

DIN EN 61970-301

ICS 33.040.20; 35.240.99

Ersatz für
DIN EN 61970-301:2004-11
Siehe Anwendungsbeginn

**Schnittstelle für Anwendungsprogramme für Netzführungssysteme
(EMS-API) –****Teil 301: Allgemeines Informationsmodell (CIM), Basismodell
(IEC 61970-301:2011);****Englische Fassung EN 61970-301:2011**

Energy management system application program interface (EMS-API) –
Part 301: Common information model (CIM) base
(IEC 61970-301:2011);
English version EN 61970-301:2011

Interface de programmation d'application pour système de gestion d'énergie (EMS-API) –
Partie 301: Base de modèle d'information commun (CIM)
(CEI 61970-301:2011);
Version anglaise EN 61970-301:2011

Gesamtumfang 257 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE

Anwendungsbeginn

Anwendungsbeginn für die von CENELEC am 2011-09-30 angenommene Europäische Norm als DIN-Norm ist 2012-03-01.

Für DIN EN 61970-301:2004-11 gilt eine Übergangsfrist bis zum 2014-09-30.

Nationales Vorwort

Vorausgegangener Norm-Entwurf: E DIN EN 61970-301:2008-02.

Für diese Norm ist das nationale Arbeitsgremium K 952 „Netzleittechnik“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE (www.dke.de) zuständig.

Die enthaltene IEC-Publikation wurde vom TC 57 „Power systems management and associated information exchange“ erarbeitet.

Das IEC-Komitee hat entschieden, dass der Inhalt dieser Publikation bis zu dem Datum (stability date) unverändert bleiben soll, das auf der IEC-Website unter „<http://webstore.iec.ch>“ zu dieser Publikation angegeben ist. Zu diesem Zeitpunkt wird entsprechend der Entscheidung des Komitees die Publikation

- bestätigt,
- zurückgezogen,
- durch eine Folgeausgabe ersetzt oder
- geändert.

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf eine Norm ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils neueste gültige Ausgabe der in Bezug genommenen Norm.

Für den Fall einer datierten Verweisung im normativen Text bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe der Norm.

Der Zusammenhang der zitierten Normen mit den entsprechenden Deutschen Normen ergibt sich, soweit ein Zusammenhang besteht, grundsätzlich über die Nummer der entsprechenden IEC-Publikation. Beispiel: IEC 60068 ist als EN 60068 als Europäische Norm durch CENELEC übernommen und als DIN EN 60068 ins Deutsche Normenwerk aufgenommen.

Das Präsidium des DIN hat mit Präsidialbeschluss 1/2004 festgelegt, dass DIN-Normen, deren Inhalt sich auf internationale Arbeitsergebnisse der Informationsverarbeitung gründet, unter bestimmten Bedingungen allein in englischer Sprache veröffentlicht werden dürfen. Diese Bedingungen sind für die vorliegende Norm erfüllt.

Da sich die Nutzer der vorliegenden Norm der englischen Sprache als Fachsprache bedienen, wird die englische Fassung der IEC 61970-301 veröffentlicht. Zu deren Abschnitt 3, der die Begriffe festlegt, wurde eine Übersetzung angefertigt und als Nationaler Anhang NA der vorliegenden Norm hinzugefügt. Diese Norm steht nicht in unmittelbarem Zusammenhang mit Rechtsvorschriften und ist nicht als Sicherheitsnorm anzusehen.

Änderungen

Gegenüber DIN EN 61970-301:2004-11 wurden folgende Änderungen vorgenommen:

- a) partielle Eigentumsbeschreibungen hinzugefügt;
- b) Nicht-SI-Einheiten wurden in SI-Einheiten geändert;
- c) verschiedene editorielle Änderungen zur Korrektur des UML-Modells vorgenommen;
- d) neue Äquivalent-Einheit zum Modell des Äquivalentnetzwerkes hinzugefügt;
- e) neuer Kontrollbereich mit Lastprognose hinzugefügt;
- f) Erweiterung zur Unterstützung des Energieflusses „case input“ hinzugefügt;
- g) neue Kontingenz-Einheit zum Behandeln der Kontingenz hinzugefügt.

Frühere Ausgaben

DIN EN 61970-301: 2004-11

Nationaler Anhang NA (informativ)

NA.3 Begriffe

Für die Anwendung dieser Norm gelten die Begriffe nach IEC 61970-2 sowie die folgenden.

ANMERKUNG Für allgemeine Glossardefinitionen siehe das Internationale Elektrotechnische Wörterbuch IEC 60050.

3.1

energy management system (Energiemanagementsystem) EMS

Computersystem, das eine Softwareplattform für die grundlegenden Supportdienstleistungen und eine Reihe von Anwendungen zur Sicherstellung der Funktionalität der elektrischen Energieerzeugung und der Übertragungseinrichtungen beinhaltet, um eine angemessene Sicherheit der Energieversorgung bei minimalen Kosten zu sichern

3.2

application program interface (Anwendungsprogramm-Schnittstelle) API

eine Menge von öffentlichen Funktionen, die von einer ausführbaren Komponente zum Aufruf durch andere Anwendungskomponenten zur Verfügung gestellt wird

**Energy management system application program interface (EMS-API) -
Part 301: Common information model (CIM) base
(IEC 61970-301:2011)**

Interface de programmation d'application
pour système de gestion d'énergie
(EMS-API) -
Part 301: Base de modèle d'information
commun (CIM)
(CEI 61970-301:2011)

Schnittstelle für Anwendungsprogramme
für Netzführungssysteme (EMS-API) -
Teil 301: Allgemeines Informationsmodell
(CIM), Basismodell
(IEC 61970-301:2011)

This European Standard was approved by CENELEC on 2011-09-30. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and the United Kingdom.

CENELEC

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

Management Centre: Avenue Marnix 17, B - 1000 Brussels

Foreword

The text of document 57/1136/FDIS, future edition 3 of IEC 61970-301, prepared by IEC/TC 57 "Power systems management and associated information exchange" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN 61970-301:2011.

The following dates are fixed:

- latest date by which the document has to be implemented at national level by publication of an identical national standard or by endorsement (dop) 2012-06-30
- latest date by which the national standards conflicting with the document have to be withdrawn (dow) 2014-09-30

This document supersedes EN 61970-301:2004.

EN 61970-301:2011 includes the following significant technical changes with respect to EN 61970-301:2004:

- regulation control models were added with a new `RegulatingControl` class which provides the capability to model multiple equipments participating in a regulation scheme;
- new "OperationalLimits" package for equipment ratings;
- partial ownership specification added;
- `LoadResponseCharacteristic` class was enhanced to better model the characteristic response of the load demand due to changes in system conditions such as voltage and frequency;
- new `ControlArea` package added with load forecast and area interchange specifications;
- EPRI CIM for Planning project proposals to support the exchange of planning models were incorporated;
- branch group "interface" monitoring specification was added;
- composite switch model was changed to better model distribution use cases;
- extensions were added to support powerflow "case input" so that profiles are not required to use time based schedules and detailed connectivity (via new bus-branch model);
- new `Equivalents` package added to model equivalent networks;
- new `Contingency` package added to handle contingencies;
- various editorial changes to cleanup UML model;
- non-SI units have been changed to SI units.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CENELEC [and/or CEN] shall not be held responsible for identifying any or all such patent rights.

Endorsement notice

The text of the International Standard IEC 61970-301:2011 was approved by CENELEC as a European Standard without any modification.

In the official version, for Bibliography, the following notes have to be added for the standards indicated:

IEC 61850-7-3	NOTE Harmonized as EN 61850-7-3.
IEC 61968-11:2010	NOTE Harmonized as EN 61968-11:2010 (not modified).
IEC 61970-501	NOTE Harmonized as EN 61970-501.

CONTENTS

INTRODUCTION	14
1 Scope	16
2 Normative references	16
3 Terms and definitions	17
4 CIM specification	17
4.1 CIM modeling notation	17
4.2 CIM packages	17
4.2.1 CIM packages overview	17
4.2.2 Domain	19
4.2.3 Core	20
4.2.4 OperationalLimits	20
4.2.5 Topology	20
4.2.6 Wires	20
4.2.7 Generation	20
4.2.8 LoadModel	20
4.2.9 Outage	20
4.2.10 Protection	20
4.2.11 Equivalents	21
4.2.12 Meas	21
4.2.13 SCADA	21
4.2.14 ControlArea	21
4.2.15 Contingency	21
4.3 CIM classes and relationships	21
4.3.1 Classes	21
4.3.2 Generalization	22
4.3.3 Simple association	22
4.3.4 Aggregation	23
4.4 CIM model concepts and examples	23
4.4.1 Concepts	23
4.4.2 Containment, equipment hierarchies and naming	23
4.4.3 Connectivity model	25
4.4.4 Inheritance hierarchy	28
4.4.5 Transformer model	29
4.4.6 Measurements and controls	31
4.4.7 Regulating control models	35
4.5 Modeling guidelines	35
4.5.1 Modeling for change	35
4.5.2 Process for amendments to the CIM	36
4.5.3 Changes to the CIM UML model	36
4.5.4 Changes to the CIM standards documents	36
4.5.5 CIM profiles	36
4.6 Modeling tools	36
4.7 User implementation conventions	37
4.7.1 Conventions beyond UML	37
4.7.2 Number of Terminals for ConductingEquipment objects	37

4.8	CIM modeling examples	37
5	Detailed model	37
5.1	Overview	37
5.2	Context	38
6	Package architecture (normative)	39
6.1	IEC 61970	39
6.1.1	IEC 61970 package summary	39
6.1.2	IEC 61970CIMVersion	40
6.2	Domain	40
6.2.1	Domain package summary	40
6.2.2	AbsoluteDate Datatype	44
6.2.3	AbsoluteDateTime Datatype	44
6.2.4	ActivePower Datatype	44
6.2.5	ActivePowerChangeRate Datatype	45
6.2.6	Admittance Datatype	45
6.2.7	AngleDegrees Datatype	45
6.2.8	AngleRadians Datatype	45
6.2.9	ApparentPower Datatype	45
6.2.10	Boolean Primitive	45
6.2.11	Capacitance Datatype	46
6.2.12	Conductance Datatype	46
6.2.13	CostPerEnergyUnit Datatype	46
6.2.14	CostRate Datatype	46
6.2.15	Currency enumeration	46
6.2.16	CurrentFlow Datatype	47
6.2.17	Damping Datatype	47
6.2.18	Float Primitive	47
6.2.19	FloatQuantity Datatype	47
6.2.20	Frequency Datatype	48
6.2.21	Hours Datatype	48
6.2.22	Impedance Datatype	48
6.2.23	Inductance Datatype	48
6.2.24	Integer Primitive	48
6.2.25	IntegerQuantity Datatype	49
6.2.26	KWActivePower Datatype	49
6.2.27	LongLength Datatype	49
6.2.28	Minutes Datatype	49
6.2.29	MonetaryAmountPerEnergyUnit enumeration	49
6.2.30	MonetaryAmountPerHeatUnit enumeration	49
6.2.31	MonetaryAmountRate enumeration	50
6.2.32	Money Datatype	50
6.2.33	PerCent Datatype	50
6.2.34	Pressure Datatype	50
6.2.35	PU Datatype	50
6.2.36	Reactance Datatype	51
6.2.37	ReactivePower Datatype	51
6.2.38	RealEnergy Datatype	51
6.2.39	Resistance Datatype	51

