de ligne dont le numéro d'ordre est égal à 1.
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 177 montre toutes les extrémités d'association de Clamp avec d'autres classes.

Tableau 177 – Extrémités d'association de Wires::Clamp avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[1..1] ACLineSegment	ACLineSegment	Le segment de ligne auquel est connecté le serre-fil.
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.8 CompositeSwitch

Modèle d'un ensemble de Switches (commutateurs) individuels compris dans la même armoire et si possible, avec des verrouillages qui restreignent la combinaison des positions de commutation. On peut les trouver généralement dans des réseaux électriques de distribution en moyenne tension.

Un CompositeSwitch peut représenter un Ring-Main-Unit (RMU) ou un appareillage de commutation monté par bloc, avec des dispositifs internes primaires tels qu'un jeu de barres interne plus 3 ou 4 interrupteurs internes qui peuvent être individuellement ouverts ou fermés. Un CompositeSwitch et un ensemble de Switches qu'il contient peuvent aussi être utilisés pour représenter un commutateur à plusieurs positions, par exemple un commutateur qui peut connecter un circuit à la terre (Ground), à l'air (Open), ou à un jeu de barres (Busbar).

Le Tableau 178 montre tous les attributs de CompositeSwitch.

Tableau 178 – Attributs de Wires::CompositeSwitch

nom	type	description
compositeSwitchType	CompositeSwitchType	Un code alphanumérique qui peut être utilisé comme référence pour une information supplémentaire telle que la description du schéma de verrouillage ("interlocking scheme"), s'il y en a un.
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 179 montre toutes les extrémités d'association de CompositeSwitch avec d'autres classes.

Tableau 179 – Extrémités d'association de Wires::CompositeSwitch avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[0..*] Switches	Switch	Les Switch contenus dans ce commutateur combiné (Composite switch).
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.9 Type de données CompositeSwitchType

Un code alphanumérique qui peut être utilisé comme référence pour une information supplémentaire telle que la description du schéma de verrouillage ("interlocking scheme"), s'il y en a un.

Le Tableau 180 montre tous les attributs de CompositeSwitchType.

Tableau 180 – Attributs de Wires::CompositeSwitchType

nom	type	description
multiplier	UnitMultiplier	Multiplicateur.
unit	UnitSymbol	Unités.
value	String	Valeur.

6.8.10 Conducteur

Combinaison de matériau conducteur ayant des caractéristiques électriques cohérentes et formant un seul système électrique, utilisé pour transporter du courant entre des points dans le système électrique.

Le Tableau 181 montre tous les attributs de Conductor.

Tableau 181 – Attributs de Wires::Conductor

nom	type	description
length	Longueur	Longueur du segment pour calculer les capacités d'une section de ligne
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 182 montre toutes les extrémités d'association de Conductor avec d'autres classes.

Tableau 182 – Extrémités d'association de Wires::Conductor avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.11 Connecteur

Conducteur, ou groupe de conducteurs, d'une impédance négligeable, servant à connecter un autre équipement conducteur au sein d'un seul poste et qui sont modélisés avec une seule borne logique.

Le Tableau 183 montre tous les attributs de Connector.

Tableau 183 – Attributs de Wires::Connector

nom	type	description
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 184 montre toutes les extrémités d'association de Connector avec d'autres classes.

Tableau 184 – Extrémités d'association de Wires::Connector avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.12 Enumération CoolantType

Méthode de refroidissement d'une machine.

Le Tableau 185 montre tous les libellés de CoolantType.

Tableau 185 – Libellés de Wires::CoolantType

libellé	description
air	Air.
hydrogenGas	Gaz hydrogène..
water	Eau.

6.8.13 Cut

Un cut sépare en deux parties un segment de ligne. Le cut se présente comme un commutateur qui est inséré entre ces deux parties et les connecte l'une à l'autre. Comme le cut est normalement ouvert, il n'y a pas de connexion galvanique entre les deux parties du segment de ligne. Il est cependant possible de fermer le cut pour obtenir une connexion galvanique.

Les bornes du cut sont orientées vers les bornes du segment de ligne qui ont le même numéro d'ordre. La borne de cut dont le numéro d'ordre est égal à 1 est par conséquent orientée vers la borne du segment de ligne dont le numéro d'ordre est égal à 1.

Les bornes du cut ont aussi une fonction de point de connexion pour des cavaliers et d'autres équipements, par exemple un générateur mobile. Pour y parvenir, des nœuds de connectivité sont placés aux bornes du cut. Une fois que les nœuds de connectivité sont en place, il est possible d'y connecter n'importe quel équipement.

Le Tableau 186 montre tous les attributs de Cut.

Tableau 186 – Attributs de Wires::Cut

nom	type	description
lengthFromTerminal1	Longueur	La longueur jusqu'à l'endroit où se trouve le cut à partir du premier côté du segment de ligne du cut, c'est-à-dire la borne du segment de ligne dont le numéro d'ordre est égal à 1.
normalOpen	Boolean	hérité de: Switch
ratedCurrent	CurrentFlow	hérité de: Switch
switchOnCount	Integer	hérité de: Switch
switchOnDate	DateTime	hérité de: Switch
retained	Boolean	hérité de: Switch
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 187 montre toutes les extrémités d'association de Cut avec d'autres classes.

Tableau 187 – Extrémités d'association de Wires::Cut avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[1..1] ACLineSegment	ACLineSegment	Le segment de ligne auquel est appliqué le cut.
[1..1]	[0..*] SwitchSchedules	SwitchSchedule	hérité de: Switch
[1..1]	[0..*] SwitchPhase	SwitchPhase	hérité de: Switch
[0..*]	[0..1] CompositeSwitch	CompositeSwitch	hérité de: Switch
[0..*]	[0..*] SwitchingOperations	SwitchingOperation	hérité de: Switch
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRTType	PSRTType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.14 DCLineSegment

Fil ou combinaison de fils non isolés les uns des autres, ayant des caractéristiques électriques cohérentes, utilisés pour transporter du courant continu entre des points dans la section courant continu du système électrique.

Le Tableau 188 montre tous les attributs de DCLineSegment.

Tableau 188 – Attributs de Wires::DCLineSegment

nom	type	description
dcSegmentInductance	Inductance	Inductance du segment de ligne courant continu.
dcSegmentResistance	Resistance	Résistance du segment de ligne courant continu.
length	Longueur	hérité de: Conducteur
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 189 montre toutes les extrémités d'association de DCLineSegment avec d'autres classes.

Tableau 189 – Extrémités d'association de Wires::DCLineSegment avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.15 Disconnecter

Dispositif de commutation mécanique exploité manuellement ou par un moteur utilisé pour changer les connexions dans un circuit ou pour isoler un circuit ou un équipement d'une source d'électricité. Les circuits doivent être ouverts ou fermés quand un courant négligeable est interrompu ou produit.

Le Tableau 190 montre tous les attributs de Disconnecter.

Tableau 190 – Attributs de Wires::Disconnecter

nom	type	description
normalOpen	Boolean	hérité de: Switch
ratedCurrent	CurrentFlow	hérité de: Switch
switchOnCount	Integer	hérité de: Switch
switchOnDate	DateTime	hérité de: Switch
retained	Boolean	hérité de: Switch
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 191 montre toutes les extrémités d'association de Disconnecter avec d'autres classes.

Tableau 191 – Extrémités d'association de Wires::Disconnecter avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[1..1]	[0..*] SwitchSchedules	SwitchSchedule	hérité de: Switch
[1..1]	[0..*] SwitchPhase	SwitchPhase	hérité de: Switch
[0..*]	[0..1] CompositeSwitch	CompositeSwitch	hérité de: Switch
[0..*]	[0..*] SwitchingOperations	SwitchingOperation	hérité de: Switch
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.16 EnergyConsumer

Consommateur générique d'énergie – un point de consommation sur le modèle de système électrique.

Le Tableau 192 montre tous les attributs de EnergyConsumer.

Tableau 192 – Attributs de Wires::EnergyConsumer

nom	type	description
customerCount	Integer	Nombre de clients individuels représentés par cette demande.
grounded	WindingConnection	Utilisé pour des connexions Yn et Zn. Vrai si le neutre est solidement mis à la terre.
pfixed	ActivePower	Puissance active de la charge dont la grandeur est fixe. La convention de signe charge est utilisée, en d'autres termes le signe positif signifie un flux à partir d'un nœud.
pfixedPct	PerCent	Puissance active fixe exprimée en pourcentage de la puissance active fixe du groupe de charge. La convention de signe charge est utilisée, en d'autres termes le signe positif signifie un flux à partir d'un nœud.
phaseConnection	PhaseShuntConnectionKind	Le type de connexion de phase, par exemple montage en étoile ou en triangle.
qfixed	ReactivePower	Puissance réactive de la charge dont la grandeur est fixe. La convention de signe charge est utilisée, en d'autres termes le signe positif signifie un flux à partir d'un nœud.
qfixedPct	PerCent	Puissance réactive fixe exprimée en pourcentage de la puissance réactive fixe du groupe de charge. La convention de signe charge est utilisée, en d'autres termes le signe positif signifie un flux à partir d'un nœud.
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 193 montre toutes les extrémités d'association de EnergyConsumer avec d'autres classes.

Tableau 193 – Extrémités d'association de Wires::EnergyConsumer avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] LoadResponse	LoadResponseCharacteristic	La caractéristique de réponse de charge pour cette charge. Si elle manque, cette charge est supposée être une puissance constante.
[1..1]	[0..*] EnergyConsumerPhase	EnergyConsumerPhase	Les modèles individuels de phase pour ce consommateur d'énergie.
[1..*]	[0..1] PowerCutZone	PowerCutZone	Le consommateur d'énergie est affecté à cette zone de coupure de puissance.
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.17 EnergyConsumerPhase

Une phase unique d'un consommateur d'énergie.

Le Tableau 194 montre tous les attributs de EnergyConsumerPhase.

Tableau 194 – Attributs de Wires::EnergyConsumerPhase

nom	type	description
pfixed	ActivePower	Puissance active de la charge dont la grandeur est fixe. La convention de signe charge est utilisée, en d'autres termes le signe positif signifie un flux à partir d'un nœud.
pfixedPct	PerCent	Puissance active fixe exprimée en pourcentage de la puissance active fixe du groupe de charge. La convention de signe charge est utilisée, en d'autres termes le signe positif signifie un flux à partir d'un nœud.
phase	SinglePhaseKind	Phase de ce composant consommateur d'énergie. Si le consommateur d'énergie est connecté en étoile, la connexion va de la phase indiquée à la terre centrale ou au point neutre. Si le consommateur d'énergie est connecté en triangle, la phase indique qu'un consommateur d'énergie est connecté de la phase indiquée à la phase logique suivante non neutre.
qfixed	ReactivePower	Puissance réactive de la charge dont la grandeur est fixe. La convention de signe charge est utilisée, en d'autres termes le signe positif signifie un flux à partir d'un nœud.
qfixedPct	PerCent	Puissance réactive fixe exprimée en pourcentage de la puissance réactive fixe du groupe de charge. La convention de signe charge est utilisée, en d'autres termes le signe positif signifie un flux à partir d'un nœud.
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 195 montre toutes les extrémités d'association de EnergyConsumerPhase avec d'autres classes.

Tableau 195 – Extrémités d'association de Wires::EnergyConsumerPhase avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[1..1] EnergyConsumer	EnergyConsumer	Le consommateur d'énergie auquel appartient cette phase.
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.18 EnergySource

Équivalent générique pour un fournisseur d'énergie sur un niveau de tension de transport ou de distribution.

Le Tableau 196 montre tous les attributs de EnergySource.

Tableau 196 – Attributs de Wires::EnergySource

nom	type	description
activePower	ActivePower	Charge de source haute tension.
nominalVoltage	Voltage	Tension nominale phase à phase.
r	Resistance	Résistance directe de Thevenin.
r0	Resistance	Résistance homopolaire de Thevenin.
rn	Resistance	Résistance inverse de Thevenin.
voltageAngle	AngleRadians	Angle de phase d'un circuit à phase ouvert.
voltageMagnitude	Voltage	Valeur de la tension phase à phase d'un circuit ouvert.
x	Reactance	Réactance directe de Thevenin.
x0	Reactance	Réactance homopolaire de Thevenin.
xn	Reactance	Réactance inverse de Thevenin.
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 197 montre toutes les extrémités d'association de EnergySource avec d'autres classes.

Tableau 197 – Extrémités d'association de Wires::EnergySource avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.19 FrequencyConverter

Un dispositif pour convertir une fréquence en une autre (par exemple, fréquence F1 en F2) comprend une paire d'instances FrequencyConverter. L'une convertit de F1 en CC (courant continu), l'autre de CC en F2.

Le Tableau 198 montre tous les attributs de FrequencyConverter.

Tableau 198 – Attributs de Wires::FrequencyConverter

nom	type	description
fréquence	Fréquence	Fréquence du côté CA (courant alternatif).
maxP	ActivePower	La puissance active maximale du côté CC à laquelle il convient que le convertisseur de fréquence fonctionne.
maxU	Voltage	La tension maximale du côté CC à laquelle il convient que le convertisseur de fréquence fonctionne.
minP	ActivePower	La puissance active minimale du côté CC à laquelle il convient que le convertisseur de fréquence fonctionne.
minU	Voltage	La tension minimale du côté CC à laquelle il convient que le convertisseur de fréquence fonctionne.
operatingMode	OperatingMode	Mode de fonctionnement du convertisseur de fréquence
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 199 montre toutes les extrémités d'association de FrequencyConverter avec d'autres classes.

