

DIN EN 61970-301

ICS 33.040.20; 35.240.50

Einsprüche bis 2013-07-13
Vorgesehen als Ersatz für
DIN EN 61970-301:2012-03

Entwurf**Schnittstelle für Anwendungsprogramme für Netzführungssysteme
(EMS-API) –****Teil 301: Allgemeines Informationsmodell (CIM), Basismodell****(IEC 57/1331/FDIS:2013);****Englische Fassung FprEN 61970-301:2013**

Energy Management System Application Program Interface (EMS-API) –

Part 301: Common Information Model (CIM) Base

(IEC 57/1331/FDIS:2013);

English version FprEN 61970-301:2013

Interface de programmation d'application pour système de gestion d'énergie (EMS-API) –

Partie 301: Base de modèle d'information commun (CIM)

(CEI 57/1331/FDIS:2013);

Version anglaise FprEN 61970-301:2013

Anwendungswarnvermerk

Dieser Norm-Entwurf mit Erscheinungsdatum 2013-05-13 wird der Öffentlichkeit zur Prüfung und
Stellungnahme vorgelegt.

Weil die beabsichtigte Norm von der vorliegenden Fassung abweichen kann, ist die Anwendung dieses
Entwurfes besonders zu vereinbaren.

Stellungnahmen werden erbeten

- vorzugsweise online im Norm-Entwurfs-Portal des DIN unter www.entwuerfe.din.de bzw. für Norm-Entwürfe der DKE auch im Norm-Entwurfs-Portal der DKE unter www.entwuerfe.normenbibliothek.de, sofern dort wiedergegeben;
- oder als Datei per E-Mail an dke@din.de möglichst in Form einer Tabelle. Die Vorlage dieser Tabelle kann im Internet unter www.din.de/stellungnahme oder für Stellungnahmen zu Norm-Entwürfen der DKE unter www.dke.de/stellungnahme abgerufen werden;
- oder in Papierform an die DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE (Hausanschrift: Stresemannallee 15, 60596 Frankfurt am Main).

Die Empfänger dieses Norm-Entwurfs werden gebeten, mit ihren Kommentaren jegliche relevanten
Patentrechte, die sie kennen, mitzuteilen und unterstützende Dokumentationen zur Verfügung zu stellen.

Gesamtumfang 316 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE

Nationales Vorwort

Die Englische Fassung des europäischen Dokuments FprEN 61970-301:2013 „Schnittstelle für Anwendungsprogramme für Netzführungssysteme (EMS-API) – Teil 301: Allgemeines Informationsmodell (CIM), Basismodell“, (Schluss-Entwurf) ist unverändert in diesen Norm-Entwurf übernommen worden.

Die Internationale Elektrotechnische Kommission (IEC) und das Europäische Komitee für Elektrotechnische Normung (CENELEC) haben vereinbart, dass ein bei IEC erarbeiteter Entwurf für eine Internationale Norm zeitgleich (parallel) bei IEC und CENELEC zur Abstimmung als FDIS (en: Final Draft International Standard) bzw. Schluss-Entwurf für eine Europäische Norm gestellt wird, um eine Beschleunigung und Straffung der Normungsarbeit zu erreichen. Dem entsprechend ist das internationale Dokument IEC 57/1331/FDIS:2013 „Energy Management System Application Program Interface (EMS-API) – Part 301: Common Information Model (CIM) Base“ unverändert in den Schluss-Entwurf FprEN 61970-301:2013 übernommen worden.

Das internationale Dokument wurde vom TC 57 „Power systems management and associated information exchange“ der Internationalen Elektrotechnischen Kommission (IEC) erarbeitet.

Dokumente, die bei CENELEC als Europäische Norm angenommen und ratifiziert werden, sind unverändert als Deutsche Normen zu übernehmen.

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

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 den vorliegenden Normentwurf erfüllt.