6.2.40	RotationSpeed Datatype	51
6.2.41	Seconds Datatype	52
6.2.42	ShortLength Datatype.....	52
6.2.43	String Primitive.....	52
6.2.44	StringQuantity Datatype	52
6.2.45	Susceptance Datatype	52
6.2.46	Temperature Datatype.....	53
6.2.47	UnitMultiplier enumeration.....	53
6.2.48	UnitSymbol enumeration	53
6.2.49	Voltage Datatype.....	54
6.2.50	VoltagePerReactivePower Datatype	54
6.2.51	Volume Datatype.....	54
6.2.52	WaterLevel Datatype.....	55
6.2.53	Weight Datatype.....	55
6.3	Core.....	55
6.3.1	Core package summary	55
6.3.2	BasePower.....	59
6.3.3	BaseVoltage.....	60
6.3.4	BasicIntervalSchedule.....	60
6.3.5	Bay	61
6.3.6	BreakerConfiguration Enumeration.....	62
6.3.7	BusbarConfiguration Enumeration	62
6.3.8	Company	62
6.3.9	CompanyType enumeration	63
6.3.10	ConductingEquipment	63
6.3.11	ConnectivityNodeContainer	64
6.3.12	Curve	64
6.3.13	CurveData.....	65
6.3.14	CurveStyle enumeration	65
6.3.15	Equipment.....	65
6.3.16	EquipmentContainer.....	66
6.3.17	GeographicalRegion.....	67
6.3.18	IdentifiedObject	67
6.3.19	IrregularIntervalSchedule.....	68
6.3.20	IrregularTimePoint.....	69
6.3.21	ModelingAuthority.....	69
6.3.22	ModelingAuthoritySet	69
6.3.23	OperatingParticipant.....	70
6.3.24	OperatingShare.....	70
6.3.25	PhaseCode enumeration.....	71
6.3.26	PowerSystemResource	71
6.3.27	PsrList.....	72
6.3.28	PSRType	73
6.3.29	RegularIntervalSchedule	73
6.3.30	RegularTimePoint.....	74
6.3.31	ReportingGroup.....	74
6.3.32	ReportingSuperGroup	74
6.3.33	SubGeographicalRegion	75
6.3.34	Substation.....	75

6.3.35	Terminal	76
6.3.36	Unit	77
6.3.37	VoltageLevel	78
6.4	OperationalLimits	79
6.4.1	OperationalLimits package summary	79
6.4.2	ActivePowerLimit	80
6.4.3	ApparentPowerLimit	81
6.4.4	BranchGroup	81
6.4.5	BranchGroupTerminal	82
6.4.6	CurrentLimit	82
6.4.7	OperationalLimit	82
6.4.8	OperationalLimitDirectionKind enumeration	83
6.4.9	OperationalLimitSet	83
6.4.10	OperationalLimitType	84
6.4.11	VoltageLimit	84
6.5	Topology	85
6.5.1	Topology package summary	85
6.5.2	BusNameMarker	88
6.5.3	ConnectivityNode	89
6.5.4	TopologicalIsland	90
6.5.5	TopologicalNode	90
6.6	Wires	91
6.6.1	Wires package summary	91
6.6.2	ACLineSegment	101
6.6.3	Breaker	102
6.6.4	BusbarSection	103
6.6.5	CompositeSwitch	104
6.6.6	CompositeSwitchType Datatype	105
6.6.7	Conductor	105
6.6.8	ConductorType	106
6.6.9	Connector	107
6.6.10	CoolantType enumeration	107
6.6.11	DCLineSegment	108
6.6.12	Disconnecter	109
6.6.13	EnergyConsumer	110
6.6.14	EnergySource	111
6.6.15	FrequencyConverter	112
6.6.16	Fuse	113
6.6.17	Ground	114
6.6.18	GroundDisconnecter	114
6.6.19	HeatExchanger	115
6.6.20	Jumper	116
6.6.21	Junction	117
6.6.22	Line	118
6.6.23	LoadBreakSwitch	118
6.6.24	MutualCoupling	119
6.6.25	OperatingMode Datatype	120
6.6.26	Plant	120
6.6.27	PowerTransformer	121

6.6.28	ProtectedSwitch	122
6.6.29	ReactiveCapabilityCurve	123
6.6.30	RectifierInverter	123
6.6.31	RegulatingCondEq	124
6.6.32	RegulatingControl.....	125
6.6.33	RegulatingControlModeKind enumeration.....	126
6.6.34	RegulationSchedule	126
6.6.35	SeriesCompensator.....	127
6.6.36	ShuntCompensator	128
6.6.37	StaticVarCompensator	129
6.6.38	SVCControlMode enumeration	130
6.6.39	Switch	131
6.6.40	SynchronousMachine	131
6.6.41	SynchronousMachineOperatingMode enumeration.....	134
6.6.42	SynchronousMachineType enumeration.....	134
6.6.43	TapChanger	134
6.6.44	TapChangerKind enumeration.....	135
6.6.45	TransformerControlMode enumeration	135
6.6.46	TransformerCoolingType enumeration.....	136
6.6.47	TransformerWinding.....	136
6.6.48	VoltageControlZone	137
6.6.49	WindingConnection enumeration.....	138
6.6.50	WindingTest.....	138
6.6.51	WindingType enumeration.....	139
6.6.52	WireArrangement	139
6.6.53	WireType	140
6.7	Generation – Generation package summary	140
6.8	Production	141
6.8.1	Production package summary	141
6.8.2	AirCompressor	145
6.8.3	CAESPlant.....	146
6.8.4	Classification Datatype	147
6.8.5	CogenerationPlant.....	147
6.8.6	CombinedCyclePlant.....	148
6.8.7	CostPerHeatUnit Datatype	148
6.8.8	Emission Datatype	148
6.8.9	EmissionAccount.....	149
6.8.10	EmissionCurve.....	149
6.8.11	EmissionType enumeration	150
6.8.12	EmissionValueSource enumeration	150
6.8.13	FossilFuel	150
6.8.14	FuelAllocationSchedule.....	151
6.8.15	FuelType enumeration.....	152
6.8.16	GeneratingUnit.....	152
6.8.17	GeneratorControlMode enumeration.....	155
6.8.18	GeneratorControlSource enumeration	155
6.8.19	GeneratorOperatingMode enumeration	155
6.8.20	GenUnitOpCostCurve.....	155

6.8.21	GenUnitOpSchedule.....	156
6.8.22	GrossToNetActivePowerCurve	157
6.8.23	HeatInputCurve	157
6.8.24	HeatRate Datatype.....	158
6.8.25	HeatRateCurve.....	158
6.8.26	HydroEnergyConversionKind enumeration	159
6.8.27	HydroGeneratingEfficiencyCurve	159
6.8.28	HydroGeneratingUnit.....	160
6.8.29	HydroPlantType enumeration	162
6.8.30	HydroPowerPlant	162
6.8.31	HydroPump	163
6.8.32	HydroPumpOpSchedule	164
6.8.33	IncrementalHeatRateCurve	165
6.8.34	InflowForecast.....	166
6.8.35	LevelVsVolumeCurve	166
6.8.36	NuclearGeneratingUnit	167
6.8.37	PenstockLossCurve.....	169
6.8.38	PenstockType enumeration	169
6.8.39	Reservoir.....	169
6.8.40	ShutdownCurve.....	171
6.8.41	SpillwayGateType Enumeration.....	171
6.8.42	StartIgnFuelCurve	171
6.8.43	StartMainFuelCurve.....	172
6.8.44	StartRampCurve.....	173
6.8.45	StartupModel.....	173
6.8.46	SteamSendoutSchedule	174
6.8.47	SurgeTankCode enumeration	175
6.8.48	TailbayLossCurve.....	175
6.8.49	TargetLevelSchedule.....	176
6.8.50	ThermalGeneratingUnit	176
6.9	GenerationDynamics.....	178
6.9.1	GenerationDynamics package summary.....	178
6.9.2	BoilerControlMode enumeration	179
6.9.3	BWRSteamSupply.....	179
6.9.4	CombustionTurbine	181
6.9.5	CTTempActivePowerCurve	182
6.9.6	DrumBoiler	182
6.9.7	FossilSteamSupply.....	184
6.9.8	HeatRecoveryBoiler.....	185
6.9.9	HydroTurbine	186
6.9.10	PrimeMover.....	187
6.9.11	PWRSteamSupply.....	188
6.9.12	SteamSupply.....	189
6.9.13	SteamTurbine.....	189
6.9.14	Subcritical	190
6.9.15	Supercritical	191
6.9.16	TurbineType enumeration.....	193
6.10	LoadModel	193
6.10.1	LoadModel package summary	193

6.10.2	ConformLoad	194
6.10.3	ConformLoadGroup.....	195
6.10.4	ConformLoadSchedule.....	196
6.10.5	CustomerLoad.....	197
6.10.6	DayType	197
6.10.7	EnergyArea	198
6.10.8	InductionMotorLoad.....	198
6.10.9	Load.....	199
6.10.10	LoadArea	200
6.10.11	LoadGroup.....	201
6.10.12	LoadResponseCharacteristic.....	201
6.10.13	NonConformLoad	203
6.10.14	NonConformLoadGroup	203
6.10.15	NonConformLoadSchedule	204
6.10.16	PowerCutZone	205
6.10.17	Season.....	205
6.10.18	SeasonDayTypeSchedule	206
6.10.19	SeasonName enumeration	206
6.10.20	StationSupply	206
6.10.21	SubLoadArea	207
6.11	Outage.....	208
6.11.1	Outage package summary.....	208
6.11.2	ClearanceTag	209
6.11.3	ClearanceTagType.....	210
6.11.4	OutageSchedule	210
6.11.5	SwitchingOperation	211
6.11.6	SwitchState enumeration.....	212
6.12	Protection	212
6.12.1	Protection package summary	212
6.12.2	CurrentRelay	212
6.12.3	ProtectionEquipment	213
6.12.4	RecloseSequence	214
6.12.5	SynchrocheckRelay.....	215
6.13	Equivalents.....	216
6.13.1	Equivalents package summary	216
6.13.2	EquivalentBranch	217
6.13.3	EquivalentEquipment	218
6.13.4	EquivalentNetwork	219
6.13.5	EquivalentShunt	219
6.14	Meas.....	220
6.14.1	Meas package summary	220
6.14.2	Accumulator	224
6.14.3	AccumulatorLimit.....	225
6.14.4	AccumulatorLimitSet	225
6.14.5	AccumulatorValue	225
6.14.6	Analog	226
6.14.7	AnalogLimit	227
6.14.8	AnalogLimitSet.....	227
6.14.9	AnalogValue.....	228