Tableau 199 – Extrémités d'association de Wires::FrequencyConverter avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] RegulatingControl	RegulatingControl	hérité de: RegulatingCondEq
[0..1]	[1..*] Controls	Control	hérité de: RegulatingCondEq
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.20 Fuse

Dispositif de protection contre les surintensités avec un fusible à ouverture de circuit qui est chauffé et rompu quand une surintensité le traverse. Un fusible est considéré comme un organe de coupure parce qu'il interrompt le courant.

Le Tableau 200 montre tous les attributs de Fuse.

Tableau 200 – Attributs de Wires::Fuse

nom	type	description
normalOpen	Boolean	hérité de: Switch
ratedCurrent	CurrentFlow	hérité de: Switch
switchOnCount	Integer	hérité de: Switch
switchOnDate	DateTime	hérité de: Switch
retained	Boolean	hérité de: Switch
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 201 montre toutes les extrémités d'association de Fuse avec d'autres classes.

Tableau 201 – Extrémités d'association de Wires::Fuse avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[1..1]	[0..*] SwitchSchedules	SwitchSchedule	hérité de: Switch
[1..1]	[0..*] SwitchPhase	SwitchPhase	hérité de: Switch
[0..*]	[0..1] CompositeSwitch	CompositeSwitch	hérité de: Switch
[0..*]	[0..*] SwitchingOperations	SwitchingOperation	hérité de: Switch
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.21 Ground

Point commun pour connecter un équipement conducteur mis à la terre tel que des condensateurs-shunts. Le modèle de réseau peut avoir plus d'une mise à la terre.

Le Tableau 202 montre tous les attributs de Ground.

Tableau 202 – Attributs de Wires::Ground

nom	type	description
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 203 montre toutes les extrémités d'association de Ground avec d'autres classes.

Tableau 203 – Extrémités d'association de Wires::Ground avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.22 GroundDisconnector

Un dispositif mécanique de commutation, motorisé ou opéré manuellement, utilisé pour isoler un circuit ou un équipement de Ground (terre).

Le Tableau 204 montre tous les attributs de GroundDisconnector.

Tableau 204 – Attributs de Wires::GroundDisconnector

nom	type	description
normalOpen	Boolean	hérité de: Switch
ratedCurrent	CurrentFlow	hérité de: Switch
switchOnCount	Integer	hérité de: Switch
switchOnDate	DateTime	hérité de: Switch
retained	Boolean	hérité de: Switch
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 205 montre toutes les extrémités d'association de GroundDisconnector avec d'autres classes.

Tableau 205 – Extrémités d'association de Wires::GroundDisconnecter avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[1..1]	[0..*] SwitchSchedules	SwitchSchedule	hérité de: Switch
[1..1]	[0..*] SwitchPhase	SwitchPhase	hérité de: Switch
[0..*]	[0..1] CompositeSwitch	CompositeSwitch	hérité de: Switch
[0..*]	[0..*] SwitchingOperations	SwitchingOperation	hérité de: Switch
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.23 Jumper

Courte section de conducteur avec une impédance négligeable pouvant être enlevée et remplacée manuellement si le circuit n'est plus alimenté. Noter que les branches à impédance nulle peuvent être potentiellement modélisées par d'autres types d'équipement.

Le Tableau 206 montre tous les attributs de Jumper.

Tableau 206 – Attributs de Wires::Jumper

nom	type	description
normalOpen	Boolean	hérité de: Switch
ratedCurrent	CurrentFlow	hérité de: Switch
switchOnCount	Integer	hérité de: Switch
switchOnDate	DateTime	hérité de: Switch
retained	Boolean	hérité de: Switch
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 207 montre toutes les extrémités d'association de Jumper avec d'autres classes.

Tableau 207 – Extrémités d'association de Wires::Jumper avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[1..1]	[0..*] SwitchSchedules	SwitchSchedule	hérité de: Switch
[1..1]	[0..*] SwitchPhase	SwitchPhase	hérité de: Switch
[0..*]	[0..1] CompositeSwitch	CompositeSwitch	hérité de: Switch
[0..*]	[0..*] SwitchingOperations	SwitchingOperation	hérité de: Switch
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.24 Junction

Point auquel un ou plusieurs équipements conducteurs sont connectés avec une résistance homopolaire.

Le Tableau 208 montre tous les attributs de Junction.

Tableau 208 – Attributs de Wires::Junction

nom	type	description
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 209 montre toutes les extrémités d'association de Junction avec d'autres classes.

Tableau 209 – Extrémités d'association de Wires::Junction avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.25 Line

Contient des équipements au-delà du poste appartenant à une ligne de transport d'énergie électrique.

Le Tableau 210 montre tous les attributs de Line.

Tableau 210 – Attributs de Wires::Line

nom	type	description
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 211 montre toutes les extrémités d'association de Line avec d'autres classes.

Tableau 211 – Extrémités d'association de Wires::Line avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] Region	SubGeographicalRegion	La sous-région géographique de la ligne.
[0..1]	[0..*] Equipments	Equipement	hérité de: EquipmentContainer
[1..1]	[0..*] ConnectivityNodes	ConnectivityNode	hérité de: ConnectivityNodeContainer
[0..1]	[0..*] TopologicalNode	TopologicalNode	hérité de: ConnectivityNodeContainer
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.26 LoadBreakSwitch

Dispositif de commutation mécanique capable de produire, transporter et interrompre des courants en conditions normales de fonctionnement.

Le Tableau 212 montre tous les attributs de LoadBreakSwitch.

Tableau 212 – Attributs de Wires::LoadBreakSwitch

nom	type	description
breakingCapacity	CurrentFlow	hérité de: ProtectedSwitch
normalOpen	Boolean	hérité de: Switch
ratedCurrent	CurrentFlow	hérité de: Switch
switchOnCount	Integer	hérité de: Switch
switchOnDate	DateTime	hérité de: Switch
retained	Boolean	hérité de: Switch
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 213 montre toutes les extrémités d'association de LoadBreakSwitch avec d'autres classes.

**Tableau 213 – Extrémités d'association de Wires::LoadBreakSwitch
avec d'autres classes**

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..*] OperatedByProtectionEquipment	ProtectionEquipment	hérité de: ProtectedSwitch
[1..1]	[0..*] RecloseSequences	RecloseSequence	hérité de: ProtectedSwitch
[1..1]	[0..*] SwitchSchedules	SwitchSchedule	hérité de: Switch
[1..1]	[0..*] SwitchPhase	SwitchPhase	hérité de: Switch
[0..*]	[0..1] CompositeSwitch	CompositeSwitch	hérité de: Switch
[0..*]	[0..*] SwitchingOperations	SwitchingOperation	hérité de: Switch
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.27 MutualCoupling

Cette classe représente le couplage mutuel de ligne homopolaire.

Le Tableau 214 montre tous les attributs de MutualCoupling.

Tableau 214 – Attributs de Wires::MutualCoupling

nom	type	description
b0ch	Susceptance	Conductance shunt de couplage mutuel homopolaire (de chargement), uniformément répartie, du segment de ligne complet.
distance11	Longueur	Distance par rapport au début de la région couplée à partir de la première borne de la ligne dont le numéro d'ordre est égal à 1.
distance12	Longueur	Distance par rapport à la fin de la région couplée à partir de la première borne de la ligne dont le numéro d'ordre est égal à 1.
distance21	Longueur	Distance par rapport au début de la région couplée à partir de la deuxième borne de la ligne dont le numéro d'ordre est égal à 1.
distance22	Longueur	Distance par rapport à la fin de la région couplée à partir de la deuxième borne de la ligne dont le numéro d'ordre est égal à 1.
g0ch	Conductance	Conductance shunt de couplage mutuel homopolaire (de chargement), uniformément répartie, du segment de ligne complet.
r0	Resistance	Couplage d'impédance mutuel homopolaire entre branches, résistance.
x0	Reactance	Couplage d'impédance mutuel homopolaire entre branches, réactance.
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 215 montre toutes les extrémités d'association de MutualCoupling avec d'autres classes.

Tableau 215 – Extrémités d'association de Wires::MutualCoupling avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[1..1] Second_Terminal	Terminal	La borne de départ pour le calcul des distances le long de la deuxième branche du couplage mutuel.
[0..*]	[1..1] First_Terminal	Terminal	La borne de départ pour le calcul des distances le long de la première branche du couplage mutuel. Normalement, le couplage mutuel MutualCoupling ne serait utilisé que pour des bornes de segments de ligne CA (c'est-à-dire en courant alternatif). Il convient que les première et seconde bornes d'un couplage mutuel pointent des segments de ligne CA différents.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.28 Type de données OperatingMode

Nom littéral d'un mode opératoire.

Le Tableau 216 montre tous les attributs de OperatingMode.

Tableau 216 – Attributs de Wires::OperatingMode

nom	type	description
multiplier	UnitMultiplier	
unit	UnitSymbol	
value	String	

6.8.29 PerLengthImpedance

Type commun pour des catalogues électriques d'impédance par unité de longueur.

Le Tableau 217 montre tous les attributs de PerLengthImpedance.

Tableau 217 – Attributs de Wires::PerLengthImpedance

nom	type	description
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 218 montre toutes les extrémités d'association de PerLengthImpedance avec d'autres classes.

Tableau 218 – Extrémités d'association de Wires::PerLengthImpedance avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[0..*] ACLineSegments	ACLineSegment	Tous les segments de ligne décrits par cette impédance par unité de longueur.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.30 PerLengthPhaseImpedance

Paramètres linéiques d'impédance et d'admittance pour les lignes à n fils non équilibrées, sous forme de matrice.

Le Tableau 219 montre tous les attributs de PerLengthPhaseImpedance.

Tableau 219 – Attributs de Wires::PerLengthPhaseImpedance

nom	type	description
conductorCount	Integer	Nombre de fils de phase, neutres et autres fils retenus. Contraint le nombre d'éléments de matrice et les codes de phase pouvant être utilisés avec cette matrice.
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 220 montre toutes les extrémités d'association de PerLengthPhaseImpedance avec les autres classes.

Tableau 220 – Extrémités d'association de Wires::PerLengthPhaseImpedance avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[1..1]	[1..*] PhaseImpedanceData	PhaseImpedanceData	Toutes les données qui appartiennent à cette impédance de phase de conducteur.
[0..1]	[0..*] ACLineSegments	ACLineSegment	hérité de: PerLengthImpedance
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.31 PerLengthSequenceImpedance

Paramètres linéiques d'impédance et d'admittance en séquence, pour les lignes transposées monophasées, diphasées ou triphasées. Pour les lignes monophasées, définir $x=x_0=x_{self}$. Pour les lignes diphasées, définir $x=x_s-x_m$ et $x_0=x_s+x_m$.

Le Tableau 221 montre tous les attributs de PerLengthSequenceImpedance.

Tableau 221 – Attributs de Wires::PerLengthSequenceImpedance

nom	type	description
b0ch	SusceptancePerLength	Susceptance shunt homopolaire (chargement), par unité de longueur.
bch	SusceptancePerLength	Susceptance shunt directe (chargement), par unité de longueur.
g0ch	ConductancePerLength	Conductance shunt homopolaire (chargement), par unité de longueur.
gch	ConductancePerLength	Conductance shunt directe (chargement), par unité de longueur.
r	ResistancePerLength	Résistance série directe, par unité de longueur.
r0	ResistancePerLength	Résistance série homopolaire, par unité de longueur.
x	ReactancePerLength	Réactance série directe, par unité de longueur.
x0	ReactancePerLength	Réactance série homopolaire, par unité de longueur.
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 222 montre toutes les extrémités d'association de PerLengthSequenceImpedance avec les autres classes.

Tableau 222 – Extrémités d'association de Wires::PerLengthSequenceImpedance avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[0..*] ACLineSegments	ACLineSegment	hérité de: PerLengthImpedance
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.32 Classe racine PhaseImpedanceData

Triplet de valeurs d'éléments des matrices des résistances, des réactances et des susceptances.

Le Tableau 223 montre tous les attributs de PhaseImpedanceData.

Tableau 223 – Attributs de Wires::PhaseImpedanceData

nom	type	description
sequenceNumber	Integer	Indice d'élément en colonne, en supposant une matrice symétrique. Plages de 1 à $N + N*(N-1)/2$.
b	SusceptancePerLength	Valeur d'élément de la matrice des susceptances, par unité de longueur.
r	ResistancePerLength	Valeur d'élément de la matrice des résistances, par unité de longueur.
x	ReactancePerLength	Valeur d'élément de la matrice des réactances, par unité de longueur.

Le Tableau 224 montre toutes les extrémités d'association de PhaseImpedanceData avec les autres classes.

Tableau 224 – Extrémités d'association de Wires::PhaseImpedanceData avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[1..*]	[1..1] PhaseImpedance	PerLengthPhaseImpedance	Impédance de phase de conducteur à laquelle ces données appartiennent.

6.8.33 Enumération PhaseShuntConnectionKind

La configuration des connexions de phase pour un appareil à une seule borne comme une charge ou un condensateur.

Le Tableau 225 montre tous les libellés de PhaseShuntConnectionKind.

Tableau 225 – Libellés de Wires::PhaseShuntConnectionKind

libellé	description
D	Montage en triangle.
Y	Montage en étoile.
Yn	Wye (c'est-à-dire I grec, montage étoile), avec neutre sorti pour la mise à la terre.
I	Enroulement indépendant, pour connexions monophasées.