Da sich die Benutzer des vorliegenden Normentwurfs der englischen Sprache als Fachsprache bedienen, wird die Englische Fassung der EN 61970-301 veröffentlicht. Zu deren Abschnitt 3, der die Begriffe festlegt, wurde eine Übersetzung angefertigt und als informativer Nationaler Anhang NA des vorliegenden Normentwurfs hinzugefügt. Für viele der verwendeten Begriffe existieren keine gebräuchlichen deutschen Benennungen, da sich die deutschen Anwender in der Regel ebenfalls der englischen Benennungen bedienen. Dieser Normentwurf steht nicht in unmittelbarem Zusammenhang mit Rechtsvorschriften und ist nicht als Sicherheitsnorm anzusehen.

Es ist vorgesehen, auch bei der entsprechenden zukünftigen Deutschen Norm auf die deutsche Sprachfassung zu verzichten und diese in der englischsprachigen Fassung zu veröffentlichen.

Änderungen

Gegenüber DIN EN 61970-301:2012-03 wurden folgende Änderungen vorgenommen:

- a) es wurden Erweiterungen hinzugefügt, um den UCTE-Datenaustausch zu unterstützen,
- b) Verschiebung mehrerer Klassen von IEC 61970 zum "Assets Package" in IEC 61968,
- c) Modifikation und Erweiterung des Transformatorsteuerungs- Modells, um von ENTSO-E benötigte Modelle für Phasenverschiebetransformator zu unterstützen,

- d) Ergänzung von fehlenden Reihen-Impedanzen mit Null- und Negativwerten,
- e) neues Paket für Zustandsgrößen hinzugefügt, um den Austausch von Netzwerkmodell-Lösungen für Leistungsfluss, Schätzung von Zustandsgrößen etc. zu unterstützen,
- f) Folgende Klassen wurden hinzugefügt: PhaseTapChanger, RatioTapChanger, ImpedanceVariationCurve, RatioVariationCurve, TapSchedule, SwitchSchedule, PhaseVariationCurve, EquivalentInjection hinzugefügt zum Paket "Equivalents", WindGeneratingUnit und NuclearGeneratingUnit ergänzt als Untergruppe von GeneratingUnit,
- g) Folgende Klassen wurden entfernt: Company, Load, CustomerLoad, InductionMotorLoad, Subtypes of ConformLoad, NonConFormLoad, HeatExchanger, Datatypes ShortLength and LongLength wurde entfernt und ersetzt durch Length, MeasurementType Klasse wurde entfernt und ersetzt durch das Attribut Measurement.measurementType,
- h) Verschiedene redaktionelle Änderungen zur Bereinigung des UML-Modells.

Nationaler Anhang NA (informativ)

3 Begriffe

3.1 Begriffsübersicht

Für die Anwendung dieses Dokuments gelten die Begriffe nach IEC 61970-2 und die folgenden Begriffe.

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

3.2

Netzführungssystem

(en: energy management system, 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

3.3

Anwendungsprogramm-Schnittstelle

(en: application program interface, API)

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

– Leerseite –

CONTENTS

FOREWORD	18
INTRODUCTION	21
1 Scope	23
2 Normative references	23
3 Terms and definitions	24
4 CIM specification	24
4.1 CIM modeling notation	24
4.2 CIM packages	24
4.3 CIM classes and relationships	26
4.3.1 Classes	26
4.3.2 Generalization	27
4.3.3 Simple association	28
4.3.4 Aggregation	29
4.4 CIM model concepts and examples	29
4.4.1 Concepts	29
4.4.2 Containment, equipment hierarchies and naming	29
4.4.3 Connectivity model	32
4.4.4 Inheritance hierarchy	35
4.4.5 Transformer model	37
4.4.6 Regulating transformer modeling	38
4.4.7 Measurements and controls	41
4.4.8 Regulating control models	45
4.5 Modeling guidelines	45
4.5.1 Modeling for change	45
4.5.2 Process for amendments to the CIM	46
4.5.3 Changes to the CIM UML model	46
4.5.4 Changes to the CIM standards documents	46
4.5.5 CIM profiles	46
4.6 Modeling tools	47
4.7 User implementation conventions	47
4.8 CIM modeling examples	47
5 Detailed model	48
5.1 Overview	48
5.2 Context	48
6 Package architecture	49
6.1 Top package IEC61970	49
6.1.1 General	49
6.1.2 IEC61970CIMVersion root class	50
6.1.3 Package Domain	51
6.1.4 Package Core	70
6.1.5 Package OperationalLimits	96
6.1.6 Package Topology	104
6.1.7 Package Wires	111
6.1.8 Package Generation	175
6.1.9 Package LoadModel	239