6.14.10 Command	228
6.14.11 Control	229
6.14.12 ControlType.....	230
6.14.13 Discrete.....	230
6.14.14 DiscreteValue.....	231
6.14.15 Limit	231
6.14.16 LimitSet.....	232
6.14.17 Measurement	232
6.14.18 MeasurementType.....	233
6.14.19 MeasurementValue	234
6.14.20 MeasurementValueQuality.....	234
6.14.21 MeasurementValueSource	235
6.14.22 Quality61850	235
6.14.23 SetPoint	236
6.14.24 StringMeasurement	237
6.14.25 StringMeasurementValue	238
6.14.26 Validity enumeration	238
6.14.27 ValueAliasSet.....	239
6.14.28 ValueToAlias	239
6.15 SCADA	240
6.15.1 SCADA package summary	240
6.15.2 CommunicationLink.....	241
6.15.3 RemoteControl	242
6.15.4 RemotePoint	242
6.15.5 RemoteSource	243
6.15.6 RemoteUnit	243
6.15.7 RemoteUnitType enumeration	244
6.15.8 Source enumeration	244
6.16 ControlArea.....	244
6.16.1 ControlArea package summary.....	244
6.16.2 AltGeneratingUnitMeas.....	246
6.16.3 AltTieMeas	247
6.16.4 ControlArea	247
6.16.5 ControlAreaGeneratingUnit.....	248
6.16.6 ControlAreaTypeKind enumeration	248
6.16.7 TieFlow	248
6.17 Contingency.....	249
6.17.1 Contingency package summary.....	249
6.17.2 Contingency	250
6.17.3 ContingencyElement	250
6.17.4 ContingencyEquipment.....	250
6.17.5 ContingencyEquipmentStatusKind enumeration.....	251
Bibliography.....	252
Annex ZA (normative) Normative references to international publications with their corresponding European publications.....	253
Figure 1 – CIM IEC 61970-301 package diagram	19
Figure 2 – Example of generalization	22
Figure 3 – Example of simple association.....	23

Figure 4 – Example of aggregation.....	23
Figure 5 – Equipment containers.....	25
Figure 6 – Connectivity model.....	26
Figure 7 – Simple network example	27
Figure 8 – Simple network connectivity modeled with CIM topology.....	28
Figure 9 – Equipment inheritance hierarchy	29
Figure 10 – Transformer model	30
Figure 11 – Navigating from PSR to MeasurementValue	32
Figure 12 – Measurement placement	34
Figure 13 – Regulating control models	35
Figure 14 – CIM top level packages	38
Figure 15 – Main.....	39
Figure 16 – CombinedDatatypes	40
Figure 17 – BasicDatatypes	40
Figure 18 – ElectricityDatatypes.....	41
Figure 19 – EnumeratedUnitDatatypes.....	42
Figure 20 – GeneralDatatypes	43
Figure 21 – MonetaryDatatypes	43
Figure 22 – TimeDatatypes	44
Figure 23 – Reporting	56
Figure 24 – Main.....	57
Figure 25 – CurveSchedule.....	58
Figure 26 – Datatypes.....	58
Figure 27 – DocumentationExampleAggregation.....	59
Figure 28 – DocumentationExampleAssociation.....	59
Figure 29 – Ownership.....	59
Figure 30 – OperationalLimits	79
Figure 31 – BranchGroup.....	80
Figure 32 – TopologicalNodeTerminal.....	85
Figure 33 – TopologyMeasRelations	86
Figure 34 – TopologyReporting	87
Figure 35 – Main.....	88
Figure 36 – DocumentationExampleInheritance	92
Figure 37 – MutualCoupling	93
Figure 38 – Datatypes.....	94
Figure 39 – InheritanceHierarchy	95
Figure 40 – LineModel	96
Figure 41 – NamingHierarchyPart1	97
Figure 42 – NamingHierarchyPart2	98
Figure 43 – RegulatingEquipment	99
Figure 44 – TransformerModel.....	100
Figure 45 – VoltageControl	101
Figure 46 – Main.....	141

Figure 47 – Nuclear	141
Figure 48 – Main.....	142
Figure 49 – Datatypes.....	143
Figure 50 – Hydro	144
Figure 51 – Thermal.....	145
Figure 52 – Main.....	179
Figure 53 – Datatypes.....	179
Figure 54 – Main.....	194
Figure 55 – Datatypes.....	194
Figure 56 – Datatypes.....	208
Figure 57 – Main.....	209
Figure 58 – Main.....	212
Figure 59 – Main.....	217
Figure 60 – Datatypes.....	220
Figure 61 – Control	221
Figure 62 – InheritanceStructure	222
Figure 63 – Measurement	223
Figure 64 – Quality.....	224
Figure 65 – Datatypes.....	240
Figure 66 – Main.....	241
Figure 67 – ControlArea	245
Figure 68 – ControlAreaInheritance.....	246
Figure 69 – Datatypes.....	246
Figure 70 – Contingency	249
Table 1 – MeasurementType naming conventions.....	33
Table 2 – MeasurementValueSource naming conventions.....	34

INTRODUCTION

This standard is one of several parts 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 principle 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 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 normalized information model comprising the CIM behind all these individual standard 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 free of charge 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
Royaume-Uni (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.

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

Part 301: Common information model (CIM) base

1 Scope

This part of IEC 61970 deals with the common information model (CIM), an abstract model that represents all the major objects in an electric utility enterprise typically involved in utility operations.

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.

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 (supervisory control and data acquisition) 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 and syntax) 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.

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 provides the current grouping of packages into standard documents.

2 Normative references

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

IEC 61850 (all parts), *Communication networks and systems in substations*

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

ISO 8601:2004, *Data elements and interchange formats – Information interchange – Representation of dates and times*

3 Terms and definitions

For the purposes of this document, the terms and definitions of IEC 61970-2 apply, as well as the following.

NOTE Refer to International Electrotechnical Vocabulary, IEC 60050, for general glossary definitions.

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 assure 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

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

4.2.1 CIM packages overview

The CIM is partitioned into a set of packages. A package is a general purpose means of grouping related model elements. There is no specific semantic meaning. 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. These groups are assigned to different working groups within technical committee 57.

WG13 packages include:

a) IEC 61970-301 (this document) with the following packages:

- Domain
- Core
- OperationalLimits
- Topology
- Wires
- Generation

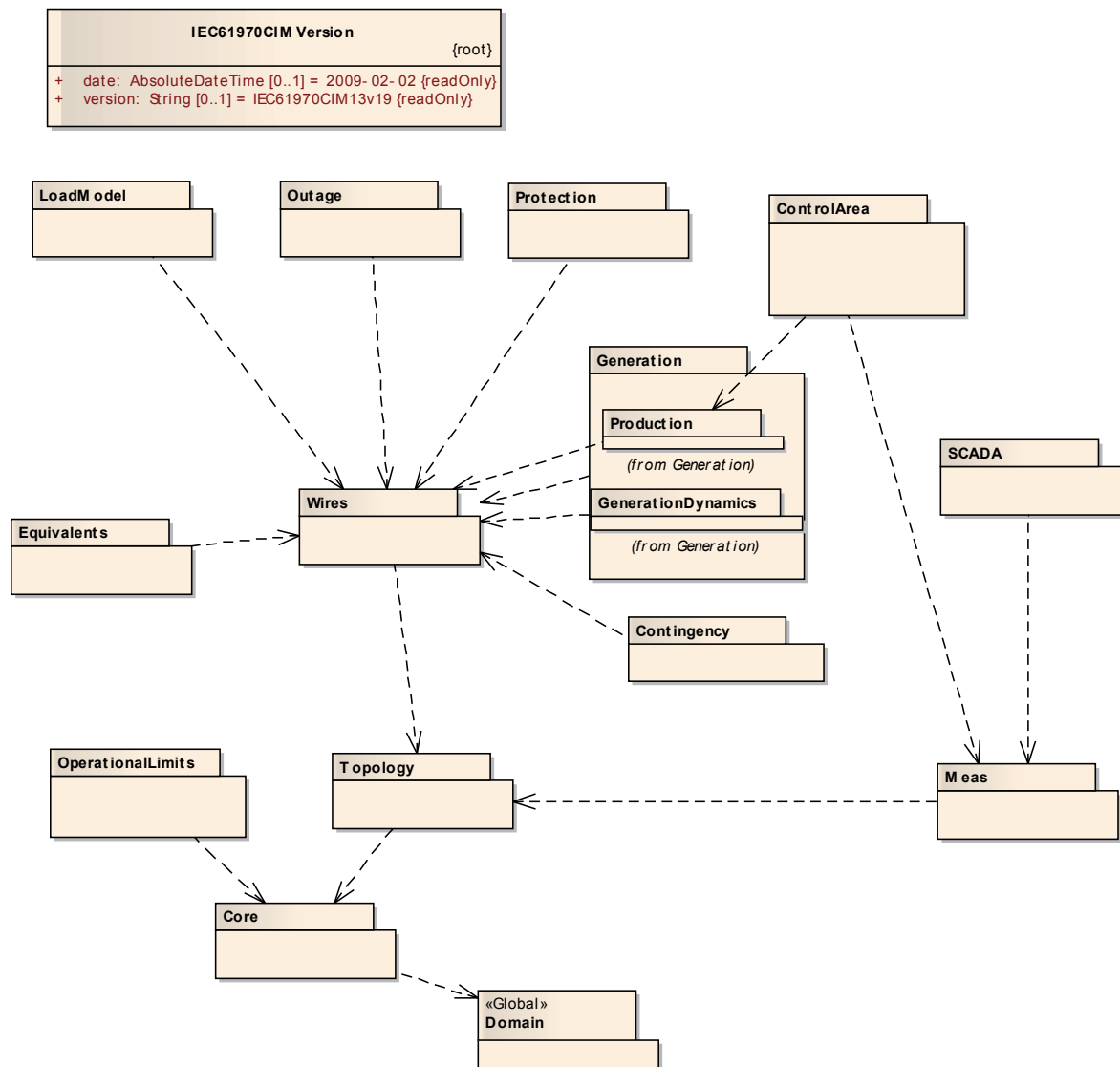
- Generation.GenerationDynamics
- Generation.Production
- LoadModel
- Outage
- Protection
- Equivalents
- Meas
- SCADA
- ControlArea
- Contingency

b) Packages dealing with market operations, such as energy scheduling, financial, and reservations.

WG14 packages in the IEC 61968 series of standards describe additional parts of the CIM that deal with other logical views of utility operations including include assets, location, activities, consumers, documentation, work management, and distribution-specific network models.

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 document will have dependencies upon some of the packages described in this document, and particularly the Domain and Core packages, though other dependencies will also exist.