6.8.34 PhaseTapChanger

Un modèle de changeur de prise déphaseur pour transformateur, lequel commande la différence d'angle de phase à travers le transformateur de puissance et potentiellement le flux de puissance activé à travers le transformateur de puissance. Ce modèle de prise pour phase peut aussi avoir un effet sur la valeur de la tension.

Le Tableau 226 montre tous les attributs de PhaseTapChanger.

Tableau 226 – Attributs de Wires::PhaseTapChanger

nom	type	description
highStep	Integer	hérité de: TapChanger
initialDelay	Seconds	hérité de: TapChanger
lowStep	Integer	hérité de: TapChanger
lfcFlag	Boolean	hérité de: TapChanger
neutralStep	Integer	hérité de: TapChanger
neutralU	Voltage	hérité de: TapChanger
normalStep	Integer	hérité de: TapChanger
regulationStatus	Boolean	hérité de: TapChanger
subsequentDelay	Seconds	hérité de: TapChanger
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 227 montre toutes les extrémités d'association de PhaseTapChanger avec d'autres classes.

Tableau 227 – Extrémités d'association de Wires::PhaseTapChanger avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[1..1] TransformerEnd	TransformerEnd	Extrémité de transformateur à laquelle appartient ce changeur de prise pour phase.
[0..*]	[0..1] PhaseTapChangerTabular	PhaseTapChangerTabular	Le tableau du changeur de prise pour phase pour ce changeur de prise déphaseur.
[0..*]	[0..1] TapChangerControl	TapChangerControl	hérité de: TapChanger
[1..1]	[0..1] SvTapStep	SvTapStep	hérité de: TapChanger
[1..1]	[0..*] TapSchedules	TapSchedule	hérité de: TapChanger
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.35 PhaseTapChangerAsymetrical

Décrit le modèle de prise pour un transformateur déphaseur asymétrique dans lequel le vecteur de tension de différence s'ajoute à la tension côté primaire. L'angle entre la tension côté primaire et la tension de différence est appelé angle de connexion d'enroulement. Le déphasage dépend à la fois de la valeur de la tension de différence et de l'angle de connexion d'enroulement.

Le Tableau 228 montre tous les attributs de PhaseTapChangerAsymetrical.

Tableau 228 – Attributs de Wires::PhaseTapChangerAsymetrical

nom	type	description
windingConnectionAngle	AngleDegrees	L'angle de phase entre l'enroulement en phase et l'enroulement en discordance de phases utilisé pour créer le déphasage. L'enroulement en discordance de phases produit ce qu'on appelle la tension de différence. Attribuer 90 degrés à cet angle n'est pas identique à un transformateur symétrique.
voltageStepIncrement	PerCent	hérité de: PhaseTapChangerNonLinear
xMax	Reactance	hérité de: PhaseTapChangerNonLinear
xMedian	Reactance	hérité de: PhaseTapChangerNonLinear
highStep	Integer	hérité de: TapChanger
initialDelay	Seconds	hérité de: TapChanger
lowStep	Integer	hérité de: TapChanger
ltoFlag	Boolean	hérité de: TapChanger
neutralStep	Integer	hérité de: TapChanger
neutralU	Voltage	hérité de: TapChanger
normalStep	Integer	hérité de: TapChanger
regulationStatus	Boolean	hérité de: TapChanger
subsequentDelay	Seconds	hérité de: TapChanger
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 229 montre toutes les extrémités d'association de PhaseTapChangerAsymetrical avec d'autres classes.

Tableau 229 – Extrémités d'association de Wires::PhaseTapChangerAsymetrical avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[1..1] TransformerEnd	TransformerEnd	hérité de: PhaseTapChanger
[0..*]	[0..1] PhaseTapChangerTabular	PhaseTapChangerTabular	hérité de: PhaseTapChanger
[0..*]	[0..1] TapChangerControl	TapChangerControl	hérité de: TapChanger
[1..1]	[0..1] SvTapStep	SvTapStep	hérité de: TapChanger
[1..1]	[0..*] TapSchedules	TapSchedule	hérité de: TapChanger
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.36 PhaseTapChangerLinear

Décrit un changeur de prise dont la relation est linéaire entre l'échelon de prise et la différence d'angle de phase à travers le transformateur. C'est un modèle mathématique qui est une approximation pour un changeur de prise pour phase réelle.

Le Tableau 230 montre tous les attributs de PhaseTapChangerLinear.

Tableau 230 – Attributs de Wires::PhaseTapChangerLinear

nom	type	description
stepPhaseShiftIncrement	AngleDegrees	Déphasage par position de régulateur. Une valeur positive indique un déphasage positif depuis l'enroulement où la prise est située jusqu'à l'autre enroulement (pour un transformateur à deux enroulements). L'incrément de déphasage réel pourrait être calculé plus précisément à partir des modèles symétriques ou asymétriques ou à partir d'une consultation de la table des échelons de prise si ceux-ci sont disponibles.
highStep	Integer	hérité de: TapChanger
initialDelay	Seconds	hérité de: TapChanger
lowStep	Integer	hérité de: TapChanger
lfcFlag	Boolean	hérité de: TapChanger
neutralStep	Integer	hérité de: TapChanger
neutralU	Voltage	hérité de: TapChanger
normalStep	Integer	hérité de: TapChanger
regulationStatus	Boolean	hérité de: TapChanger
subsequentDelay	Seconds	hérité de: TapChanger
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 231 montre toutes les extrémités d'association de PhaseTapChangerLinear avec d'autres classes.

Tableau 231 – Extrémités d'association de Wires::PhaseTapChangerLinear avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[1..1] TransformerEnd	TransformerEnd	hérité de: PhaseTapChanger
[0..*]	[0..1] PhaseTapChangerTabular	PhaseTapChangerTabular	hérité de: PhaseTapChanger
[0..*]	[0..1] TapChangerControl	TapChangerControl	hérité de: TapChanger
[1..1]	[0..1] SvTapStep	SvTapStep	hérité de: TapChanger
[1..1]	[0..*] TapSchedules	TapSchedule	hérité de: TapChanger
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.37 PhaseTapChangerNonLinear

Le changeur de prise pour phase non linéaire décrit le comportement non linéaire d'un changeur de prise pour phase. C'est une classe de base pour les modèles de changeur de prise pour phase symétrique et asymétrique. Ces modèles sont présentés en détail dans la CEI 61970-301.

Le Tableau 232 montre tous les attributs de PhaseTapChangerNonLinear.

Tableau 232 – Attributs de Wires::PhaseTapChangerNonLinear

nom	type	description
voltageStepIncrement	Percent	Augmentation d'un pas de la tension sur l'enroulement en discordance de phases, spécifié en pourcentage de la tension nominale à l'extrémité du transformateur.
xMax	Reactance	La réactance à l'échelon de prise maximal.
xMedian	Reactance	La réactance à l'échelon de prise moyen.
highStep	Integer	hérité de: TapChanger
initialDelay	Seconds	hérité de: TapChanger
lowStep	Integer	hérité de: TapChanger
lfcFlag	Boolean	hérité de: TapChanger
neutralStep	Integer	hérité de: TapChanger
neutralU	Voltage	hérité de: TapChanger
normalStep	Integer	hérité de: TapChanger
regulationStatus	Boolean	hérité de: TapChanger
subsequentDelay	Seconds	hérité de: TapChanger
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 233 montre toutes les extrémités d'association de PhaseTapChangerNonLinear avec d'autres classes.

Tableau 233 – Extrémités d'association de Wires::PhaseTapChangerNonLinear avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[1..1] TransformerEnd	TransformerEnd	hérité de: PhaseTapChanger
[0..*]	[0..1] PhaseTapChangerTabular	PhaseTapChangerTabular	hérité de: PhaseTapChanger
[0..*]	[0..1] TapChangerControl	TapChangerControl	hérité de: TapChanger
[1..1]	[0..1] SvTapStep	SvTapStep	hérité de: TapChanger
[1..1]	[0..*] TapSchedules	TapSchedule	hérité de: TapChanger
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.38 PhaseTapChangerSymetrical

Décrit un modèle de prise de transformateur déphaseur symétrique dans lequel la valeur de tension du côté secondaire est la même que du côté primaire. La valeur de la tension de différence est la base d'un triangle équilatéral où les côtés correspondent aux tensions primaire et secondaire. La différence d'angle de phase correspond à l'angle supérieur et peut être exprimée comme le double de l'arc tangente de la moitié de la tension totale de différence.

Le Tableau 234 montre tous les attributs de PhaseTapChangerSymetrical.

Tableau 234 – Attributs de Wires::PhaseTapChangerSymetrical

nom	type	description
voltageStepIncrement	PerCent	hérité de: PhaseTapChangerNonLinear
xMax	Reactance	hérité de: PhaseTapChangerNonLinear
xMedian	Reactance	hérité de: PhaseTapChangerNonLinear
highStep	Integer	hérité de: TapChanger
initialDelay	Seconds	hérité de: TapChanger
lowStep	Integer	hérité de: TapChanger
ltcFlag	Boolean	hérité de: TapChanger
neutralStep	Integer	hérité de: TapChanger
neutralU	Tension	hérité de: TapChanger
normalStep	Integer	hérité de: TapChanger
regulationStatus	Boolean	hérité de: TapChanger
subsequentDelay	Seconds	hérité de: TapChanger
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
nom	String	hérité de: IdentifiedObject

Le Tableau 235 montre toutes les extrémités d'association de PhaseTapChangerSymetrical avec d'autres classes.

Tableau 235 – Extrémités d'association de Wires::PhaseTapChangerSymetrical avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[1..1] TransformerEnd	TransformerEnd	hérité de: PhaseTapChanger
[0..*]	[0..1] PhaseTapChangerTabular	PhaseTapChangerTabular	hérité de: PhaseTapChanger
[0..*]	[0..1] TapChangerControl	TapChangerControl	hérité de: TapChanger
[1..1]	[0..1] SvTapStep	SvTapStep	hérité de: TapChanger
[1..1]	[0..*] TapSchedules	TapSchedule	hérité de: TapChanger
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.39 PhaseTapChangerTabular

Décrit une courbe tabulaire pour la manière dont la différence d'angle de phase et l'impédance varient avec l'échelon de prise.

Le Tableau 236 montre tous les attributs de PhaseTapChangerTabular.

Tableau 236 – Attributs de Wires::PhaseTapChangerTabular

nom	type	description
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 237 montre toutes les extrémités d'association de PhaseTapChangerTabular avec d'autres classes.

Tableau 237 – Extrémités d'association de Wires::PhaseTapChangerTabular avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[0..*] PhaseTapChanger	PhaseTapChanger	Les changeurs de prise déphaseur auxquels s'applique ce tableau de prise pour phase.
[1..1]	[1..*] PhaseTapChangerTabularPoint	PhaseTapChangerTabularPoint	Les points de ce tableau.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.40 Classe racine PhaseTapChangerTabularPoint

Décrit chaque échelon de prise dans la courbe tabulaire du changeur de prise pour phase.

Le Tableau 238 montre tous les attributs de PhaseTapChangerTabularPoint.

Tableau 238 – Attributs de Wires::PhaseTapChangerTabularPoint

nom	type	description
angle	AngleDegrees	La différence d'angle en degrés.
b	PerCent	Ecart de susceptance de la branche magnétisante, en pourcentage de la valeur nominale. La susceptance réelle est calculée de la manière suivante: susceptance magnétisante calculée = $b(\text{nominale}) * (1 + b(\text{de cette classe})/100)$. Le terme $b(\text{nominale})$ est défini comme la susceptance magnétisante statique à l'extrémité ou aux extrémités du transformateur de puissance associé. Ce modèle suppose que l'impédance a une forme en étoile (modèle pi).
g	PerCent	Ecart de conductance de la branche magnétisante, en pourcentage de la valeur nominale. La conductance réelle est calculée de la manière suivante: conductance magnétisante calculée = $g(\text{nominale}) * (1 + g(\text{de cette classe})/100)$. Le terme $g(\text{nominale})$ est défini comme la conductance magnétisante statique à l'extrémité ou aux extrémités du transformateur de puissance associé. Ce modèle suppose que l'impédance a une forme en étoile (modèle pi).
r	PerCent	L'écart de résistance, en pourcentage de la valeur nominale. La résistance réelle est calculée de la manière suivante: résistance calculée = $r(\text{nominale}) * (1 + r(\text{de cette classe})/100)$. Le terme $r(\text{nominale})$ est défini comme la résistance statique à l'extrémité ou aux extrémités du transformateur de puissance associé. Ce modèle suppose que l'impédance a une forme en étoile (modèle pi).
step	Integer	L'échelon de prise.
x	PerCent	L'écart de réactance série, en pourcentage de la valeur nominale. La réactance réelle est calculée de la manière suivante: réactance calculée = $x(\text{nominale}) * (1 + x(\text{de cette classe})/100)$. Le terme $x(\text{nominale})$ est défini comme la réactance série statique à l'extrémité ou aux extrémités du transformateur de puissance associé. Ce modèle suppose que l'impédance a une forme en étoile (modèle pi).

Le Tableau 239 montre toutes les extrémités d'association de PhaseTapChangerTabularPoint avec d'autres classes.

Tableau 239 – Extrémités d'association de Wires::PhaseTapChangerTabularPoint avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[1..*]	[1..1] PhaseTapChangerTabular	PhaseTapChangerTabular	Le tableau de ce point.