6.1.10 Package Outage	252
6.1.11 Package Protection	257
6.1.12 Package Equivalents	263
6.1.13 Package Meas	268
6.1.14 Package SCADA	294
6.1.15 Package ControlArea.....	299
6.1.16 Package Contingency	305
6.1.17 Package StateVariables.....	308
Bibliography.....	313
Figure 1 – CIM IEC 61970-301 package diagram	26
Figure 2 – Example of generalization.....	28
Figure 3 – Example of simple association	29
Figure 4 – Example of aggregation.....	29
Figure 5 – Equipment containers.....	31
Figure 6 – Connectivity model.....	32
Figure 7 – Simple network example.....	34
Figure 8 – Simple network connectivity modeled with CIM Topology	34
Figure 9 – Equipment inheritance hierarchy.....	36
Figure 10 – Transformer model	38
Figure 11 – Symmetrical Phase Shifter	39
Figure 12 – Asymmetrical Phase Shifter.....	40
Figure 13 – Navigating from PSR to MeasurementValue.....	42
Figure 14 – Measurement placement.....	44
Figure 15 – Regulating control models.....	45
Figure 16 – CIM top level packages.....	48
Figure 17 – Logical diagram IEC61970::Main	50
Figure 18 – Logical diagram Domain::CombinedDatatypes.....	51
Figure 19 – Logical diagram Domain::BasicDatatypes	51
Figure 20 – Logical diagram Domain::ElectricityDatatypes	52
Figure 21 – Logical diagram Domain::EnumeratedUnitDatatypes	53
Figure 22 – Logical diagram Domain::GeneralDatatypes.....	54
Figure 23 – Logical diagram Domain::MonetaryDatatypes.....	55
Figure 24 – Logical diagram Domain::TimeDatatypes	55
Figure 25 – Logical diagram Core::Reporting.....	70
Figure 26 – Logical diagram Core::Main.....	71
Figure 27 – Logical diagram Core::CurveSchedule.....	72
Figure 28 – Logical diagram Core::Datatypes	72
Figure 29 – Logical diagram Core::DocumentationExampleAggregation	73
Figure 30 – Logical diagram Core::DocumentationExampleAssociation	73
Figure 31 – Logical diagram Core::Ownership	74
Figure 32 – Logical diagram OperationalLimits::OperationalLimits	97
Figure 33 – Logical diagram OperationalLimits::BranchGroup	98
Figure 34 – Logical diagram Topology::TopologicalNodeTerminal.....	105