Figure 1 shows the packages defined for IEC 61970-301 CIM base and their dependency relationships. The dashed line indicates a dependency relationship, with the arrowhead pointing from the dependent package to the package on which it has a dependency.



IEC 1989/11

Figure 1 – CIM IEC 61970-301 package diagram

The subclauses below summarize the contents of each CIM package. Clause 6 contains the specification for each of the CIM packages.

NOTE The contents of the CIM defined in this specification were auto-generated from the CIM UML electronic model release IEC 61970CIM13v19, which is available at through the CIM users group.

4.2.2 Domain

The Domain package is a data dictionary of quantities and units that define datatypes for attributes (properties) that may be used by any class in any other package.

This package contains the definition of primitive datatypes, including units of measure and permissible values. Each datatype contains a value attribute and an optional unit of measure, which is specified as a static variable initialized to the textual description of the unit of measure. The value of the "units" string may be country or customer specific. Typical values are given. Permissible values for enumerations are listed in the documentation for the attribute using UML constraint syntax inside curly braces. Lengths of variable strings are listed in the descriptive text where required.

4.2.3 Core

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.

4.2.4 OperationalLimits

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

4.2.5 Topology

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.

4.2.6 Wires

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.

4.2.7 Generation

Generation is divided into two separate packages, Production and Generation, which are described below.

4.2.7.1 Generation.Production

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.

4.2.7.2 Generation.GenerationDynamics

The GenerationDynamics package contains prime movers, such as turbines and boilers, which are needed for simulation and educational purposes.

4.2.8 LoadModel

This package is responsible for modeling the energy consumers and the system load as curves and associated curve data. Special circumstances that may affect the load, such as seasons and daytypes, are also included here.

This information is used by Load Forecasting and Load Management.

4.2.9 Outage

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.

4.2.10 Protection

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.

4.2.11 **Equivalents**

The equivalents package models equivalent networks.

4.2.12 **Meas**

Contains entities that describe dynamic measurement data exchanged between applications.

4.2.13 **SCADA**

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 IEC 61850 definitions.

This package also supports alarm presentation but it is not expected to be used by other applications.

4.2.14 **ControlArea**

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.

4.2.15 **Contingency**

Contingencies to be studied.

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 are also shown with a note identifying the package which owns the class.

Classes and objects model what is in a power system that needs to be represented in a common way to EMS 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 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 behavior. In order to make the CIM as generic as possible, it is highly desirable to make it easy to configure for specific implementations. In general, it is easier to change the value or domain of an attribute than to change a class definition. These principles imply that the CIM should avoid defining too many specific sub-types of classes. Instead, the CIM defines generic classes with attributes giving the type name. Applications may then use this information to instantiate specific object types as required. Applications may need additional information to define the set of valid types and relationships.

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, float, boolean, string, and enumeration, 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. The definition of data types is contained in the Domain package described in Subclause 4.2.2. The

UML stereotypes of Primitive, enumeration, and Datatype are added to classes used as types. The Datatype 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.

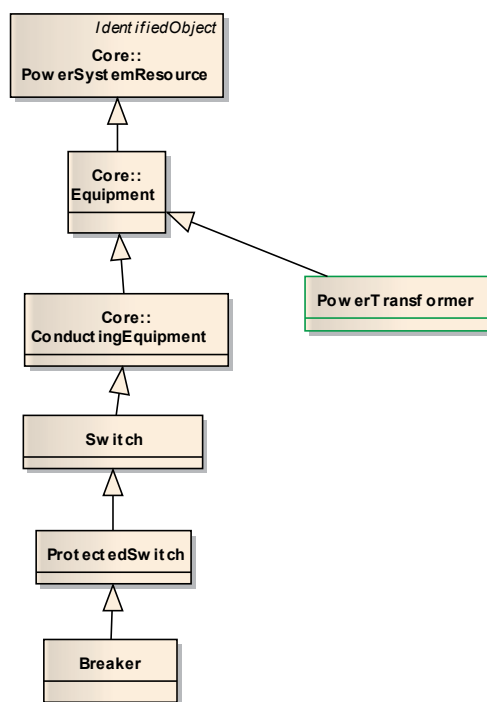
All CIM attributes are implicitly optional in the sense that profiles using the CIM may eliminate any attributes.

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. For example, a PowerTransformer is a specific type of Equipment. Generalization provides for the specific class to inherit attributes and relationships from all the more general classes above it.

Figure 2 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 Equipment. Note that PowerSystemResource inherits from class IdentifiedObject which is not on the diagram.



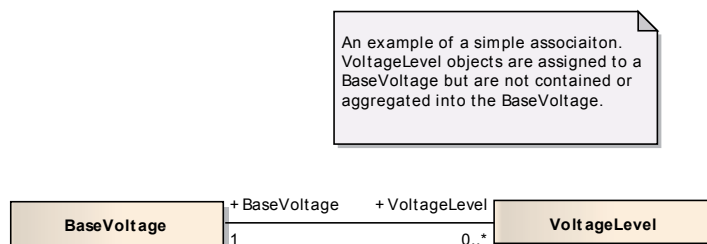
IEC 1990/11

Figure 2 – 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, in the CIM there is an association between a BaseVoltage and a VoltageLevel (See Figure 3 which is taken from the Wires package). 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.

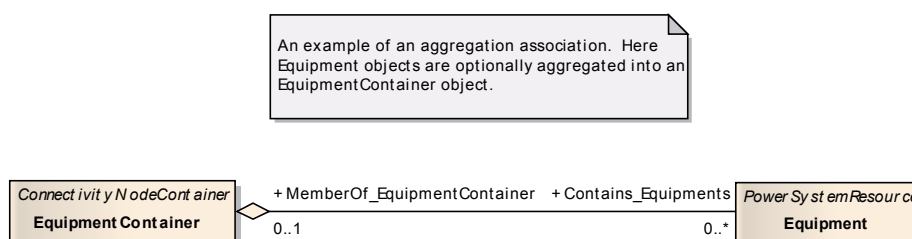


IEC 1991/11

Figure 3 – 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 4 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.



IEC 1992/11

Figure 4 – 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 5 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.

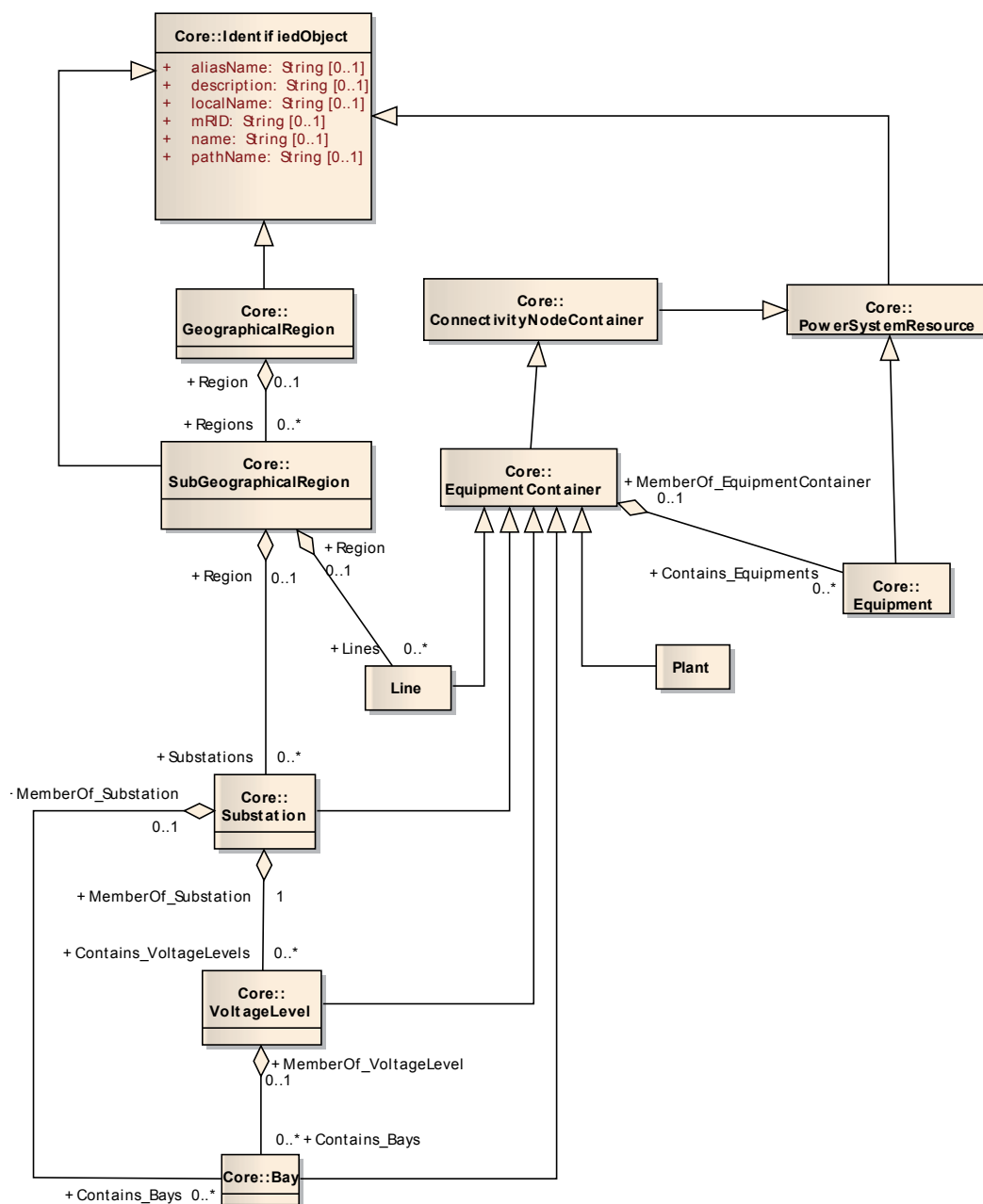
One containment hierarchy is used with the IdentifiedObject class to create hierarchical naming intended for human consumption. One hierarchy is specifically used to name equipment according to its function in the power system. This is the functional naming hierarchy. Other common identifications beside 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 contains six attributes to be used for naming all PowerSystemResource objects. It is intended that values should be assigned to each attribute in a manner consistent with the attribute definitions.

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):

- **name:** A free text name.
- **localName:** A local short name of the instance. Objects that are structured in a functional naming hierarchy have this name local to each particular level in the hierarchy. The name shall be unique among objects contained by the same parent.
- **pathName:** Objects that are structured in a functional naming hierarchy have a pathName which contains all the IdentifiedObject.localNames from the object to the root. The pathName, then, is a concatenation of all these names from the leaf object up to the root of the containment hierarchy, similar to a file path name. For example, if node "A" contains node "B" that contains node "C", then the pathName for node "C" may look like "A.B.C". The type of delimiters used between localNames is not specified but is a local implementation issue.
- **aliasName:** A free text name of the instance. This attribute can also be used for localization.
- **description:** A free format description of the instance.
- **mRID (Master Resource ID):** A globally unique machine-readable identifier for an object instance.