6.8.41 Plant

Une Plant (une centrale) est une collection d'équipements utilisés à des fins de production.

Le Tableau 240 montre tous les attributs de Plant.

Tableau 240 – Attributs de Wires::Plant

nom	type	description
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 241 montre toutes les extrémités d'association de Plant avec d'autres classes.

Tableau 241 – Extrémités d'association de Wires::Plant avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[0..*] Equipments	Equipment	hérité de: EquipmentContainer
[1..1]	[0..*] ConnectivityNodes	ConnectivityNode	hérité de: ConnectivityNodeContainer
[0..1]	[0..*] TopologicalNode	TopologicalNode	hérité de: ConnectivityNodeContainer
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.42 PowerTransformer

Appareil électrique constitué de deux enroulements couplés ou plus, avec ou sans cœur magnétique, destiné à introduire un couplage mutuel entre les circuits électriques. Les transformateurs peuvent être utilisés pour contrôler la tension et le déplacement de phase (flux de puissance active).

Un transformateur de puissance peut être composé de cuves distinctes de transformateur qui ne sont pas nécessairement identiques.

Un transformateur de puissance peut être modélisé avec ou sans cuve, il est prévu pour être utilisé à la fois dans une représentation équilibrée et une représentation déséquilibrée. Un transformateur de puissance a typiquement deux bornes, mais il peut en avoir une (terre), trois ou davantage.

Le Tableau 242 montre tous les attributs de PowerTransformer.

Tableau 242 – Attributs de Wires::PowerTransformer

nom	type	description
vectorGroup	String	Groupe vectoriel du transformateur pour le relayage protecteur, par exemple Dyn1. Pour les transformateurs déséquilibrés, cela peut ne pas être déterminé simplement à partir des connexions d'enroulements constitutives et des déplacements d'angle de phase. La chaîne vectorGroup comporte les composants suivants dans l'ordre de l'énumération: connexion d'enroulement haute tension, connexion d'enroulement moyenne tension (pour des transformateurs à trois enroulements), numéro de position de déphasage de 0 à 11, connexion d'enroulement basse tension numéro de position de déphasage de 0 à 11 (position d'horloge). Les connexions d'enroulement sont D (triangle), Y (étoile), YN (étoile avec neutre), Z (zig-zag), ZN (zig-zag avec neutre), A (autotransformateur). Les majuscules indiquent la haute tension, les minuscules la basse ou moyenne tension. L'enroulement haute tension a toujours pour position 0, il n'est pas compris dans la chaîne de groupe vectoriel. Quelques exemples: YNy0 (deux enroulements étoile à étoile, sans déphasage), YNd11 (deux enroulements étoile à triangle avec un déphasage de 330 degrés), YNyn0d5 (trois enroulements transformateurs étoile à haute tension neutre, étoile à moyenne tension neutre sans déphasage, triangle basse tension à déphasage de 150 degrés). Le déphasage est défini comme l'écart angulaire entre les phaseurs représentant les tensions entre le point neutre (réel ou fictif) et les bornes homologues de deux enroulements lorsqu'un système de tension direct est appliqué aux bornes de l'enroulement à haute tension dans l'ordre de séquence alphabétique de ces bornes si elles sont repérées par des lettres, ou dans leur ordre de séquence numérique si elles sont repérées par des chiffres: les phaseurs sont supposés tourner en sens inverse des aiguilles d'une horloge.
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 243 montre toutes les extrémités d'association de PowerTransformer avec d'autres classes.

Tableau 243 – Extrémités d'association de Wires::PowerTransformer avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[0..*] PowerTransformerEnd	PowerTransformerEnd	Les extrémités de ce transformateur de puissance.
[0..1]	[0..*] TransformerTanks	TransformerTank	Tous les transformateurs qui appartiennent à cette batterie.
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.43 PowerTransformerEnd

Un PowerTransformerEnd est associé à chaque Terminal d'un PowerTransformer.

Les valeurs d'impédance r , $r0$, x et $x0$ d'un PowerTransformerEnd représentent un équivalent étoile de la manière suivante

- pour un PowerTransformer à deux Terminal (bornes), le PowerTransformerEnd haute tension a des valeurs non nulles pour r , $r0$, x et $x0$ tandis que le PowerTransformerEnd basse tension a des valeurs nulles pour r , $r0$, x et $x0$.
- pour un PowerTransformer à trois Terminal (bornes), les trois PowerTransformerEnd représentent un équivalent étoile dont chaque branche de l'étoile est représentée par r , $r0$, x et des valeurs $x0$.
- pour un PowerTransformer à plus de trois Terminal (bornes), il n'est pas possible d'utiliser les valeurs d'impédance de PowerTransformerEnd. Utiliser dans ce cas TransformerMeshImpedance ou diviser le transformateur en plusieurs PowerTransformers.

Le Tableau 244 montre tous les attributs de PowerTransformerEnd.

Tableau 244 – Attributs de Wires::PowerTransformerEnd

nom	type	description
b	Susceptance	Susceptance de branche magnétique (B mag). La valeur peut être positive ou négative.
b0	Susceptance	Susceptance de branche magnétique homopolaire.
connectionKind	WindingConnection	Sorte de connexion.
g	Conductance	Conductance de la branche magnétisante.
g0	Conductance	Conductance homopolaire de la branche magnétisante (modèle en étoile).
phaseAngleClock	Integer	Déplacement de l'angle de phase de tension aux bornes où les heures d'une horloge représentent les 360 degrés. Les valeurs valides vont de 0 à 11. Par exemple pour l'extrémité côté secondaire d'un transformateur dont le code de groupe vectoriel est 'Dyn11', spécifier le type de connexion sous forme étoile avec neutre et spécifier l'angle de phase 11 d'après l'horloge. La valeur de position pour le numéro d'extrémité spécifié 1 du transformateur est supposée être zéro. Noter que le numéro d'extrémité du transformateur n'est pas supposé être identique au numéro d'ordre de la borne.
r	Resistance	Résistance (modèle en étoile) de l'extrémité du transformateur.
r0	Resistance	Résistance série homopolaire (modèle en étoile) de l'extrémité du transformateur.
ratedS	ApparentPower	Caractéristique nominale de puissance apparente normale.
ratedU	Voltage	Tension assignée: entre phases pour les enroulements triphasés et soit entre phases, soit entre phase et neutre pour les enroulements monophasés.
x	Reactance	Réactance série directe (modèle en étoile) de l'extrémité du transformateur.
x0	Reactance	Réactance série homopolaire de l'extrémité du transformateur.
bmagSat	PerCent	hérité de: TransformerEnd
endNumber	Integer	hérité de: TransformerEnd
grounded	Boolean	hérité de: TransformerEnd
magBaseU	Voltage	hérité de: TransformerEnd
magSatFlux	PerCent	hérité de: TransformerEnd
rground	Resistance	hérité de: TransformerEnd
xground	Reactance	hérité de: TransformerEnd
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 245 montre toutes les extrémités d'association de PowerTransformerEnd avec d'autres classes.

Tableau 245 – Extrémités d'association de Wires::PowerTransformerEnd avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] PowerTransformer	PowerTransformer	Le transformateur de puissance de cette extrémité de transformateur de puissance.
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: TransformerEnd
[0..*]	[0..1] Terminal	Terminal	hérité de: TransformerEnd
[1..1]	[0..1] PhaseTapChanger	PhaseTapChanger	hérité de: TransformerEnd
[1..1]	[0..1] RatioTapChanger	RatioTapChanger	hérité de: TransformerEnd
[0..*]	[0..1] StarImpedance	TransformerStarImpedance	hérité de: TransformerEnd
[0..*]	[0..1] CoreAdmittance	TransformerCoreAdmittance	hérité de: TransformerEnd
[1..1]	[0..*] FromMeshImpedance	TransformerMeshImpedance	hérité de: TransformerEnd
[1..*]	[0..*] ToMeshImpedance	TransformerMeshImpedance	hérité de: TransformerEnd
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.44 ProtectedSwitch

Un ProtectedSwitch (un commutateur protégé) est un dispositif de commutation qui peut être actionné par un ProtectionEquipment (un équipement de protection).

Le Tableau 246 montre tous les attributs de ProtectedSwitch.

Tableau 246 – Attributs de Wires::ProtectedSwitch

nom	type	description
breakingCapacity	CurrentFlow	Courant de défaut maximal auquel un dispositif de coupure peut fonctionner en toute sécurité dans les conditions d'utilisation spécifiées.
normalOpen	Boolean	hérité de: Switch
ratedCurrent	CurrentFlow	hérité de: Switch
switchOnCount	Integer	hérité de: Switch
switchOnDate	DateTime	hérité de: Switch
retained	Boolean	hérité de: Switch
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 247 montre toutes les extrémités d'association de ProtectedSwitch avec d'autres classes.

**Tableau 247 – Extrémités d'association de Wires::ProtectedSwitch
avec d'autres classes**

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..*] OperatedByProtectionEquipment	ProtectionEquipment	Les équipements de protection qui font fonctionner ce ProtectedSwitch.
[1..1]	[0..*] RecloseSequences	RecloseSequence	Un disjoncteur peut avoir aucun, un ou plusieurs réenclenchements automatiques après un déclenchement.
[1..1]	[0..*] SwitchSchedules	SwitchSchedule	hérité de: Switch
[1..1]	[0..*] SwitchPhase	SwitchPhase	hérité de: Switch
[0..*]	[0..1] CompositeSwitch	CompositeSwitch	hérité de: Switch
[0..*]	[0..*] SwitchingOperations	SwitchingOperation	hérité de: Switch
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.45 RatioTapChanger

Un changeur de prise qui change le rapport de tensions ayant un impact sur la grandeur de la tension, mais pas sur l'angle de phase à travers le transformateur.

Le Tableau 248 montre tous les attributs de RatioTapChanger.

Tableau 248 – Attributs de Wires::RatioTapChanger

nom	type	description
stepVoltageIncrement	PerCent	Augmentation d'un pas de prise du changeur, exprimée en pourcentage de la tension nominale, par position de changeur.
tcuControlMode	TransformerControlMode	Spécifie le mode de commande de régulation (tension ou puissance réactive) du RatioTapChanger.
highStep	Integer	hérité de: TapChanger
initialDelay	Seconds	hérité de: TapChanger
lowStep	Integer	hérité de: TapChanger
ltcFlag	Boolean	hérité de: TapChanger
neutralStep	Integer	hérité de: TapChanger
neutralU	Voltage	hérité de: TapChanger
normalStep	Integer	hérité de: TapChanger
regulationStatus	Boolean	hérité de: TapChanger
subsequentDelay	Seconds	hérité de: TapChanger
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 249 montre toutes les extrémités d'association de RatioTapChanger avec d'autres classes.

Tableau 249 – Extrémités d'association de Wires::RatioTapChanger avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] RatioTapChangerTabular	RatioTapChangerTabular	Le tableau de rapport de prise pour ce changeur de rapport de prise.
[0..1]	[1..1] TransformerEnd	TransformerEnd	Extrémité de transformateur à laquelle appartient ce changeur de rapport de prise.
[0..*]	[0..1] TapChangerControl	TapChangerControl	hérité de: TapChanger
[1..1]	[0..1] SvTapStep	SvTapStep	hérité de: TapChanger
[1..1]	[0..*] TapSchedules	TapSchedule	hérité de: TapChanger
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.46 RatioTapChangerTabular

Décrit une courbe qui montre comment la valeur de tension et l'impédance varient avec l'échelon de prise.

Le Tableau 250 montre tous les attributs de RatioTapChangerTabular.

Tableau 250 – Attributs de Wires::RatioTapChangerTabular

nom	type	description
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 251 montre toutes les extrémités d'association de RatioTapChangerTabular avec d'autres classes.

Tableau 251 – Extrémités d'association de Wires::RatioTapChangerTabular avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[0..*] RatioTapChanger	RatioTapChanger	Le changeur de rapport de prise de ce tableau de rapport de prise.
[1..1]	[1..*] RatioTapChangerTabularPoint	RatioTapChangerTabularPoint	Points de ce tableau.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.47 Classe racine RatioTapChangerTabularPoint

Décrit chaque échelon de prise dans la courbe tabulaire de changeur de rapport de prise.

Le Tableau 252 montre tous les attributs de RatioTapChangerTabularPoint.

Tableau 252 – Attributs de Wires::RatioTapChangerTabularPoint

nom	type	description
b	PerCent	Ecart de susceptance de la branche magnétisante, en pourcentage de la valeur nominale. La susceptance réelle est calculée de la manière suivante: susceptance magnétisante calculée = $b(\text{nominale}) * (1 + b(\text{de cette classe})/100)$. Le terme $b(\text{nominale})$ est défini comme la susceptance magnétisante statique à l'extrémité ou aux extrémités du transformateur de puissance associé. Ce modèle suppose que l'impédance a une forme en étoile (modèle pi).
g	PerCent	Ecart de conductance de la branche magnétisante, en pourcentage de la valeur nominale. La conductance réelle est calculée de la manière suivante: conductance magnétisante calculée = $g(\text{nominale}) * (1 + g(\text{de cette classe})/100)$. Le terme $g(\text{nominale})$ est défini comme la conductance magnétisante statique à l'extrémité ou aux extrémités du transformateur de puissance associé. Ce modèle suppose que l'impédance a une forme en étoile (modèle pi).

nom	type	description
r	PerCent	L'écart de résistance, en pourcentage de la valeur nominale. La réactance réelle est calculée de la manière suivante: résistance calculée = $r(\text{nominale}) * (1 + r(\text{de cette classe})/100)$. Le terme $r(\text{nominale})$ est défini comme la résistance statique à l'extrémité ou aux extrémités du transformateur de puissance associé. Ce modèle suppose que l'impédance a une forme en étoile (modèle pi).
rapport	Float	Le rapport de tension par unité. Cette valeur est donc proche de un.
step	Integer	L'échelon de prise.
x	PerCent	L'écart de réactance série, en pourcentage de la valeur nominale. La réactance réelle est calculée de la manière suivante: réactance calculée = $x(\text{nominale}) * (1 + x(\text{de cette classe})/100)$. Le terme $x(\text{nominale})$ est défini comme la réactance série statique à l'extrémité ou aux extrémités du transformateur de puissance associé. Ce modèle suppose que l'impédance a une forme en étoile (modèle pi).