Figure 35 – Logical diagram Topology::Topology	106
Figure 36 – Logical diagram Topology::TopologyMeasRelations	107
Figure 37 – Logical diagram Topology::TopologyReporting	108
Figure 38 – Logical diagram Topology::Main	109
Figure 39 – Logical diagram Wires::DocumentationExampleInheritance	112
Figure 40 – Logical diagram Wires::MutualCoupling	113
Figure 41 – Logical diagram Wires::Schedules	114
Figure 42 – Logical diagram Wires::Datatypes	115
Figure 43 – Logical diagram Wires::InheritanceHierarchy	116
Figure 44 – Logical diagram Wires::LineModel	117
Figure 45 – Logical diagram Wires::NamingHierarchyPart1	118
Figure 46 – Logical diagram Wires::NamingHierarchyPart2	119
Figure 47 – Logical diagram Wires::RegulatingEquipment	120
Figure 48 – Logical diagram Wires::TransformerModel	121
Figure 49 – Logical diagram Wires::VoltageControl	122
Figure 50 – Logical diagram Generation::Main	176
Figure 51 – Logical diagram GenerationDynamics::Main	176
Figure 52 – Logical diagram GenerationDynamics::Datatypes	177
Figure 53 – Logical diagram Production::Nuclear	193
Figure 54 – Logical diagram Production::Main	194
Figure 55 – Logical diagram Production::Datatypes	195
Figure 56 – Logical diagram Production::Hydro	196
Figure 57 – Logical diagram Production::Thermal	197
Figure 58 – Logical diagram LoadModel::Main	239
Figure 59 – Logical diagram LoadModel::Datatypes	240
Figure 60 – Logical diagram Outage::Datatypes	252
Figure 61 – Logical diagram Outage::Main	253
Figure 62 – Logical diagram Protection::Main	257
Figure 63 – Logical diagram Equivalents::Main	263
Figure 64 – Logical diagram Meas::Datatypes	269
Figure 65 – Logical diagram Meas::Meas	270
Figure 66 – Logical diagram Meas::Control	271
Figure 67 – Logical diagram Meas::InheritanceStructure	272
Figure 68 – Logical diagram Meas::Measurement with limits	273
Figure 69 – Logical diagram Meas::Quality	274
Figure 70 – Logical diagram SCADA::Datatypes	294
Figure 71 – Logical diagram SCADA::Main	295
Figure 72 – Logical diagram ControlArea::ControlArea	300
Figure 73 – Logical diagram ControlArea::ControlAreaInheritance	301
Figure 74 – Logical diagram ControlArea::Datatypes	301
Figure 75 – Logical diagram Contingency::Contingency	305
Figure 76 – Logical diagram StateVariables::StateVariables	308

Table 1 – MeasurementType naming conventions	43
Table 2 – MeasurementValueSource naming conventions.....	44
Table 3 – Attributes	48
Table 4 – Association ends.....	49
Table 5 – Enums	49
Table 6 – Attributes of IEC61970::IEC61970CIMVersion.....	50
Table 7 – Attributes of Domain::AbsoluteDate.....	55
Table 8 – Attributes of Domain::AbsoluteDateTime	56
Table 9 – Attributes of Domain::ActivePower	56
Table 10 – Attributes of Domain::ActivePowerChangeRate	56
Table 11 – Attributes of Domain::Admittance	57
Table 12 – Attributes of Domain::AngleDegrees.....	57
Table 13 – Attributes of Domain::AngleRadians	57
Table 14 – Attributes of Domain::ApparentPower	57
Table 15 – Attributes of Domain::Capacitance	58
Table 16 – Attributes of Domain::Conductance	58
Table 17 – Attributes of Domain::CostPerEnergyUnit.....	58
Table 18 – Attributes of Domain::CostRate.....	59
Table 19 – Literals of Domain::Currency	59
Table 20 – Attributes of Domain::CurrentFlow.....	59
Table 21 – Attributes of Domain::Damping	60
Table 22 – Attributes of Domain::FloatQuantity.....	60
Table 23 – Attributes of Domain::Frequency	60
Table 24 – Attributes of Domain::Hours.....	61
Table 25 – Attributes of Domain::Impedance	61
Table 26 – Attributes of Domain::Inductance	61
Table 27 – Attributes of Domain::IntegerQuantity	61
Table 28 – Attributes of Domain::KWActivePower	62
Table 29 – Attributes of Domain::Length	62
Table 30 – Attributes of Domain::Minutes.....	62
Table 31 – Literals of Domain::MonetaryAmountPerEnergyUnit	63
Table 32 – Literals of Domain::MonetaryAmountPerHeatUnit	63
Table 33 – Literals of Domain::MonetaryAmountRate	63
Table 34 – Attributes of Domain::Money.....	63
Table 35 – Attributes of Domain::PerCent	64
Table 36 – Attributes of Domain::Pressure	64
Table 37 – Attributes of Domain::PU.....	64
Table 38 – Attributes of Domain::Reactance.....	64
Table 39 – Attributes of Domain::ReactivePower	65
Table 40 – Attributes of Domain::RealEnergy	65
Table 41 – Attributes of Domain::Resistance	65
Table 42 – Attributes of Domain::RotationSpeed.....	66
Table 43 – Attributes of Domain::Seconds.....	66