IEC 1993/11

Figure 5 – Equipment containers

The diagrams NamingHierarchyPart1 and NamingHierarchyPart2 in the Wires package show the functional naming hierarchy (refer to the Wires package documentation in Clause 6 for the details).

4.4.3 Connectivity model

4.4.3.1 Connectivity description

Figure 6 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.

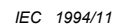


Figure 6 – Connectivity model

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.

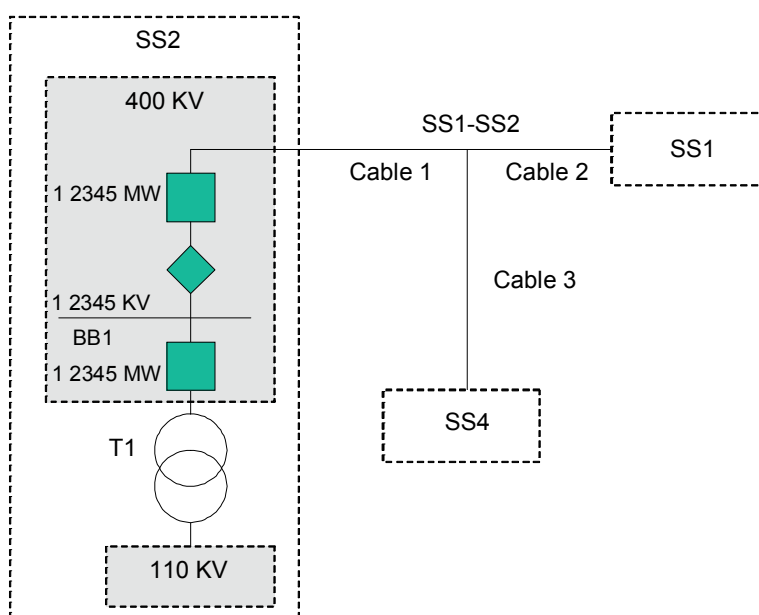
EquipmentContainers, which are a specialization of a ConnectivityNodeContainer, may contain zero or more ConnectivityNodes. The associations, ConductingEquipment – Terminal and Terminal – ConnectivityNode, capture the as built topology of an actual power system network. For each Terminal connected to 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.

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 6, 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. In a study context, the measurement values would have a calculation source instead.

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

4.4.3.2 Connectivity and containment example

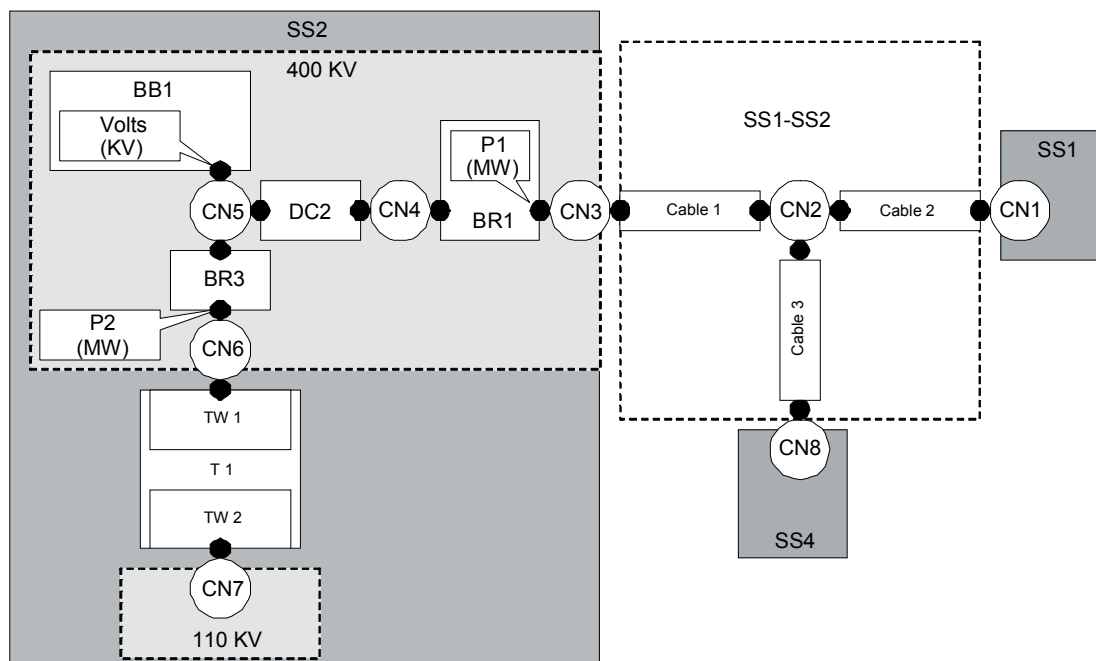
To illustrate how the connectivity model and containment model would appear as objects, a small example is presented in Figure 7. The example shows a transmission line with a T-junction spanning two substations and a substation having two voltage levels with a transformer between them. The transmission 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.



IEC 1995/11

Figure 7 – Simple network example

Figure 8 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 8. 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.



IEC 1996/11

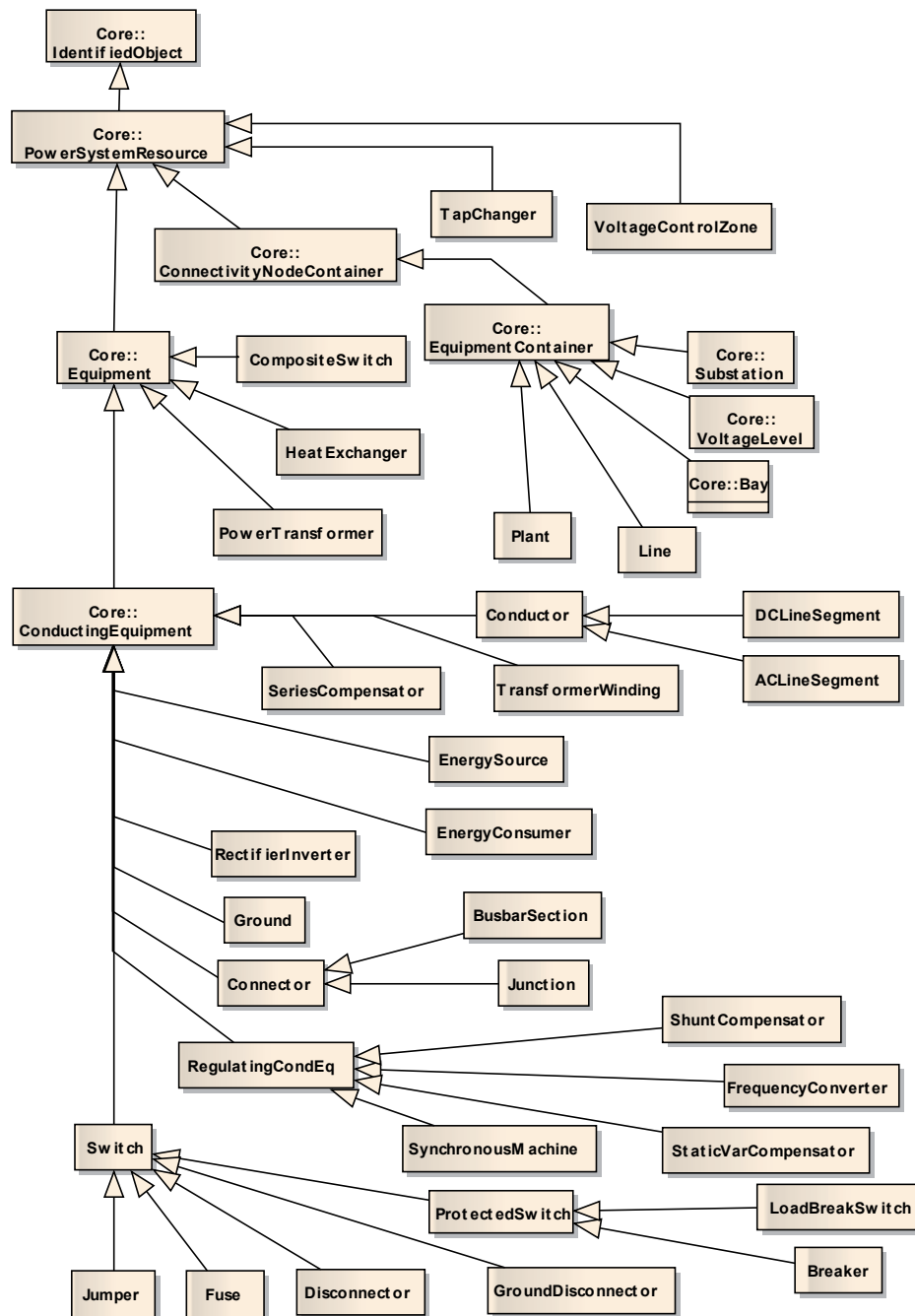
Figure 8 – Simple network connectivity modeled with CIM topology

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 400KV. 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 400KV, which means it belongs to the 400KV VoltageLevel instead of BR3.

4.4.4 Inheritance hierarchy

Figure 9 shows an overview of the inheritance hierarchy modeled in the CIM. This overview, which is included as one of the Wires package diagrams, actually spans most of the CIM packages.



IEC 1997/11

Figure 9 – Equipment inheritance hierarchy

4.4.5 Transformer model

Figure 10 shows a portion of the Wires class diagram which models a PowerTransformer device.

As shown, a PowerTransformer is a specialized class of Equipment, which is a specialized class of a PowerSystemResource, as is ConductingEquipment and TapChanger. This is shown by the use of the generalization-type of relationship, which uses an arrow to point to the general class, and permits the PowerTransformer to inherit attributes from both Equipment and PowerSystemResource.