Le Tableau 253 montre toutes les extrémités d'association de RatioTapChangerTabularPoint avec d'autres classes.

Tableau 253 – Extrémités d'association de Wires::RatioTapChangerTabularPoint avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[1..*]	[1..1] RatioTapChangerTabular	RatioTapChangerTabular	Tableau de ce point.

6.8.48 ReactiveCapabilityCurve

Courbe enveloppe assignée de puissance réactive en fonction de la puissance active de la machine synchrone, en mode générateur et moteur. Il existe, pour chaque valeur de puissance active, une valeur limite supérieure et inférieure correspondante de la puissance réactive. Traditionnellement, il y aura une courbe distincte pour chaque état du fluide de refroidissement, tel que la pression de l'hydrogène. Les valeurs de l'axe Y1 représentent le minimum de la puissance réactive alors que les valeurs de l'axe Y2 représentent le maximum de la puissance réactive.

Le Tableau 254 montre tous les attributs de ReactiveCapabilityCurve.

Tableau 254 – Attributs de Wires::ReactiveCapabilityCurve

nom	type	description
coolantTemperature	Température	Température du fluide de refroidissement de la machine (par exemple l'air ambiant ou l'eau de circulation du stator).
hydrogenPressure	Pression	Pression du fluide de refroidissement à base d'hydrogène
curveStyle	CurveStyle	hérité de: Curve
xMultiplier	UnitMultiplier	hérité de: Curve
xUnit	UnitSymbol	hérité de: Curve
y1Multiplier	UnitMultiplier	hérité de: Curve
y1Unit	UnitSymbol	hérité de: Curve
y2Multiplier	UnitMultiplier	hérité de: Curve
y2Unit	UnitSymbol	hérité de: Curve
y3Multiplier	UnitMultiplier	hérité de: Curve
y3Unit	UnitSymbol	hérité de: Curve
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 255 montre toutes les extrémités d'association de ReactiveCapabilityCurve avec d'autres classes.

Tableau 255 – Extrémités d'association de Wires::ReactiveCapabilityCurve avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[1..*] InitiallyUsedBySynchronousMachines	SynchronousMachine	Machines synchrones utilisant cette courbe comme courbe par défaut.
[0..*]	[1..*] SynchronousMachines	SynchronousMachine	Machines synchrones utilisant cette courbe.
[1..1]	[0..*] CurveDatas	CurveData	hérité de: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.49 Recloser

Interrupteur de défaut monté sur poteau, à relais intégrés de phase et de mise à la terre, transformateur de courant (CT) et commandes supplémentaires.

Le Tableau 256 montre tous les attributs de Recloser.

Tableau 256 – Attributs de Wires::Recloser

nom	type	description
breakingCapacity	CurrentFlow	hérité de: ProtectedSwitch
normalOpen	Boolean	hérité de: Switch
ratedCurrent	CurrentFlow	hérité de: Switch
switchOnCount	Integer	hérité de: Switch
switchOnDate	DateTime	hérité de: Switch
retained	Boolean	hérité de: Switch
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 257 montre toutes les extrémités d'association de Recloser avec d'autres classes.

Tableau 257 – Extrémités d'association de Wires::Recloser avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..*] OperatedByProtectionEquipment	ProtectionEquipment	hérité de: ProtectedSwitch
[1..1]	[0..*] RecloseSequences	RecloseSequence	hérité de: ProtectedSwitch
[1..1]	[0..*] SwitchSchedules	SwitchSchedule	hérité de: Switch
[1..1]	[0..*] SwitchPhase	SwitchPhase	hérité de: Switch
[0..*]	[0..1] CompositeSwitch	CompositeSwitch	hérité de: Switch
[0..*]	[0..*] SwitchingOperations	SwitchingOperation	hérité de: Switch
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.50 RectifierInverter

Équipement de conversion alternatif/continu bidirectionnel pouvant être utilisé pour contrôler le courant continu, la tension continue, le flux d'énergie continue ou l'angle d'amorçage.

Le Tableau 258 montre tous les attributs de RectifierInverter.

Tableau 258 – Attributs de Wires::RectifierInverter

nom	type	description
bridges	Integer	Nombre de ponts
commutatingReactance	Reactance	Réactance de commutation, sur la fréquence du bus alternatif.
commutatingResistance	Resistance	Résistance de commutation.
compoundResistance	Resistance	Résistance compound.
fréquence	Fréquence	Fréquence du côté CA (courant alternatif).
maxP	ActivePower	La puissance active maximale du côté CC à laquelle il convient que le convertisseur fonctionne.
maxU	Voltage	La tension maximale du côté CC à laquelle il convient que le convertisseur fonctionne.
minCompoundVoltage	Voltage	Tension continue compoundée minimale.
minP	ActivePower	La puissance active minimale du côté CC à laquelle il convient que le convertisseur fonctionne.
minU	Voltage	La tension minimale du côté CC à laquelle il convient que le convertisseur fonctionne.
operatingMode	OperatingMode	Mode de fonctionnement du convertisseur.
ratedU	Voltage	Tension de base primaire du redresseur/onduleur.
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 259 montre toutes les extrémités d'association de RectifierInverter avec d'autres classes.

Tableau 259 – Extrémités d'association de Wires::RectifierInverter avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.51 RegulatingCondEq

Un type d'équipement conducteur qui peut réguler une grandeur (tension ou flux) en un point spécifique du réseau.

Le Tableau 260 montre tous les attributs de RegulatingCondEq.

Tableau 260 – Attributs de Wires::RegulatingCondEq

nom	type	description
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 261 montre toutes les extrémités d'association de RegulatingCondEq avec d'autres classes.

**Tableau 261 – Extrémités d'association de Wires::RegulatingCondEq
avec d'autres classes**

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] RegulatingControl	RegulatingControl	Le plan de commande de régulation auquel cet équipement participe.
[0..1]	[1..*] Controls	Control	Les sorties de commande utilisées pour commander effectivement un dispositif de régulation, par exemple la magnétisation d'une machine synchrone ou l'actionneur disjoncteur d'une batterie de condensateurs.
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.52 RegulatingControl

Spécifie un ensemble d'équipements qui fonctionnent ensemble pour réguler une grandeur de système électrique telle que la tension ou le flux.

Le Tableau 262 montre tous les attributs de RegulatingControl.

Tableau 262 – Attributs de Wires::RegulatingControl

nom	type	description
mode	RegulatingControlModeKind	Le mode de commande de régulation actuellement disponible. La présente spécification permet de déterminer le genre de régulation sans la nécessité d'obtenir les unités à partir d'un programme.
targetRange	Float	Il s'agit de la plage cible d'entrée de cas. Elle exécute la même fonction que l'attribut value2 sur le programme de régulation dans le cas où les programmes ne sont pas utilisés. Les unités pour ceux qui sont appropriés pour ce mode.
targetValue	Float	La valeur cible spécifiée pour "case input". Cette valeur peut être utilisée pour la valeur cible sans utilisation de programmes. Cette valeur a les unités appropriées à l'attribut de mode.
discrete	Boolean	La régulation est effectuée dans un mode discret.
monitoredPhase	PhaseCode	Tension de phase commandant ce régulateur, mesurée à l'emplacement du régulateur.
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 263 montre toutes les extrémités d'association de RegulatingControl avec d'autres classes.

Tableau 263 – Extrémités d'association de Wires::RegulatingControl avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] Terminal	Terminal	La borne associée à cette commande de régulation. La borne est associée au lieu d'un nœud, puisque la borne pourrait être connectée soit à un nœud topologique (bus dans un modèle de topologie nodale) ou à un nœud de connectivité (modèle de détail de commutateur). Il est quelquefois utile de modéliser la régulation à une borne d'un objet jeu de barres puisque le jeu de barres peut se trouver à la fois dans un modèle de topologie nodale et un modèle de détail de commutateur.
[0..1]	[0..*] RegulatingCondEq	RegulatingCondEq	L'équipement qui participe à ce plan de commande de régulation.
[1..1]	[0..*] RegulationSchedule	RegulationSchedule	Programme pour cette commande de régulation.
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.53 Enumération RegulatingControlModeKind

Le genre de modèle de régulation. Par exemple: tension de régulation, puissance réactive, puissance active, etc.

Le Tableau 264 montre tous les libellés de RegulatingControlModeKind.

Tableau 264 – Libellés de Wires::RegulatingControlModeKind

libellé	description
voltage	La tension est spécifiée.
activePower	La puissance active est spécifiée.
reactivePower	La puissance réactive est spécifiée.
currentFlow	L'intensité est spécifiée.
balise	Le mode de régulation est fixe et donc ne régule pas.
admittance	L'admittance est spécifiée.
timeScheduled	La commande s'active/se désactive par heure du jour. Les heures peuvent varier à la fin de semaine ou dans des saisons différentes.
temperature	La commande s'active/se désactive en fonction de la température locale (c'est-à-dire un thermostat).
powerFactor	Le facteur de puissance est spécifié.

6.8.54 RegulationSchedule

Modèle prédéfini sur une longue durée pour une variable commandée, par exemple la tension du jeu de barres.

Le Tableau 265 montre tous les attributs de RegulationSchedule.

Tableau 265 – Attributs de Wires::RegulationSchedule

nom	type	description
endTime	DateTime	hérité de: RegularIntervalSchedule
timeStep	Seconds	hérité de: RegularIntervalSchedule
startTime	DateTime	hérité de: BasicIntervalSchedule
value1Unit	UnitSymbol	hérité de: BasicIntervalSchedule
value1Multiplier	UnitMultiplier	hérité de: BasicIntervalSchedule
value2Unit	UnitSymbol	hérité de: BasicIntervalSchedule
value2Multiplier	UnitMultiplier	hérité de: BasicIntervalSchedule
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 266 montre toutes les extrémités d'association de RegulationSchedule avec d'autres classes.

Tableau 266 – Extrémités d'association de Wires::RegulationSchedule avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[1..1] RegulatingControl	RegulatingControl	Les commandes de régulation qui ont ce Schedule (c'est-à-dire ce programme).
[0..1]	[0..*] VoltageControlZones	VoltageControlZone	Une VoltageControlZone (zone de contrôle de la tension) peut comporter un programme de régulation de tension.
[0..*]	[0..1] DayType	DayType	hérité de: SeasonDayTypeSchedule
[0..*]	[0..1] Season	Season	hérité de: SeasonDayTypeSchedule
[1..1]	[1..*] TimePoints	RegularTimePoint	hérité de: RegularIntervalSchedule
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.55 Resistor

Résistance, typiquement utilisée dans les configurations de filtre ou comme résistance de mise à la terre pour des transformateurs. Utilisée pour le modèle électrique des réseaux de distribution.

Le Tableau 267 montre tous les attributs de Resistor.

Tableau 267 – Attributs de Wires::Resistor

nom	type	description
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 268 montre toutes les extrémités d'association de Resistor avec d'autres classes.

Tableau 268 – Extrémités d'association de Wires::Resistor avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.56 RotatingMachine

Une machine tournante qui peut être utilisée comme générateur ou comme moteur.

Le Tableau 269 montre tous les attributs de RotatingMachine.

Tableau 269 – Attributs de Wires::RotatingMachine

nom	type	description
damping	Float	Coefficient du couple d'amortissement, constante proportionnelle qui, lorsqu'elle est multipliée par la vitesse angulaire des pôles du rotor relative au champ magnétique (fréquence), se traduit par le couple d'amortissement. Cette valeur est souvent nulle quand les sources de couple d'amortissement (enroulements amortisseurs de générateur, effets amortisseurs de charge, etc.) sont modélisés en détail.
inertia	Seconds	Constante d'inertie du générateur ou du moteur et charge mécanique. Doit être supérieure à zéro. Ceci représente la spécification pour l'énergie emmagasinée dans la masse de rotation quand le fonctionnement a lieu à la vitesse assignée. Pour un générateur, sont compris à la fois le générateur et tous les autres éléments (turbine, exciteur) qui se trouvent sur le même arbre et qui a des unités de MW-s. Pour un moteur, sont compris à la fois le moteur et sa charge mécanique. Les unités conventionnelles sont entendues par unité sur la base du MVA du générateur, habituellement exprimées en MW-s/MVA ou juste en secondes. Cette valeur est utilisée dans le référentiel électrique d'accélération pour les programmes de simulateur de formation des exploitants.
ratedS	ApparentPower	Puissance apparente nominale constructeur de la tranche.
saturationFactor	Float	Facteur de saturation à la tension assignée de borne. Il est recommandé qu'il soit supérieur ou égal à zéro.
saturationFactor120	Float	Facteur de saturation à 120 % de la tension assignée de borne. Il est recommandé qu'il soit supérieur ou égal au facteur de saturation à la tension assignée de borne.
statorLeakageReactance	Reactance	Réactance de fuite du stator. Il est recommandé qu'elle soit supérieure ou égale à zéro.
statorResistance	Resistance	Résistance du stator (armature). Il est recommandé qu'il soit supérieur ou égal à zéro. Résistance équivalente quand elle est utilisée pour un modèle GenEquiv.
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 270 montre toutes les extrémités d'association de RotatingMachine avec d'autres classes.