Table 44 – Attributes of Domain::StringQuantity.....	66
Table 45 – Attributes of Domain::Susceptance.....	67
Table 46 – Attributes of Domain::Temperature.....	67
Table 47 – Literals of Domain::UnitMultiplier	67
Table 48 – Literals of Domain::UnitSymbol.....	68
Table 49 – Attributes of Domain::Voltage	69
Table 50 – Attributes of Domain::VoltagePerReactivePower.....	69
Table 51 – Attributes of Domain::Volume	69
Table 52 – Attributes of Domain::WaterLevel.....	69
Table 53 – Attributes of Domain::Weight	70
Table 54 – Attributes of Core::ReportingGroup	74
Table 55 – Association ends of Core::ReportingGroup with other classes.....	75
Table 56 – Attributes of Core::ReportingSuperGroup.....	75
Table 57 – Association ends of Core::ReportingSuperGroup with other classes	75
Table 58 – Attributes of Core::BasePower.....	75
Table 59 – Attributes of Core::BaseVoltage.....	76
Table 60 – Association ends of Core::BaseVoltage with other classes.....	76
Table 61 – Attributes of Core::BasicIntervalSchedule	77
Table 62 – Attributes of Core::Bay.....	77
Table 63 – Association ends of Core::Bay with other classes	78
Table 64 – Literals of Core::BreakerConfiguration.....	78
Table 65 – Literals of Core::BusbarConfiguration.....	78
Table 66 – Literals of Core::CompanyType.....	79
Table 67 – Attributes of Core::ConductingEquipment.....	79
Table 68 – Association ends of Core::ConductingEquipment with other classes	79
Table 69 – Attributes of Core::ConnectivityNode.....	80
Table 70 – Association ends of Core::ConnectivityNode with other classes	80
Table 71 – Attributes of Core::ConnectivityNodeContainer.....	81
Table 72 – Association ends of Core::ConnectivityNodeContainer with other classes.....	81
Table 73 – Attributes of Core::Curve.....	81
Table 74 – Association ends of Core::Curve with other classes	82
Table 75 – Attributes of Core::CurveData.....	82
Table 76 – Association ends of Core::CurveData with other classes	82
Table 77 – Literals of Core::CurveStyle.....	83
Table 78 – Attributes of Core::Equipment.....	83
Table 79 – Association ends of Core::Equipment with other classes	83
Table 80 – Attributes of Core::EquipmentContainer.....	84
Table 81 – Association ends of Core::EquipmentContainer with other classes	84
Table 82 – Attributes of Core::GeographicalRegion.....	85
Table 83 – Association ends of Core::GeographicalRegion with other classes	85
Table 84 – Attributes of Core::IdentifiedObject	85
Table 85 – Attributes of Core::IrregularIntervalSchedule	86
Table 86 – Association ends of Core::IrregularIntervalSchedule with other classes	86