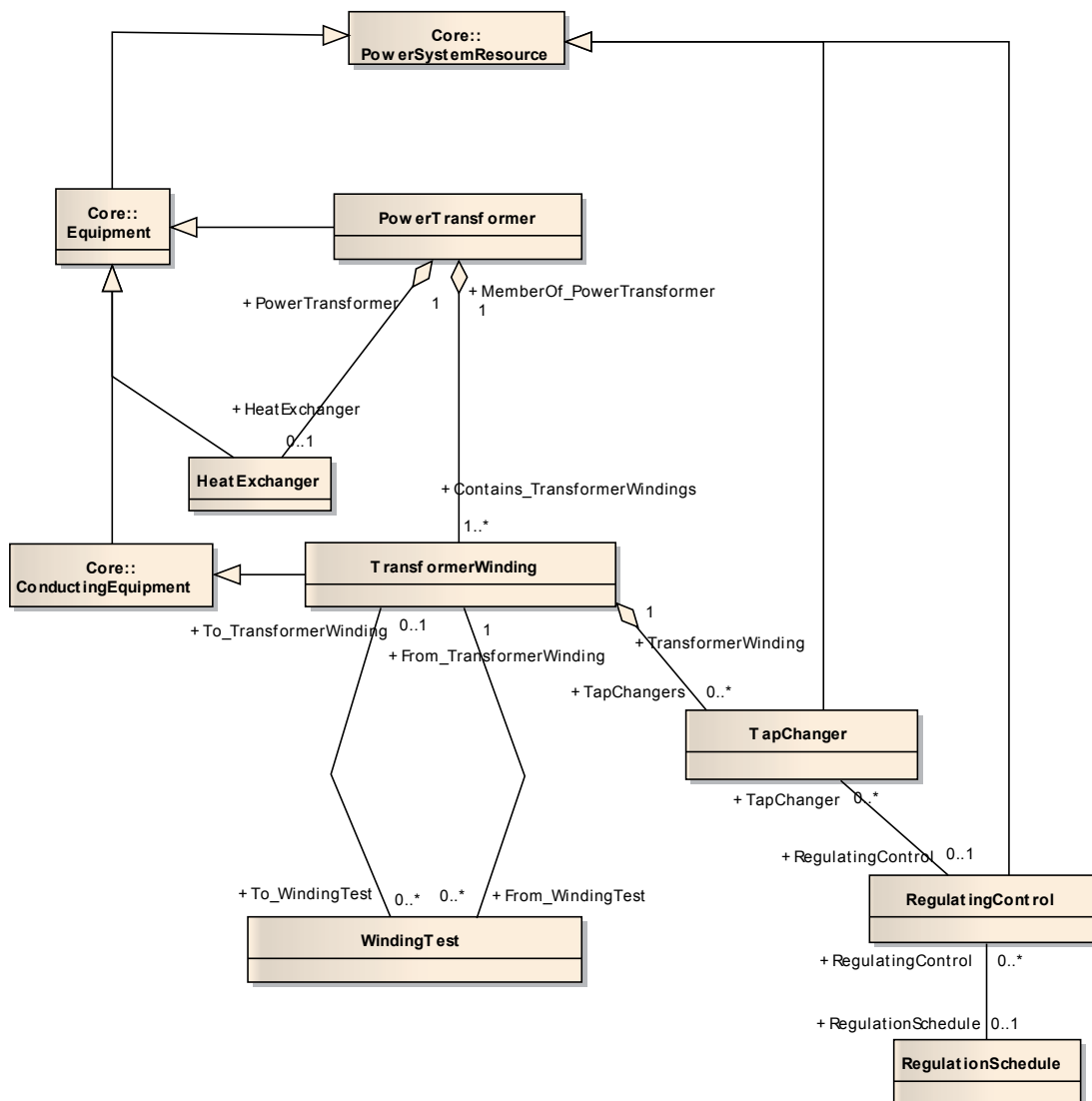
A PowerTransformer also has a relationship to TransformerWinding, which is modeled with an aggregation-type of relationship using a diamond symbol to point from the part class to the whole class. As shown, a

PowerTransformer may have (or contain) one or more TransformerWindings, but a TransformerWinding may belong to (or be a member of) only one PowerTransformer.

The TransformerWinding has other relationships as well:

- a generalization relationship with ConductingEquipment;
- an association relationship with the WindingTest class, such that a TransformerWinding object may be TestedFrom from 0, 1, or more WindingTest objects;
- an aggregation relationship with the TapChanger class, such that a TransformerWinding object may have 0, 1, or more TapChanger objects associated with it.

Clause 6 contains a complete description of each class in Figure 10 along with the definition of all the attributes and relationships supported in each class.



IEC 1998/11

Figure 10 – Transformer model

4.4.6 Measurements and controls

4.4.6.1 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.

4.4.6.2 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 etc. The Meas package supports control variables with the Control model.

4.4.6.3 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 11).

- a) Measurements of a PowerSystemResource are classified by MeasurementType.
- b) MeasurementValues of a Measurement are classified by MeasurementValueSource.
MeasurementType¹ inherits from IdentifiedObject. The values to be used for MeasurementType.name, MeasurementType.aliasName, and MeasurementType.description are specified in Table 1.
- c) MeasurementValueSource also inherits from IdentifiedObject. The values to be used for MeasurementValueSource.name and MeasurementValueSource.description are given in Table 2. 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 must 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. 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.

¹ MeasurementType is different from Unit, which is part of the Domain package described in 4.2.2. MeasurementType describes "what" is measured rather than the unit of measure including scaling, which is the purpose of Unit.

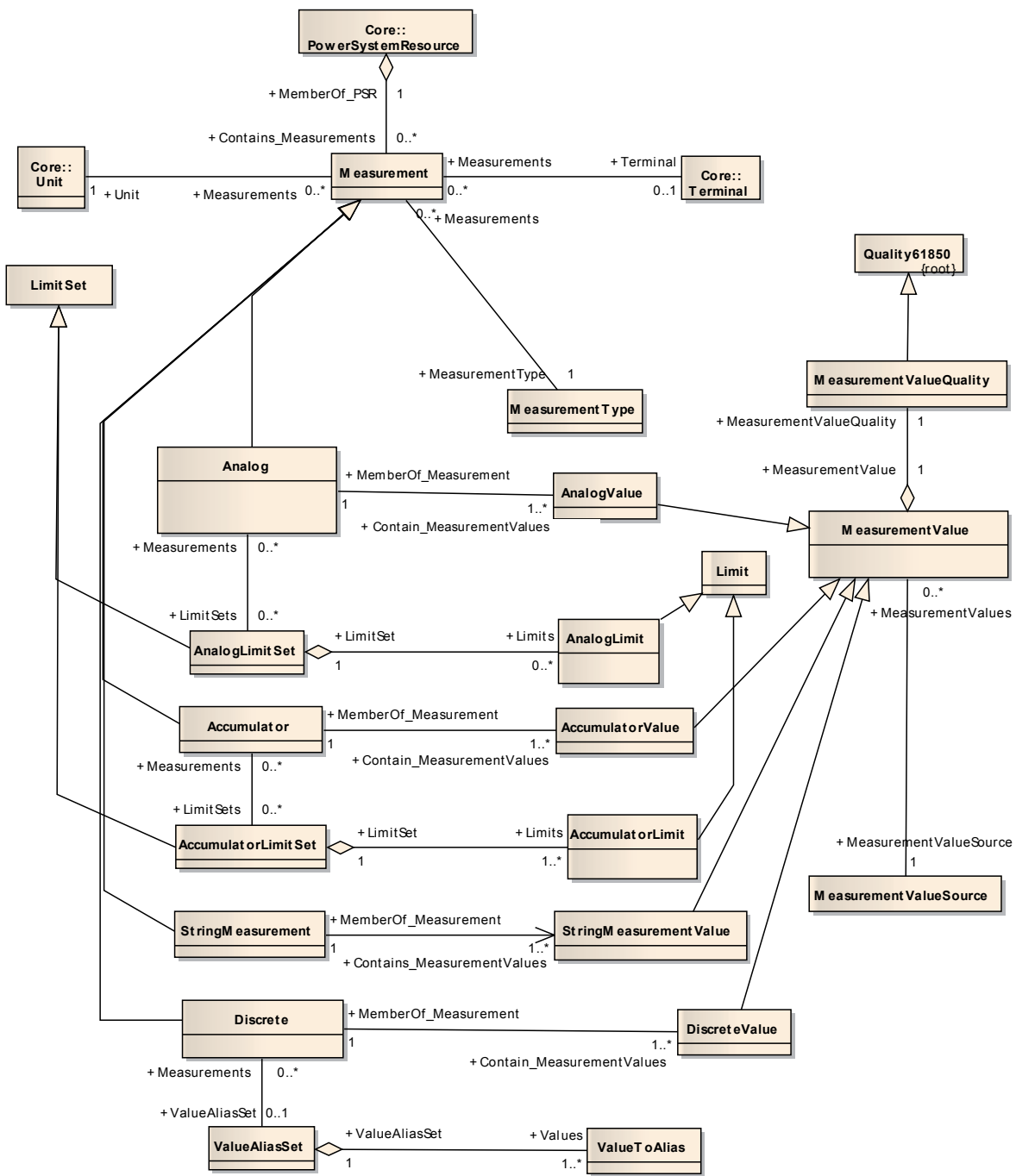


Figure 11 – Navigating from PSR to MeasurementValue

IEC 1999/11

Table 1 – MeasurementType naming conventions

name	aliasName	description
Current	Amp	Current (rms) of a non-three phase circuit
ThreePhaseCurrent	AvAmps	Total current (rms) 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 (rms) 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

Table 1 describes various types of measurements also defined in IEC 61850. The meaning of the columns in Table 1 are as follows:

- name (MeasurementType.name) is the IEC 61970 measurement type name.
- aliasName (MeasurementType.aliasName) is the name assigned to the data in IEC 61850. (Refer to Clause 6 of IEC 61850-7-4, data object name semantics).
- description (MeasurementType.description) of the data.

It shall be noted that Table 1 is a non exhaustive list and that the mapping between measurements as defined in a control center and a substation is non-trivial.

Table 2 – 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 center 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:

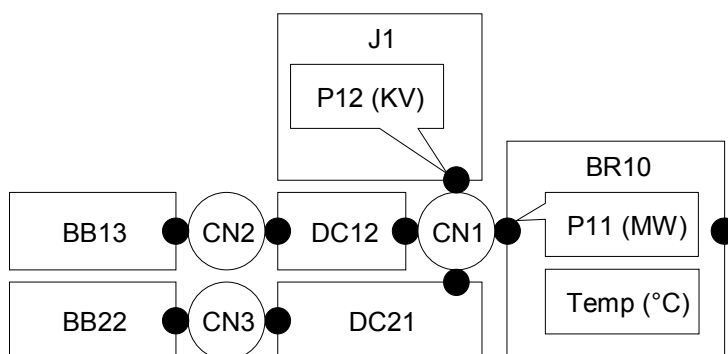
- each Measurement instance represents a technological quantity of a PowerSystemResource;
- each MeasurementValue of a Measurement represents a value for the technological quantity, as supplied from a single source;
- the source attribute in MeasurementValueQuality then indicates whether the source actually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

4.4.6.4 Attachment of measurements

As mentioned in the previous subclause and as shown in Figure 11, 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 must be attached such the flow direction is evident from the placement.

Figure 12 shows two examples of the placement of Measurements.