**Tableau 270 – Extrémités d'association de Wires::RotatingMachine
avec d'autres classes**

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] RegulatingControl	RegulatingControl	hérité de: RegulatingCondEq
[0..1]	[1..*] Controls	Control	hérité de: RegulatingCondEq
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.57 Sectionaliser

Commutateur automatique capable de se verrouiller en position ouverte pour isoler une section sujette à un défaut. Il peut ou non comporter une capacité de coupure. Il a pour rôle premier de fournir un sectionnement de défaut à des endroits où le courant de défaut est soit trop élevé soit trop faible pour coordonner correctement les fusibles.

Le Tableau 271 montre tous les attributs de Sectionaliser.

Tableau 271 – Attributs de Wires::Sectionaliser

nom	type	description
normalOpen	Boolean	hérité de: Switch
ratedCurrent	CurrentFlow	hérité de: Switch
switchOnCount	Integer	hérité de: Switch
switchOnDate	DateTime	hérité de: Switch
retained	Boolean	hérité de: Switch
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 272 montre toutes les extrémités d'association de Sectionaliser avec d'autres classes.

Tableau 272 – Extrémités d'association de Wires::Sectionaliser avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[1..1]	[0..*] SwitchSchedules	SwitchSchedule	hérité de: Switch
[1..1]	[0..*] SwitchPhase	SwitchPhase	hérité de: Switch
[0..*]	[0..1] CompositeSwitch	CompositeSwitch	hérité de: Switch
[0..*]	[0..*] SwitchingOperations	SwitchingOperation	hérité de: Switch
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.58 SeriesCompensator

Un Series Compensator (compensateur série) est un condensateur ou une bobine de réactance série ou une ligne de transport en courant alternatif sans susceptance de charge. C'est un appareil à deux bornes.

Le Tableau 273 montre tous les attributs de SeriesCompensator.

Tableau 273 – Attributs de Wires::SeriesCompensator

nom	type	description
r	Resistance	Résistance directe.
r0	Resistance	Résistance homopolaire.
x	Reactance	Réactance directe.
x0	Reactance	Réactance homopolaire.
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 274 montre toutes les extrémités d'association de SeriesCompensator avec d'autres classes.

Tableau 274 – Extrémités d'association de Wires::SeriesCompensator avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.59 ShuntCompensator

Condensateur-shunt ou bobine de réactance ou batterie commutable de condensateurs-shunts ou de bobines de réactance. Une section d'un compensateur shunt est un condensateur individuel ou une bobine de réactance individuelle. Une valeur négative pour reactivePerSection indique que le compensateur est une bobine de réactance. ShuntCompensator est un dispositif à une seule borne. La masse est impliquée.

Le Tableau 275 montre tous les attributs de ShuntCompensator.

Tableau 275 – Attributs de Wires::ShuntCompensator

nom	type	description
aVRDelay	Seconds	Durée, en secondes, nécessaire au dispositif pour être connecté ou déconnecté par une régulation automatique de tension (AVR, automatic voltage regulation).
b0PerSection	Susceptance	Susceptance shunt homopolaire (chargement), par section
bPerSection	Susceptance	Susceptance directe (chargement), par section
g0PerSection	Conductance	Conductance shunt homopolaire (chargement), par section
gPerSection	Conductance	Conductance directe (chargement), par section
grounded	WindingConnection	Utilisé pour des connexions Yn et Zn. Vrai si le neutre est solidement mis à la terre.
maximumSections	Integer	Nombre maximal de sections qui peuvent être mises en service.
nomU	Voltage	La tension à laquelle peut être calculée la puissance réactive nominale. Il convient qu'elle soit normalement différente d'au plus 10% de la tension à laquelle le condensateur est connecté au réseau.
normalSections	Integer	Le nombre normal de sections mises en service.
phaseConnection	PhaseShuntConnectionKind	Le type de connexion de phase, par exemple montage en étoile ou en triangle.
switchOnCount	Integer	Compteur de mise en route depuis la dernière initialisation ou réinitialisation du compteur de condensateur.
switchOnDate	DateTime	Date et moment où la batterie de condensateurs a été activée pour la dernière fois.
voltageSensitivity	VoltagePerReactivePower	Sensibilité de tension nécessaire au dispositif pour réguler la tension du bus en tension par puissance réactive.
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 276 montre toutes les extrémités d'association de ShuntCompensator avec d'autres classes.

**Tableau 276 – Extrémités d'association de Wires::ShuntCompensator
avec d'autres classes**

[mult à partir de]	[mult vers] nom	type	description
[1..1]	[0..*] ShuntCompensatorPhase	ShuntCompensatorPhase	Les modèles individuels de phase pour le compensateur shunt.
[1..1]	[0..1] SvShuntCompensatorSections	SvShuntCompensatorSections	L'état pour le nombre des sections de compensateur shunt en service.
[0..*]	[0..1] RegulatingControl	RegulatingControl	hérité de: RegulatingCondEq
[0..1]	[1..*] Controls	Control	hérité de: RegulatingCondEq
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.60 ShuntCompensatorPhase

Phase unique d'un compensateur shunt polyphasé quand il est possible que ses attributs soient différents pour chaque phase.

Le Tableau 277 montre tous les attributs de ShuntCompensatorPhase.

Tableau 277 – Attributs de Wires::ShuntCompensatorPhase

nom	type	description
bPerSection	Susceptance	Susceptance par section de la phase si le compensateur shunt est monté en étoile. Susceptance par section de la phase si le compensateur shunt est monté en triangle.
gPerSection	Conductance	Conductance par section de cette phase si le compensateur shunt est monté en étoile. Conductance par section phase – phase si le compensateur shunt est monté en triangle.
maximumSections	Integer	Le nombre maximal de sections qui peuvent être mises en service pour cette phase.
normalSections	Integer	Pour la phase du condensateur, le nombre normal de sections mises en service.
phase	SinglePhaseKind	Phase de ce composant de compensateur shunt. Si le compensateur shunt est monté en étoile, la connexion est réalisée de la phase indiquée vers la mise centrale à la terre ou le point neutre. Si le compensateur shunt est monté en triangle, la phase indique qu'un compensateur shunt est connecté de la phase indiquée vers la phase non neutre logique qui suit.
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 278 montre toutes les extrémités d'association de ShuntCompensatorPhase avec d'autres classes.

Tableau 278 – Extrémités d'association de Wires::ShuntCompensatorPhase avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[1..1] ShuntCompensator	ShuntCompensator	Compensateur shunt de cette phase de compensateur shunt.
[0..*]	[0..1] PSRTType	PSRTType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.61 Énumération SinglePhaseKind

Énumération des identificateurs de phase particulière. Permet de désigner des phases particulières à la fois pour l'équipement, les circuits et les charges de transport et de distribution.

Le Tableau 279 montre tous les libellés de SinglePhaseKind.

Tableau 279 – Libellés de Wires::SinglePhaseKind

libellé	description
A	Phase A.
B	Phase B.
C	Phase C.
N	Neutre.
s1	Phase secondaire 1.
s2	Phase secondaire 2.

6.8.62 StaticVarCompensator

Installation proposant une puissance réactive de shunt variable et modulable. Le SVC consiste généralement en un transformateur abaisseur, un filtre, un réacteur commandé par thyristor et des bras de condensateur commutés par thyristor.

Le SVC peut fonctionner dans le mode de sortie MVar fixe ou dans le mode de commande de tension. Dans le mode de commande de tension, la sortie du SVC sera proportionnelle à l'écart de tension au niveau du bus commandé par rapport au point de consigne de la tension. La pente caractéristique du SVC définit la proportion. Si la tension au niveau du bus commandé est égale au point de consigne de la tension, la sortie MVar du SVC est nulle.

Le Tableau 280 montre tous les attributs de StaticVarCompensator.

Tableau 280 – Attributs de Wires::StaticVarCompensator

nom	type	description
capacitiveRating	Reactance	Maximum de réactance capacitive disponible.
inductiveRating	Reactance	Maximum de réactance inductive disponible.
pente	VoltagePerReactivePower	La pente de la caractéristique d'un SVC définit comment la sortie de puissance réactive change proportionnellement à la différence entre la tension régulée du bus et le point de consigne de la tension.
sVCControlMode	SVCControlMode	Mode de commande du SVC.
voltageSetPoint	Voltage	La sortie de puissance réactive du SVC est proportionnelle à la différence entre la tension au bus régulé et le point de consigne de la tension (voltage set point). Lorsque la tension du bus régulé est égale au point de consigne de la tension, la sortie de puissance réactive est nulle.
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 281 montre toutes les extrémités d'association de StaticVarCompensator avec d'autres classes.

Tableau 281 – Extrémités d'association de Wires::StaticVarCompensator avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] RegulatingControl	RegulatingControl	hérité de: RegulatingCondEq
[0..1]	[1..*] Controls	Control	hérité de: RegulatingCondEq
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.63 Enumération SVCControlMode

Mode de commande du compensateur de VAR statique.

Le Tableau 282 montre tous les libellés de SVCControlMode.

Tableau 282 – Libellés de Wires::SVCControlMode

libellé	description
reactivePower	
voltage	
off	

6.8.64 Switch

Appareil générique visant à établir ou à interrompre le courant, ou les deux, dans un ou plusieurs circuits électriques.

Le Tableau 283 montre tous les attributs de Switch.

Tableau 283 – Attributs de Wires::Switch

nom	type	description
normalOpen	Boolean	L'attribut est utilisé dans des cas où aucune Measurement (c'est-à-dire mesure) pour la valeur de statut n'est présente. Si le Switch a une mesure de statut, alors Discrete.normalValue est censé correspondre au Switch.normalOpen.
ratedCurrent	CurrentFlow	La capacité de transport maximale du courant continu en ampères d'après le matériau et la construction de l'appareil.

nom	type	description
switchOnCount	Integer	Compteur de mise en route depuis la dernière initialisation ou réinitialisation du Switch.
switchOnDate	DateTime	Date et heure de la dernière mise en route du Switch.
retained	Boolean	La branche est retenue dans un modèle en topologie nodale.
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 284 montre toutes les extrémités d'association de Switch avec d'autres classes.

Tableau 284 – Extrémités d'association de Wires::Switch avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[1..1]	[0..*] SwitchSchedules	SwitchSchedule	Un Switch peut être associé avec SwitchSchedules.
[1..1]	[0..*] SwitchPhase	SwitchPhase	Les phases commutateur individuelles pour le commutateur.
[0..*]	[0..1] CompositeSwitch	CompositeSwitch	Commutateur composite auquel appartient ce Switch.
[0..*]	[0..*] SwitchingOperations	SwitchingOperation	Un organe de coupure peut être actionné par plusieurs programmes.
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.65 SwitchPhase

Phase particulière d'un commutateur polyphase quand il est possible que ses attributs soient différents pour chaque phase.

Le Tableau 285 montre tous les attributs de SwitchPhase.

Tableau 285 – Attributs de Wires::SwitchPhase

nom	type	description
normalOpen	Boolean	Utilisé dans des cas où aucune Measurement (c'est-à-dire mesure) pour la valeur de statut n'est présente. Si la SwitchPhase a une mesure de statut, alors Discrete.normalValue est censé correspondre à cette valeur.
phaseSide1	SinglePhaseKind	Phase de cette SwitchPhase sur le côté dont le numéro d'ordre de borne est égal à 1. Il est recommandé que ce soit une phase contenue dans cet attribut de phase de borne.
phaseSide2	SinglePhaseKind	Phase de cette SwitchPhase sur le côté dont le numéro d'ordre de borne est égal à 2. Il est recommandé que ce soit une phase contenue dans cet attribut de phase Terminal.phases.
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 286 montre toutes les extrémités d'association de SwitchPhase avec d'autres classes.

Tableau 286 – Extrémités d'association de Wires::SwitchPhase avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[1..1] Switch	Switch	Le commutateur de la phase de commutateur.
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.66 SwitchSchedule

Un programme des positions de commutateur. Si RegularTimePoint.value1 est 0, le commutateur est ouvert. Si 1, le commutateur est fermé.

Le Tableau 287 montre tous les attributs de SwitchSchedule.

Tableau 287 – Attributs de Wires::SwitchSchedule

nom	type	description
endTime	DateTime	hérité de: RegularIntervalSchedule
timeStep	Seconds	hérité de: RegularIntervalSchedule
startTime	DateTime	hérité de: BasicIntervalSchedule
value1Unit	UnitSymbol	hérité de: BasicIntervalSchedule
value1Multiplier	UnitMultiplier	hérité de: BasicIntervalSchedule
value2Unit	UnitSymbol	hérité de: BasicIntervalSchedule
value2Multiplier	UnitMultiplier	hérité de: BasicIntervalSchedule
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 288 montre toutes les extrémités d'association de SwitchSchedule avec d'autres classes.