Table 87 – Attributes of Core::IrregularTimePoint	87
Table 88 – Association ends of Core::IrregularTimePoint with other classes	87
Table 89 – Attributes of Core::OperatingParticipant.....	87
Table 90 – Association ends of Core::OperatingParticipant with other classes	87
Table 91 – Attributes of Core::OperatingShare	88
Table 92 – Association ends of Core::OperatingShare with other classes	88
Table 93 – Literals of Core::PhaseCode.....	88
Table 94 – Attributes of Core::PowerSystemResource.....	89
Table 95 – Association ends of Core::PowerSystemResource with other classes	89
Table 96 – Attributes of Core::PsrList	90
Table 97 – Association ends of Core::PsrList with other classes.....	90
Table 98 – Attributes of Core::PSRType.....	90
Table 99 – Association ends of Core::PSRType with other classes	91
Table 100 – Attributes of Core::RegularIntervalSchedule	91
Table 101 – Association ends of Core::RegularIntervalSchedule with other classes.....	91
Table 102 – Attributes of Core::RegularTimePoint.....	92
Table 103 – Association ends of Core::RegularTimePoint with other classes	92
Table 104 – Attributes of Core::SubGeographicalRegion	92
Table 105 – Association ends of Core::SubGeographicalRegion with other classes	92
Table 106 – Attributes of Core::Substation	93
Table 107 – Association ends of Core::Substation with other classes.....	93
Table 108 – Attributes of Core::Terminal.....	94
Table 109 – Association ends of Core::Terminal with other classes	94
Table 110 – Attributes of Core::Unit.....	95
Table 111 – Association ends of Core::Unit with other classes	95
Table 112 – Attributes of Core::VoltageLevel.....	96
Table 113 – Association ends of Core::VoltageLevel with other classes	96
Table 114 – Literals of OperationalLimits::OperationalLimitDirectionKind	98
Table 115 – Attributes of OperationalLimits::OperationalLimitType	99
Table 116 – Association ends of OperationalLimits::OperationalLimitType with other classes.....	99
Table 117 – Attributes of OperationalLimits::ActivePowerLimit	99
Table 118 – Association ends of OperationalLimits::ActivePowerLimit with other classes.....	100
Table 119 – Attributes of OperationalLimits::ApparentPowerLimit	100
Table 120 – Association ends of OperationalLimits::ApparentPowerLimit with other classes.....	100
Table 121 – Attributes of OperationalLimits::BranchGroup	101
Table 122 – Association ends of OperationalLimits::BranchGroup with other classes.....	101
Table 123 – Attributes of OperationalLimits::BranchGroupTerminal.....	101
Table 124 – Association ends of OperationalLimits::BranchGroupTerminal with other classes.....	101
Table 125 – Attributes of OperationalLimits::CurrentLimit	102
Table 126 – Association ends of OperationalLimits::CurrentLimit with other classes	102

Table 127 – Attributes of OperationalLimits::OperationalLimit	102
Table 128 – Association ends of OperationalLimits::OperationalLimit with other classes	103
Table 129 – Attributes of OperationalLimits::OperationalLimitSet.....	103
Table 130 – Association ends of OperationalLimits::OperationalLimitSet with other classes.....	103
Table 131 – Attributes of OperationalLimits::VoltageLimit	104
Table 132 – Association ends of OperationalLimits::VoltageLimit with other classes	104
Table 133 – Attributes of Topology::BusNameMarker	110
Table 134 – Association ends of Topology::BusNameMarker with other classes.....	110
Table 135 – Attributes of Topology::TopologicalNode	110
Table 136 – Association ends of Topology::TopologicalNode with other classes.....	111
Table 137 – Attributes of Wires::ImpedanceVariationCurve	122
Table 138 – Association ends of Wires::ImpedanceVariationCurve with other classes.....	123
Table 139 – Attributes of Wires::PhaseVariationCurve.....	123
Table 140 – Association ends of Wires::PhaseVariationCurve with other classes	124
Table 141 – Attributes of Wires::RatioVariationCurve	124
Table 142 – Association ends of Wires::RatioVariationCurve with other classes.....	124
Table 143 – Literals of Wires::RegulatingControlModeKind	125
Table 144 – Attributes of Wires::SwitchSchedule	125
Table 145 – Association ends of Wires::SwitchSchedule with other classes.....	126
Table 146 – Attributes of Wires::ACLineSegment.....	126
Table 147 – Association ends of Wires::ACLineSegment with other classes	127
Table 148 – Attributes of Wires::Breaker.....	127
Table 149 – Association ends of Wires::Breaker with other classes	128
Table 150 – Attributes of Wires::BusbarSection	128
Table 151 – Association ends of Wires::BusbarSection with other classes.....	129
Table 152 – Attributes of Wires::CompositeSwitch	130
Table 153 – Association ends of Wires::CompositeSwitch with other classes.....	130
Table 154 – Attributes of Wires::CompositeSwitchType	130
Table 155 – Attributes of Wires::Conductor	131
Table 156 – Association ends of Wires::Conductor with other classes.....	131
Table 157 – Attributes of Wires::Connector	132
Table 158 – Association ends of Wires::Connector with other classes.....	132
Table 159 – Literals of Wires::CoolantType	133
Table 160 – Attributes of Wires::DCLineSegment.....	133
Table 161 – Association ends of Wires::DCLineSegment with other classes	133
Table 162 – Attributes of Wires::Disconnecter	134
Table 163 – Association ends of Wires::Disconnecter with other classes.....	134
Table 164 – Attributes of Wires::EnergyConsumer	135
Table 165 – Association ends of Wires::EnergyConsumer with other classes.....	136
Table 166 – Attributes of Wires::EnergySource.....	136
Table 167 – Association ends of Wires::EnergySource with other classes	137
Table 168 – Attributes of Wires::FrequencyConverter.....	137