IEC 2000/11

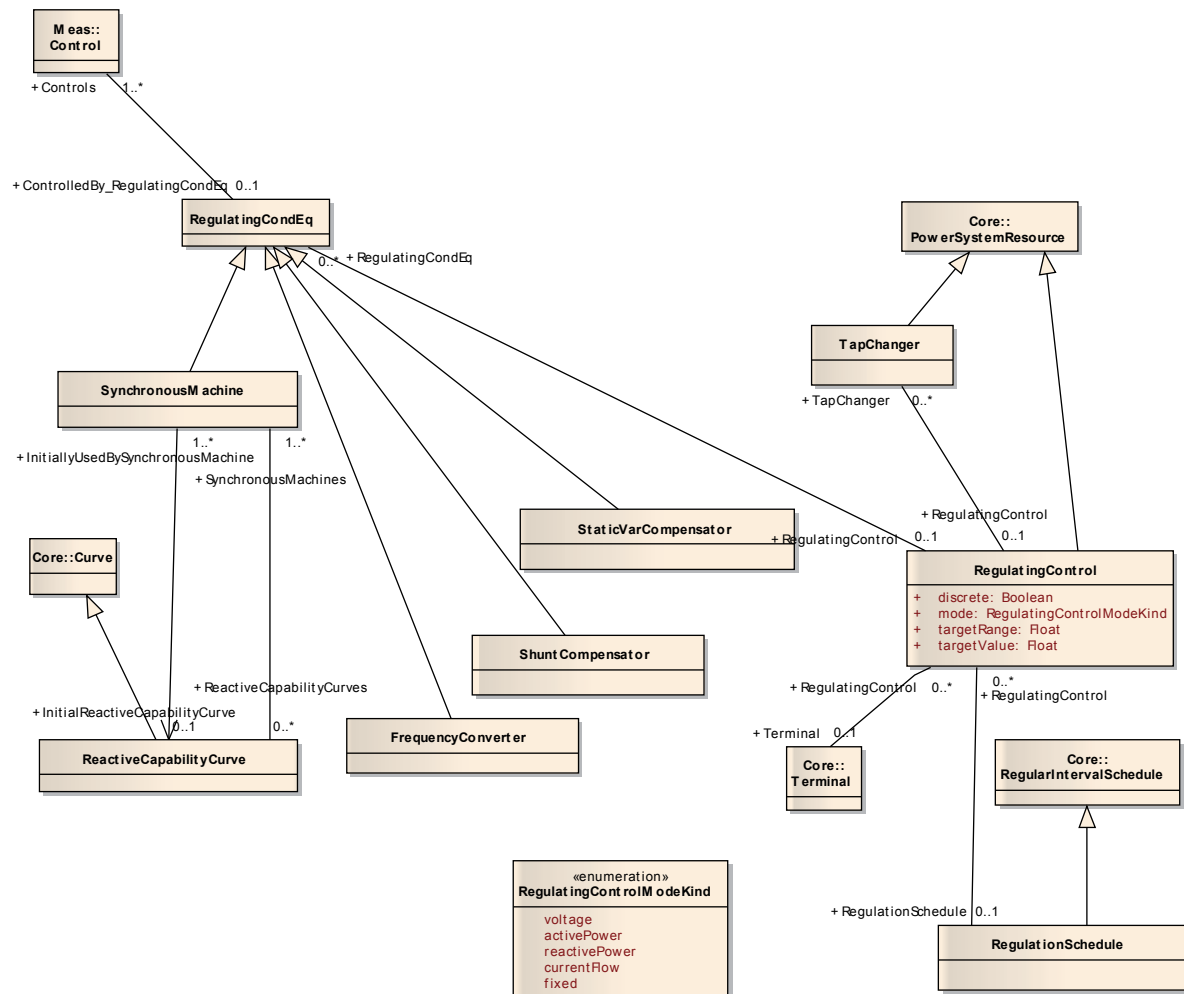
Figure 12 – Measurement placement

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.

4.4.7 Regulating control models

Regulation control, such as automatic voltage control at generators or voltage tap control at transformers is modeled using a class named RegulationControl. The RegulationControl class provides the capability to model multiple 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 13 shows how RegulatingCondEq and TapChanger classes can participate in regulation.



IEC 2001/11

Figure 13 – Regulating control models

4.5 Modeling guidelines

4.5.1 Modeling for change

The following subclauses 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 a Use Case(s) 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) is 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 or associations to existing classes;
- adding new classes that are specializations of existing classes;
- adding new classes via associations with existing classes.

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.

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 Part 3xx specification is required. In either case, the extensions will then become part of the IEC standard CIM.

4.5.5 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 Part 4xx series of standards.

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 tools such as the open source CIMTool, available at <http://www.CIMTool.org>.

4.6 Modeling tools

Previous editions of CIM were created and maintained using Rational ROSE², but this release was constructed using Enterprise Architect product. The entire CIM UML model exists as an Enterprise Architect

² Rational ROSE is the trade name of a product previously supplied by IBM. This information is given for the convenience of users of this document and does not constitute an endorsement by the IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

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 document was auto-generated using a publicly available tool called CIMinEA which uses the automation interfaces of Microsoft Word and Enterprise Architect and the standard IEC format and styles.

4.7 User implementation conventions

4.7.1 Conventions beyond UML

This 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, Jumper, Fuse, Breaker, Disconnecter, LoadBreakSwitch, SeriesCompensator. All other ConductingEquipment leaf classes 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 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 Areva 60 bus model, and the ABB 40 bus model files⁴ used for CIM XML interoperability testing.

5 Detailed model

5.1 Overview

The common information model (CIM) represents a comprehensive logical view of energy management system information. This definition includes the public classes and attributes, as well as the relationships

³ Enterprise Architect is the trade name 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 the IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

⁴ Siemens100 bus model, Areva 60 bus model, and ABB 40 model are the trade names of products supplied by Siemens, Areva, 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.

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. The Domain package defines datatypes used by the other packages. The Generation package is subdivided into Production and GenerationDynamics packages. Classes within the packages are listed alphabetically. Native class attributes are listed first, followed by inherited attributes in order of depth of inheritance, then by attribute name. Native associations are listed first for each class, followed by inherited associations in order of depth of inheritance, then alphabetically by class name, 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 14 shows the top level packages included in this document.



Figure 14 – 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 below. For any inherited attributes or association ends the “note” column will contain text indicating the attributes is inherited from a specific class. The note column for native attributes and association ends contains the actual description.

Attributes

name	type	note
native1	Float	A floating point native attribute of the class is described here
native2	ActivePower	Documenation for another native attribute of type ActivePower
Name	Float	Inherited from class IdentifiedObject

In the Attributes table, in some cases, an attribute is a constant, in which case the phrase “(const)” is added in the name column of the attributes table. In such cases, the attribute normally has an initial value also which is preceded by an equal sign and appended to the attribute name.

Association ends

Mult from	[Mult] name (type)	note
1..*	[0..1] ModelingAuthoritySet (ModelingAuthoritySet)	inhertted from class IdentifiedObject
0..*	[0..*] OperatedBy_Companies (Company)	inherited from class PowerSystemResource
1	[0..*] Contains_Measurements (Measurement)	inherited from class PowerSystemResource
1	[0..*] OperatingShare (OperatingShare)	inhertted from class PowerSystemResource

In the Association Ends table, the first column describes the multiplicity at the other end of the association (i.e., the multiplicity of the class for which association ends are being described). The multiplicity of the association end itself is included in brackets. The association end name is listed in plain text. The class at the other end of the association is within the parentheses. 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, the Attributes table is replaced by the Enums table as shown below, since enumeration literals have no type. There are no inherited enumeration literals for an enumeration class.

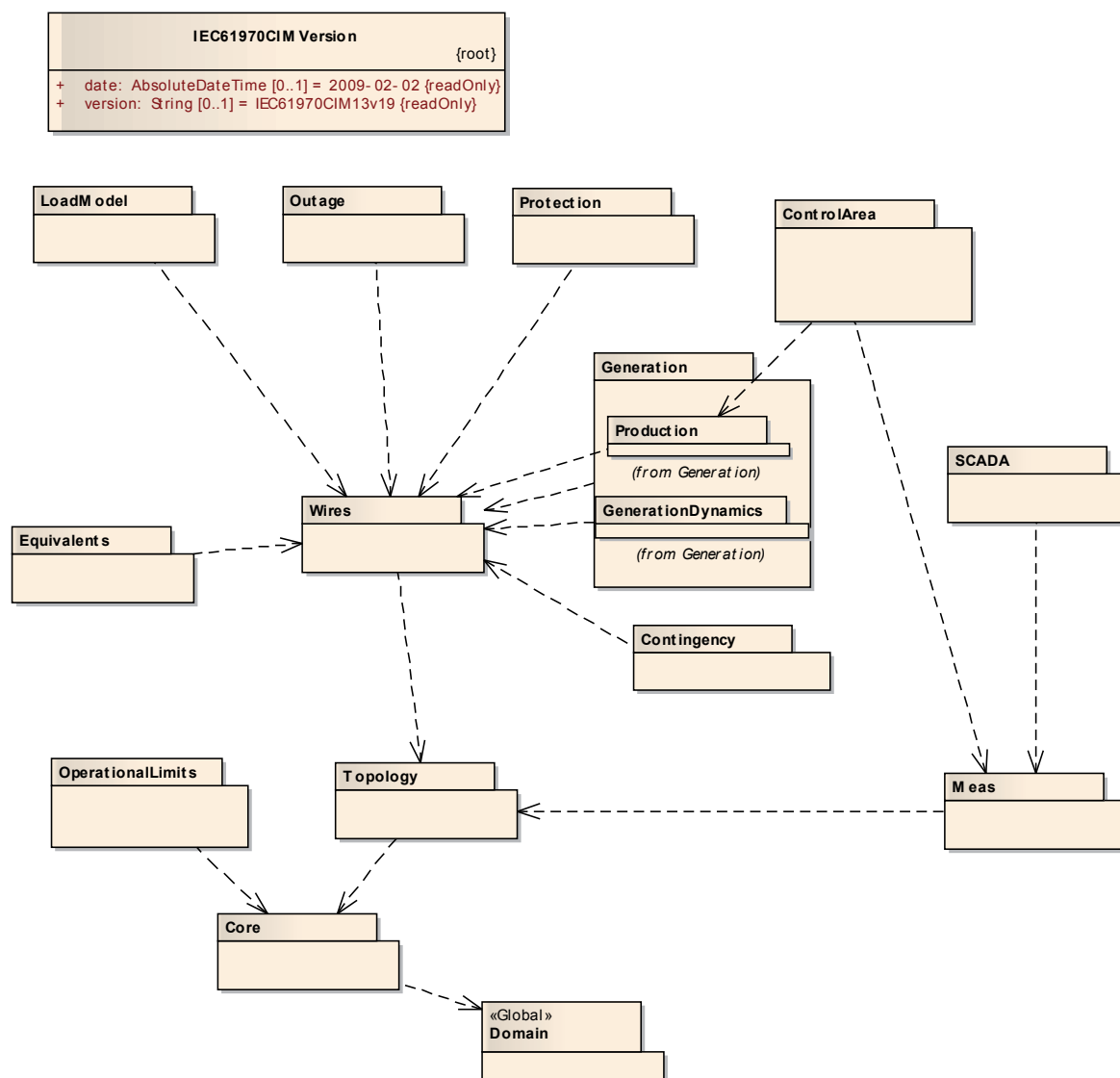
Enums

name	note
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 Package architecture (normative)

6.1 IEC 61970

6.1.1 IEC 61970 package summary



IEC 2003/11

Figure 15 – Main

Figure 15 is documented as follows. This diagram shows all 61970 packages and their logical dependencies.