Tableau 288 – Extrémités d'association de Wires::SwitchSchedule avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[1..1] Switch	Switch	Un SwitchSchedule est associé à un Switch.
[0..*]	[0..1] DayType	DayType	hérité de: SeasonDayTypeSchedule
[0..*]	[0..1] Season	Season	hérité de: SeasonDayTypeSchedule
[1..1]	[1..*] TimePoints	RegularTimePoint	hérité de: RegularIntervalSchedule
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.67 Enumération SynchronousGeneratorType

Type de générateur synchrone utilisé dans des applications de simulation dynamique.

Le Tableau 289 montre tous les libellés de SynchronousGeneratorType.

Tableau 289 – Libellés de Wires::SynchronousGeneratorType

libellé	description
roundRotor	Aussi appelé genrou.
salientPole	Aussi appelé gensal.
transient	Aussi appelé gentra.
typeF	Aussi appelé typeF.
typeJ	Aussi appelé typeJ.

6.8.68 SynchronousMachine

Appareil électromécanique fonctionnant par rotation synchrone d'un arbre avec le réseau. Il s'agit d'une machine unique opérant comme une génératrice, comme un condensateur synchrone ou comme une pompe.

Le Tableau 290 montre tous les attributs de SynchronousMachine.

Tableau 290 – Attributs de Wires::SynchronousMachine

nom	type	description
aVRToManualLag	Seconds	Retard requis pour le passage de la régulation de tension automatique (Automatic Voltage Regulation (AVR)) à Manuel pour une infraction MVAR de retard de phase.
aVRToManualLead	Seconds	Retard requis pour le passage de la régulation de tension automatique (AVR) à Manuel pour une infraction MVAR d'avance de phase.
baseQ	ReactivePower	Valeur par défaut de la puissance réactive de base. Cette valeur représente la puissance réactive initiale pouvant être utilisée par toute fonction d'application.
condenserP	ActivePower	Puissance active consommée lors du fonctionnement en mode condensateur.
coolantCondition	Float	Température ou pression du milieu du fluide de refroidissement
coolantType	CoolantType	Méthode de refroidissement de la machine.
manualToAVR	Seconds	Retard requis pour le passage de Manuel à régulation automatique de tension. Cette valeur est utilisée dans le référentiel électrique d'accélération pour les programmes de calcul de répartition
maxQ	ReactivePower	Limite maximale de la puissance réactive. Il s'agit de la limite maximale (constructeur) de la tranche.
maxU	Voltage	Limite de tension maximale de la tranche.
minQ	ReactivePower	Limite de puissance réactive minimale de la tranche.
minU	Voltage	Limite de tension minimale de la tranche.
operatingMode	SynchronousMachineOperatingMode	Mode de fonctionnement courant.
qPercent	PerCent	Pourcentage de la commande réactive coordonnée qui provient de cette machine.
r	Resistance	Résistance directe de la machine synchrone.
r0	Resistance	Résistance homopolaire de la machine synchrone.
r2	Resistance	Résistance inverse.
referencePriority	Integer	Priorité de l'unité pour une utilisation comme sélection de bus de référence pour l'angle de phase de la tension du flux d'énergie. 0 = don t care c'est-à-dire peu importe (par défaut) 1 = plus haute priorité. 2 est moins que 1 et ainsi de suite.
synchronousGeneratorType	SynchronousGeneratorType	Type de Synchronous Generator utilisé dans des applications de simulation Dynamic.
tpdo	Seconds	Constante de temps longitudinale transitoire du rotor, aussi appelé T'do
tppdo	Seconds	Constante de temps longitudinale sous-transitoire du rotor, aussi appelé T"do

nom	type	description
tppqo	Seconds	Constante de temps sous-transitoire transversale du rotor, aussi appelé $T''q_o$
tpqo	Seconds	Constante de temps transitoire transversale du rotor, aussi appelé $T'q_o$
type	SynchronousMachineType	Modes dans lesquels cette machine synchrone peut fonctionner.
x	Reactance	Réactance directe de la machine synchrone.
x0	Reactance	Réactance homopolaire de la machine synchrone.
x2	Reactance	Réactance inverse.
xDirectSubtrans	Reactance	Réactance subtransitoire longitudinale, également appelée $X''d$.
xDirectSync	Reactance	Réactance synchrone longitudinale. Le quotient, pour une machine fonctionnant à une vitesse nominale, de la valeur permanente du composant alternatif de la tension d'induit, produit par le flux longitudinal dû au courant d'induit longitudinal et de la valeur du composant alternatif de ce courant. (X_d)
xDirectTrans	Reactance	Réactance transitoire longitudinale, également appelée $X'd$.
xQuadSubtrans	Reactance	Réactance subtransitoire transversale, également appelée $X''q$.
xQuadSync	Reactance	Réactance synchrone transversale (X_q), rapport entre la composante de la tension réactive d'induit, due à la composante transversale du courant d'induit, et la composante du courant, selon des répartitions de modes à l'équilibre et à une fréquence assignée.
xQuadTrans	Reactance	Réactance transitoire transversale, également appelée $X'q$.
damping	Float	hérité de: RotatingMachine
inertia	Seconds	hérité de: RotatingMachine
ratedS	ApparentPower	hérité de: RotatingMachine
saturationFactor	Float	hérité de: RotatingMachine
saturationFactor120	Float	hérité de: RotatingMachine
statorLeakageReactance	Reactance	hérité de: RotatingMachine
statorResistance	Resistance	hérité de: RotatingMachine
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 291 montre toutes les extrémités d'association de SynchronousMachine avec d'autres classes.

Tableau 291 – Extrémités d'association de Wires::SynchronousMachine avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[1..1]	[0..1] HydroPump	HydroPump	La machine synchrone entraîne la turbine qui déplace l'eau de bas en haut. Le sens de rotation de la machine pour le pompage peut être identique ou non à celui utilisé pour la production.
[1..*]	[0..1] InitialReactiveCapabilityCurve	ReactiveCapabilityCurve	La courbe de capacité réactive par défaut à utiliser pour une machine synchrone.
[1..*]	[0..*] ReactiveCapabilityCurves	ReactiveCapabilityCurve	Toutes les courbes disponibles de capacité réactive pour cette machine synchrone.
[1..*]	[0..1] GeneratingUnit	GeneratingUnit	Une machine synchrone peut fonctionner comme génératrice et devenir par conséquent un élément de l'unité de production.
[0..*]	[0..*] PrimeMovers	PrimeMover	Machines motrices qui entraînent ce SynchronousMachine (c'est-à-dire cette machine synchrone).
[0..*]	[0..1] RegulatingControl	RegulatingControl	hérité de: RegulatingCondEq
[0..1]	[1..*] Controls	Control	hérité de: RegulatingCondEq
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	hérité de: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	hérité de: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	hérité de: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	hérité de: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.69 Enumération SynchronousMachineOperatingMode

Mode de fonctionnement de la machine synchrone.

Le Tableau 292 montre tous les libellés de SynchronousMachineOperatingMode.

Tableau 292 – Libellés de Wires::SynchronousMachineOperatingMode

libellé	description
generator	
condenser	

6.8.70 Enumération SynchronousMachineType

Type de la machine synchrone.

Le Tableau 293 montre tous les libellés de SynchronousMachineType.

Tableau 293 – Libellés de Wires::SynchronousMachineType

libellé	description
generator	
condenser	
generator_or_condenser	

6.8.71 TapChanger

Mécanisme de modification des positions des prises de réglage.

Le Tableau 294 montre tous les attributs de TapChanger.

Tableau 294 – Attributs de Wires::TapChanger

nom	type	description
highStep	Integer	Position de la prise la plus élevée possible en partant de neutre
initialDelay	Seconds	Pour un changeur de prise en charge, retard de fonctionnement initial du régleur (première modification)
lowStep	Integer	Position de prise la plus basse possible (retardée à neutre)
ItcFlag	Boolean	Spécifie si, oui ou non, un TapChanger a des capacités de changeur de prise en charge.
neutralStep	Integer	Position neutre de prise pour cet enroulement.
neutralU	Voltage	Tension à laquelle l'enroulement fonctionne au réglage de la prise neutre.
normalStep	Integer	Position de prise utilisée dans le fonctionnement "normal" du réseau pour cet enroulement. Régleur "Fixed" (fixe) indique le réglage matériel actuel.
regulationStatus	Boolean	Spécifie le statut de régulation par défaut du TapChanger. True, il régule. False, il ne régule pas.
subsequentDelay	Seconds	Pour un changeur de prise en charge, délai de fonctionnement supplémentaire du changeur de prise (deuxième modification et modifications suivantes de l'échelon)
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 295 montre toutes les extrémités d'association de TapChanger avec d'autres classes.

Tableau 295 – Extrémités d'association de Wires::TapChanger avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] TapChangerControl	TapChangerControl	Le plan de commande de régulation auquel ce changeur de prise participe.
[1..1]	[0..1] SvTapStep	SvTapStep	L'état d'échelon de prise associé au changeur de prise.
[1..1]	[0..*] TapSchedules	TapSchedule	Un TapChanger peut avoir des TapSchedules.
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.72 TapChangerControl

Décrit le comportement spécifique aux changeurs de prise, par exemple la manière dont la tension à l'extrémité d'une ligne varie avec le niveau de charge et la compensation de la chute de tension lors de l'ajustement de la prise.

Le Tableau 296 montre tous les attributs de TapChangerControl.

Tableau 296 – Attributs de Wires::TapChangerControl

nom	type	description
limitVoltage	Voltage	Tension régulée maximale autorisée sur le secondaire du PT, quelle que soit la compensation de perte de ligne. Parfois appelée «first-house protection» (protection de premier stade).
lineDropCompensation	Boolean	Si true (c'est-à-dire vrai), la compensation de perte de ligne doit être appliquée.
lineDropR	Resistance	Réglage de la résistance de compensation de perte de ligne pour le flux de puissance normal (direct).
lineDropX	Reactance	Réglage de la réactance de compensation de perte de ligne pour le flux de puissance normal (direct).
reverseLineDropR	Resistance	Réglage de la résistance de compensation de perte de ligne pour le flux de puissance inverse.
reverseLineDropX	Reactance	Réglage de la réactance de compensation de perte de ligne pour le flux de puissance inverse.
mode	RegulatingControlModeKind	hérité de: RegulatingControl
targetRange	Float	hérité de: RegulatingControl
targetValue	Float	hérité de: RegulatingControl
discrete	Boolean	hérité de: RegulatingControl
monitoredPhase	PhaseCode	hérité de: RegulatingControl
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 297 montre toutes les extrémités d'association de TapChangerControl avec d'autres classes.

Tableau 297 – Extrémités d'association de Wires::TapChangerControl avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[0..*] TapChanger	TapChanger	Changeur de prise qui participe à ce plan de commande de régulation.
[0..*]	[0..1] Terminal	Terminal	hérité de: RegulatingControl
[0..1]	[0..*] RegulatingCondEq	RegulatingCondEq	hérité de: RegulatingControl
[1..1]	[0..*] RegulationSchedule	RegulationSchedule	hérité de: RegulatingControl
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.73 Enumération TapChangerKind

Type du changeur de prise du transformateur. Indique les capacités du changeur de prise indépendamment du mode de fonctionnement.

Le Tableau 298 montre tous les libellés de TapChangerKind.

Tableau 298 – Libellés de Wires::TapChangerKind

libellé	description
balise	Non capable de commande. Cela est aussi indiqué par l'absence d'association de TapChanger à un RegulatingControl.
voltageControl	Capable de commande de la tension.
phaseControl	Capable de commande de la phase.
voltageAndPhaseControl	Capable de commande de la tension et de la phase.

6.8.74 TapSchedule

Modèle prédéfini sur une durée pour un échelon de prise.

Le Tableau 299 montre tous les attributs de TapSchedule.

Tableau 299 – Attributs de Wires::TapSchedule

nom	type	description
endTime	DateTime	hérité de: RegularIntervalSchedule
timeStep	Seconds	hérité de: RegularIntervalSchedule
startTime	DateTime	hérité de: BasicIntervalSchedule
value1Unit	UnitSymbol	hérité de: BasicIntervalSchedule
value1Multiplier	UnitMultiplier	hérité de: BasicIntervalSchedule
value2Unit	UnitSymbol	hérité de: BasicIntervalSchedule
value2Multiplier	UnitMultiplier	hérité de: BasicIntervalSchedule
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 300 montre toutes les extrémités d'association de TapSchedule avec d'autres classes.

Tableau 300 – Extrémités d'association de Wires::TapSchedule avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[1..1] TapChanger	TapChanger	Un TapSchedule est associé à un TapChanger.
[0..*]	[0..1] DayType	DayType	hérité de: SeasonDayTypeSchedule
[0..*]	[0..1] Season	Season	hérité de: SeasonDayTypeSchedule
[1..1]	[1..*] TimePoints	RegularTimePoint	hérité de: RegularIntervalSchedule
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.75 Enumération TransformerControlMode

Modes de commande pour un transformateur.

Le Tableau 301 montre tous les libellés de TransformerControlMode.

Tableau 301 – Libellés de Wires::TransformerControlMode

libellé	description
volt	Commande de tension
reactive	Commande du flux de puissance réactive

6.8.76 TransformerCoreAdmittance

L'admittance du noyau du transformateur. Utilisée pour spécifier l'admittance du noyau d'un transformateur de telle manière qu'elle puisse être répartie entre transformateurs de puissance.

Le Tableau 302 montre tous les attributs de TransformerCoreAdmittance.

Tableau 302 – Attributs de Wires::TransformerCoreAdmittance

nom	type	description
b	Susceptance	Susceptance de branche magnétique (B mag). La valeur peut être positive ou négative.
b0	Susceptance	Susceptance de branche magnétique homopolaire.
g	Conductance	Conductance de branche magnétique (G mag).
g0	Conductance	Conductance de branche magnétique homopolaire.
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 303 montre toutes les extrémités d'association de TransformerCoreAdmittance avec d'autres classes.

Tableau 303 – Extrémités d'association de Wires::TransformerCoreAdmittance avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[0..*] TransformerEnd	TransformerEnd	Toutes les extrémités de transformateur qui ont cette admittance de noyau.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.77 TransformerEnd

Un point de connexion conductrice d'un transformateur de puissance. Il correspond à une borne d'enroulement physique du transformateur. Dans les versions antérieures du CIM, la classe TransformerWinding avait un but comparable mais cette classe est plus souple parce qu'elle s'associe à une borne mais qu'elle n'est pas une spécialisation de ConductingEquipment.

Le Tableau 304 montre tous les attributs de TransformerEnd.

Tableau 304 – Attributs de Wires::TransformerEnd

nom	type	description
bmagSat	PerCent	Susceptance d'aimantation du shunt central dans la zone de saturation.
endNumber	Integer	Numéro pour cette extrémité de transformateur, qui correspond à l'ordre des extrémités dans le groupe vectoriel du transformateur de puissance ou au numéro de position de l'angle de phase. Il est recommandé que l'enroulement de tension maximum soit 1. Il est recommandé que chaque extrémité dans un transformateur de puissance ait un nombre unique d'extrémité ultérieure. Noter que le numéro d'extrémité du transformateur ne correspond pas nécessairement au numéro d'ordre de la borne.
grounded	Boolean	(pour les connexions Yn et Zn) True (c'est-à-dire vrai) si le neutre est mis à la terre solidement.
magBaseU	Voltage	Tension de référence à laquelle les mesures de saturation d'aimantation ont été effectuées
magSatFlux	PerCent	Niveau de flux au coude de la courbe de saturation et d'aimantation du cœur.

nom	type	description
rground	Resistance	(pour les connexions Yn et Zn) Partie résistive de l'impédance neutre lorsque 'grounded' (c'est-à-dire mis à la terre) est true (c'est-à-dire vrai).
xground	Reactance	(pour les connexions Yn et Zn) Partie réactive de l'impédance neutre lorsque 'grounded' (c'est-à-dire mis à la terre) est true (c'est-à-dire vrai).
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 305 montre toutes les extrémités d'association de TransformerEnd avec d'autres classes.

Tableau 305 – Extrémités d'association de Wires::TransformerEnd avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] BaseVoltage	BaseVoltage	Tension de base de l'extrémité du transformateur. Elle est primordiale pour calculer le PU.
[0..*]	[0..1] Terminal	Terminal	Borne du transformateur de puissance à laquelle appartient cette extrémité de transformateur.
[1..1]	[0..1] PhaseTapChanger	PhaseTapChanger	Changeur de prise pour phase associé à cette extrémité de transformateur.
[1..1]	[0..1] RatioTapChanger	RatioTapChanger	Changeur de prise pour rapport associé à cette extrémité de transformateur.
[0..*]	[0..1] StarImpedance	TransformerStarImpedance	(précis pour les transformateurs à deux ou trois enroulements seulement) Impédances de modèle Pi de cette extrémité de transformateur. Par convention, il est recommandé que pour un transformateur à deux enroulements, les valeurs totales du transformateur soient entrées à l'extrémité haute tension (endNumber=1).
[0..*]	[0..1] CoreAdmittance	TransformerCoreAdmittance	Admittance de noyau de cette extrémité de transformateur, représentant le courant magnétisant et les pertes dans le fer. Il convient de fournir toutes les valeurs du transformateur pour une extrémité de transformateur uniquement.
[1..1]	[0..*] FromMeshImpedance	TransformerMeshImpedance	Toutes les impédances de maille entre cette extrémité de transformateur (vers) et les autres extrémités de transformateur (de).
[1..*]	[0..*] ToMeshImpedance	TransformerMeshImpedance	Toutes les impédances de maille entre cette extrémité de transformateur (de) et les autres extrémités de transformateur (vers).
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.78 TransformerMeshImpedance

Impédance en maille du transformateur modèle en triangle (Delta-model) entre les extrémités du transformateur.

Le cas typique est que cette classe décrit par paires l'impédance entre deux extrémités de transformateur, c'est à dire que les cardinalités sont 1 aux deux associations d'extrémité de transformateur. Mais les cardinalités sont supérieures à 1 quand deux extrémités de transformateur ou davantage sont modélisées.

Le Tableau 306 montre tous les attributs de TransformerMeshImpedance.

Tableau 306 – Attributs de Wires::TransformerMeshImpedance

nom	type	description
r	Resistance	Résistance entre l'extrémité d'origine (de) et de destination (vers), du point de vue de l'extrémité d'origine (de).
r0	Resistance	Résistance homopolaire entre l'extrémité d'origine (de) et de destination (vers), du point de vue de l'extrémité d'origine (de).
x	Reactance	Réactance entre l'extrémité d'origine (de) et de destination (vers), du point de vue de l'extrémité d'origine (de).
x0	Reactance	Réactance homopolaire entre l'extrémité d'origine (de) et de destination (vers), du point de vue de l'extrémité de l'extrémité d'origine (de).
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
nom	String	hérité de: IdentifiedObject

Le Tableau 307 montre toutes les extrémités d'association de TransformerMeshImpedance avec d'autres classes.

Tableau 307 – Extrémités d'association de Wires::TransformerMeshImpedance avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[1..1] FromTransformerEnd	TransformerEnd	Extrémité d'origine à laquelle est connectée cette impédance en maille. Elle détermine la référence de tension.
[0..*]	[1..*] ToTransformerEnd	TransformerEnd	Toutes les extrémités de transformateur auxquelles est connectée cette impédance en maille.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.79 TransformerStarImpedance

Impédance en étoile du transformateur (modèle Pi) qui reflète précisément l'impédance pour les transformateurs à deux ou trois enroulements. Pour les transformateurs à quatre enroulements ou plus, il faut utiliser TransformerMeshImpedance.

Pour des réseaux de transport, utiliser les impédances PowerTransformerEnd (r, r0, x, x0, b, b0, g et g0).

Le Tableau 308 montre tous les attributs de TransformerStarImpedance.

Tableau 308 – Attributs de Wires::TransformerStarImpedance

nom	type	description
r	Resistance	Résistance de l'extrémité du transformateur.
r0	Resistance	Résistance série homopolaire de l'extrémité du transformateur..
x	Reactance	Réactance série directe de l'extrémité du transformateur.
x0	Reactance	Réactance série homopolaire de l'extrémité du transformateur.
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 309 montre toutes les extrémités d'association de TransformerStarImpedance avec d'autres classes.

Tableau 309 – Extrémités d'association de Wires::TransformerStarImpedance avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[0..*] TransformerEnd	TransformerEnd	Toutes les extrémités de transformateur qui ont cette impédance en étoile.
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.80 TransformerTank

Ensemble de deux ou plusieurs enroulements couplés qui transforme la puissance électrique entre des niveaux de tension. Ces enroulements sont liés sur un noyau commun et se trouvent dans la même cuve. La cuve de transformateur peut être utilisée pour modéliser à la fois des transformateurs monophasés et triphasés.

Le Tableau 310 montre tous les attributs de TransformerTank.

Tableau 310 – Attributs de Wires::TransformerTank

nom	type	description
aggregate	Boolean	hérité de: Equipement
normallyInService	Boolean	hérité de: Equipement
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 311 montre toutes les extrémités d'association de TransformerTank avec d'autres classes.

Tableau 311 – Extrémités d'association de Wires::TransformerTank avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..1] PowerTransformer	PowerTransformer	Batterie à laquelle ce transformateur appartient.
[0..1]	[1..*] TransformerTankEnds	TransformerTankEnd	Tous les enroulements de ce transformateur.
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	hérité de: Equipement
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	hérité de: Equipement
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	hérité de: Equipement
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.81 TransformerTankEnd

Une extrémité de cuve de transformateur représente un enroulement individuel pour des modèles non équilibrés ou pour des cuves de transformateur connectées dans une batterie (et une batterie est modélisée avec le PowerTransformer).

Le Tableau 312 montre tous les attributs de TransformerTankEnd.

Tableau 312 – Attributs de Wires::TransformerTankEnd

nom	type	description
phases	PhaseCode	Décrit les phases supportées par un équipement conducteur.
bmagSat	PerCent	hérité de: TransformerEnd
endNumber	Integer	hérité de: TransformerEnd
grounded	Boolean	hérité de: TransformerEnd
magBaseU	Voltage	hérité de: TransformerEnd
magSatFlux	PerCent	hérité de: TransformerEnd
rground	Resistance	hérité de: TransformerEnd
xground	Reactance	hérité de: TransformerEnd
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 313 montre toutes les extrémités d'association de TransformerTankEnd avec d'autres classes.

Tableau 313 – Extrémités d'association de Wires::TransformerTankEnd avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[1..*]	[0..1] TransformerTank	TransformerTank	Transformateur auquel cet enroulement appartient.
[0..*]	[0..1] BaseVoltage	BaseVoltage	hérité de: TransformerEnd
[0..*]	[0..1] Terminal	Terminal	hérité de: TransformerEnd
[1..1]	[0..1] PhaseTapChanger	PhaseTapChanger	hérité de: TransformerEnd
[1..1]	[0..1] RatioTapChanger	RatioTapChanger	hérité de: TransformerEnd
[0..*]	[0..1] StarImpedance	TransformerStarImpedance	hérité de: TransformerEnd
[0..*]	[0..1] CoreAdmittance	TransformerCoreAdmittance	hérité de: TransformerEnd
[1..1]	[0..*] FromMeshImpedance	TransformerMeshImpedance	hérité de: TransformerEnd
[1..*]	[0..*] ToMeshImpedance	TransformerMeshImpedance	hérité de: TransformerEnd
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.82 VoltageControlZone

Section du réseau électrique déterminée pour réguler la tension secondaire. Une zone de réglage de la tension consiste en une collection de postes avec un tronçon de jeu de barres déterminée dont la tension sera contrôlée.

Le Tableau 314 montre tous les attributs de VoltageControlZone.

Tableau 314 – Attributs de Wires::VoltageControlZone

nom	type	description
aliasName	String	hérité de: IdentifiedObject
mRID	String	hérité de: IdentifiedObject
name	String	hérité de: IdentifiedObject

Le Tableau 315 montre toutes les extrémités d'association de VoltageControlZone avec d'autres classes.

Tableau 315 – Extrémités d'association de Wires::VoltageControlZone avec d'autres classes

[mult à partir de]	[mult vers] nom	type	description
[0..1]	[1..1] BusbarSection	BusbarSection	Une VoltageControlZone (zone de réglage de la tension) est contrôlée par une BusbarSection (section de jeu de barres) déterminée.
[0..*]	[0..1] RegulationSchedule	RegulationSchedule	Une VoltageControlZone (zone de réglage de la tension) peut comporter un programme de régulation de tension.
[0..*]	[0..1] PSRType	PSRType	hérité de: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	hérité de: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	hérité de: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	hérité de: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	hérité de: PowerSystemResource

[mult à partir de]	[mult vers] nom	type	description
[0..*]	[0..*] ReportingGroup	ReportingGroup	hérité de: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	hérité de: IdentifiedObject
[1..1]	[0..*] Names	Name	hérité de: IdentifiedObject

6.8.83 Enumération WindingConnection

Type de connexion d'enroulement.

Le Tableau 316 montre tous les libellés de WindingConnection.

Tableau 316 – Libellés de Wires::WindingConnection

libellé	description
D	Delta (en triangle)
Y	Wye (en étoile)
Z	ZigZag
Yn	Wye (c'est-à-dire I grec, montage étoile), avec neutre sorti pour la mise à la terre.
Zn	ZigZag, avec neutre sorti pour la mise à la terre.
A	Enroulement commun d'autotransformateur
I	Enroulement indépendant, pour connexions monophasées

6.9 Paquetage Generation

6.9.1 Généralités

Ce paquetage contient des paquetages qui portent de l'information pour les applications Gestion de la production ("Unit Commitment"), Dispatching économiques des unités de production thermiques et hydrauliques ("Economic Dispatch of Hydro and Thermal Generating Units"), Prévion de la charge ("Load Forecasting"), Conduite automatique de production ("Automatic Generation Control"), et Modélisation d'unité pour les simulateurs d'entraînement ("Unit Modeling for Dynamic Training Simulator").

La Figure 60 montre le diagramme de classe Main.

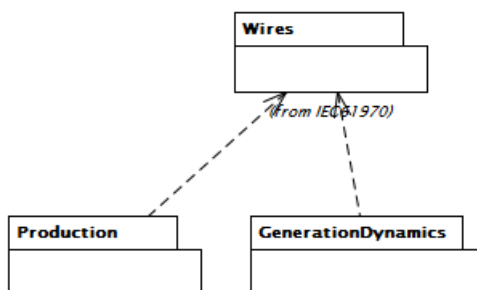


Figure 60 – Diagramme de classe Generation::Main

Ce diagramme montre les paquetages de production principale et leurs relations de dépendance par rapport à d'autres paquetages.