6.1.2 IEC 61970CIMVersion

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

Attributes

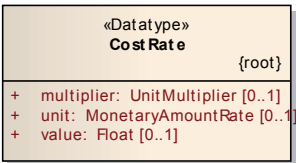
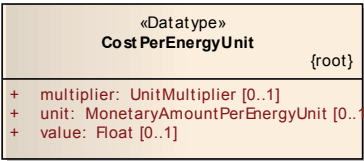
name	type	note
date=2009-02-02 (const)	AbsoluteDateTime	Form is YYYY-MM-DD for example for January 5, 2009 it is 2009-01-05.
version=IEC 61970CIM13v19 (const)	String	Form is IEC 61970CIMXXvYY where XX is the major CIM package version and the YY is the minor version. For example IEC 61970CIM13v18.

6.2 Domain

6.2.1 Domain package summary

The Domain package is a data dictionary of quantities and units that define datatypes for attributes (properties) that may be used by any class in any other package.

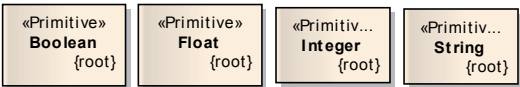
This package contains the definition of primitive datatypes, including units of measure and permissible values. Each datatype contains a value attribute and an optional unit of measure, which is specified as a static variable initialized to the textual description of the unit of measure. The value of the "units" string may be country or customer specific. Typical values are given. Permissible values for enumerations are listed in the documentation for the attribute using UML constraint syntax inside curly braces. Lengths of variable strings are listed in the descriptive text where required.



IEC 2004/11

Figure 16 – CombinedDatatypes

Figure 16 is documented as follows. This diagram includes the datatypes that are combined from datatypes in the other diagrams.



IEC 2005/11

Figure 17 – BasicDatatypes

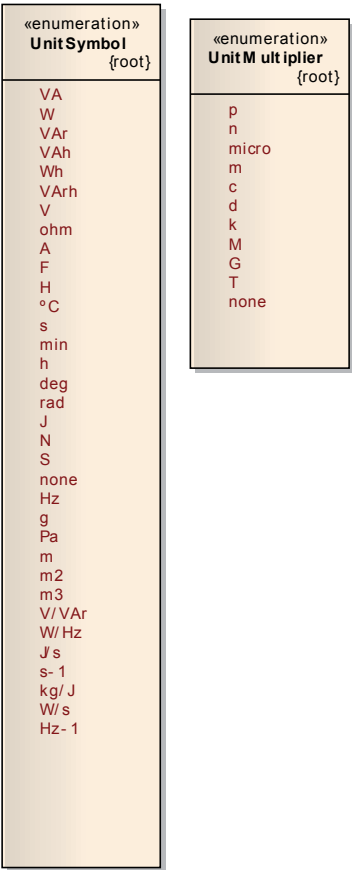
Figure 17 is documented as follows. This diagram shows basic datatypes that are the basis for all other datatypes.

«Datatype» ActivePower {root} + multiplier: UnitMultiplier [0..1] + unit: UnitSymbol [0..1] = W {readOnly} + value: Float [0..1]	«Datatype» KVAActivePower {root} + multiplier: UnitMultiplier [0..1] = k {readOnly} + unit: UnitSymbol [0..1] = W {readOnly} + value: Float [0..1]	«Datatype» ActivePowerChangeRate {root} + multiplier: UnitMultiplier [0..1] + unit: UnitSymbol [0..1] = W/s {readOnly} + value: Float [0..1]
«Datatype» ApparentPower {root} + multiplier: UnitMultiplier [0..1] + unit: UnitSymbol [0..1] = VA {readOnly} + value: Float [0..1]	«Datatype» ReactivePower {root} + multiplier: UnitMultiplier [0..1] + unit: UnitSymbol [0..1] = VAR {readOnly} + value: Float [0..1]	«Datatype» PU {root} + multiplier: UnitMultiplier [0..1] = none {readOnly} + unit: UnitSymbol [0..1] = none {readOnly} + value: Float [0..1]
«Datatype» CurrentFlow {root} + multiplier: UnitMultiplier [0..1] + unit: UnitSymbol [0..1] = A {readOnly} + value: Float [0..1]	«Datatype» Susceptance {root} + multiplier: UnitMultiplier [0..1] + unit: UnitSymbol [0..1] = S {readOnly} + value: Float [0..1]	«Datatype» Capacitance {root} + multiplier: UnitMultiplier [0..1] + unit: UnitSymbol [0..1] = F {readOnly} + value: Float [0..1]
«Datatype» Impedance {root} + multiplier: UnitMultiplier [0..1] + unit: UnitSymbol [0..1] = ohm {readOnly} + value: Float [0..1]	«Datatype» Resistance {root} + multiplier: UnitMultiplier [0..1] + unit: UnitSymbol [0..1] = ohm {readOnly} + value: Float [0..1]	«Datatype» Admittance {root} + multiplier: UnitMultiplier [0..1] + unit: UnitSymbol [0..1] = S {readOnly} + value: Float [0..1]
«Datatype» Conductance {root} + multiplier: UnitMultiplier [0..1] + unit: UnitSymbol [0..1] = S {readOnly} + value: Float [0..1]	«Datatype» Reactance {root} + multiplier: UnitMultiplier [0..1] + unit: UnitSymbol [0..1] = ohm {readOnly} + value: Float [0..1]	«Datatype» Inductance {root} + multiplier: UnitMultiplier [0..1] + unit: UnitSymbol [0..1] = H {readOnly} + value: Float [0..1]
«Datatype» RealEnergy {root} + multiplier: UnitMultiplier [0..1] + unit: UnitSymbol [0..1] = Wh {readOnly} + value: Float [0..1]	«Datatype» Voltage {root} + multiplier: UnitMultiplier [0..1] + unit: UnitSymbol [0..1] = V {readOnly} + value: Float [0..1]	«Datatype» VoltagePerReactivePower {root} + multiplier: UnitMultiplier [0..1] + unit: UnitSymbol [0..1] = V/VAR {readOnly} + value: Float [0..1]

IEC 2006/11

Figure 18 – ElectricityDatatypes

Figure 18 is documented as follows. This diagram shows electricity-related data types.



IEC 2007/11

Figure 19 – EnumeratedUnitDatatypes

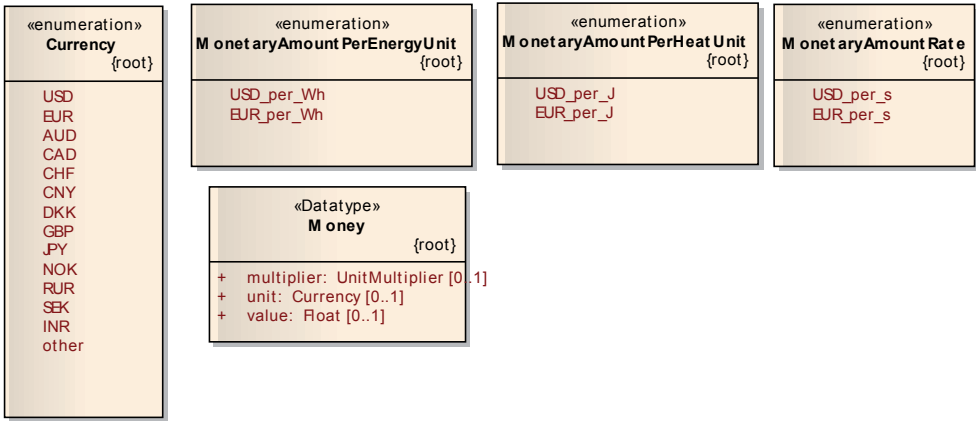
Figure 19 is documented as follows. This diagram shows enumerated data types related to the SI unit system.



IEC 2008/11

Figure 20 – GeneralDatatypes

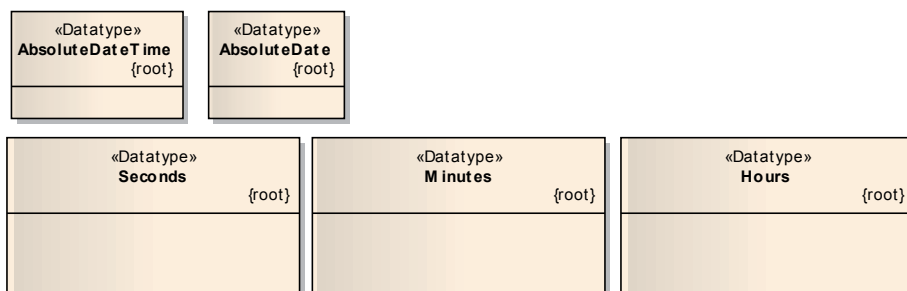
Figure 20 is documented as follows. This diagram shows general related data types.



IEC 2009/11

Figure 21 – MonetaryDatatypes

Figure 21 is documented as follows. This diagram shows monetary related data types.



IEC 2010/11

Figure 22 – TimeDatatypes

Figure 22 is documented as follows. This diagram shows time-related data types.

6.2.2 AbsoluteDate Datatype

Date and time as "yyyy-mm-dd", which conforms with ISO 8601. UTC time zone is specified as "yyyy-mm-dd".

Attributes

name	type	note
value	String	String representation of date, refer to description of the class.

6.2.3 AbsoluteDateTime Datatype

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".

AbsoluteDateTime can be used both for calendar time, e.g. 2007-02-07T10:30, and for relative time, e.g. 10:30.

Attributes

name	type	note
value	String	String representation of date and time, refer to description of the class.

6.2.4 ActivePower Datatype

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

Attributes

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

6.2.5 ActivePowerChangeRate Datatype**Attributes**

name	type	note
value	Float	
unit=W/s (const)	UnitSymbol	
multiplier	UnitMultiplier	

6.2.6 Admittance Datatype

Ratio of current to voltage.

Attributes

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

6.2.7 AngleDegrees Datatype

Measurement of angle in degrees.

Attributes

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

6.2.8 AngleRadians Datatype

Phase angle in radians.

Attributes

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

6.2.9 ApparentPower Datatype

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

Attributes

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

6.2.10 Boolean Primitive

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

6.2.11 Capacitance Datatype

A farad, the typical unit, is the capacitance in which a charge of 1 coulomb produces 1 V potential difference between its terminals.

Attributes

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

6.2.12 Conductance Datatype

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

Attributes

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

6.2.13 CostPerEnergyUnit Datatype

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

Attributes

name	type	note
value	Float	
unit	MonetaryAmountPerEnergyUnit	
multiplier	UnitMultiplier	

6.2.14 CostRate Datatype

Cost, in units of currency, per hour of elapsed time.

Attributes

name	type	note
value	Float	
unit	MonetaryAmountRate	
multiplier	UnitMultiplier	

6.2.15 Currency enumeration

Monetary currencies. Apologies for this list not being exhaustive.