Table 169 – Association ends of Wires::FrequencyConverter with other classes	138
Table 170 – Attributes of Wires::Fuse	139
Table 171 – Association ends of Wires::Fuse with other classes.....	139
Table 172 – Attributes of Wires::Ground	140
Table 173 – Association ends of Wires::Ground with other classes	140
Table 174 – Attributes of Wires::GroundDisconnector.....	141
Table 175 – Association ends of Wires::GroundDisconnector with other classes	141
Table 176 – Attributes of Wires::HeatExchanger	142
Table 177 – Association ends of Wires::HeatExchanger with other classes.....	142
Table 178 – Attributes of Wires::Jumper	142
Table 179 – Association ends of Wires::Jumper with other classes	143
Table 180 – Attributes of Wires::Junction	144
Table 181 – Association ends of Wires::Junction with other classes.....	144
Table 182 – Attributes of Wires::Line	145
Table 183 – Association ends of Wires::Line with other classes.....	145
Table 184 – Attributes of Wires::LoadBreakSwitch	145
Table 185 – Association ends of Wires::LoadBreakSwitch with other classes.....	146
Table 186 – Attributes of Wires::MutualCoupling.....	147
Table 187 – Association ends of Wires::MutualCoupling with other classes	147
Table 188 – Attributes of Wires::OperatingMode	148
Table 189 – Attributes of Wires::PhaseTapChanger	148
Table 190 – Association ends of Wires::PhaseTapChanger with other classes	149
Table 191 – Literals of Wires::PhaseTapChangerKind	150
Table 192 – Attributes of Wires::Plant.....	150
Table 193 – Association ends of Wires::Plant with other classes	150
Table 194 – Attributes of Wires::PowerTransformer	151
Table 195 – Association ends of Wires::PowerTransformer with other classes	151
Table 196 – Attributes of Wires::ProtectedSwitch.....	152
Table 197 – Association ends of Wires::ProtectedSwitch with other classes	152
Table 198 – Attributes of Wires::RatioTapChanger.....	153
Table 199 – Association ends of Wires::RatioTapChanger with other classes	153
Table 200 – Attributes of Wires::ReactiveCapabilityCurve.....	154
Table 201 – Association ends of Wires::ReactiveCapabilityCurve with other classes	155
Table 202 – Attributes of Wires::RectifierInverter	155
Table 203 – Association ends of Wires::RectifierInverter with other classes.....	156
Table 204 – Attributes of Wires::RegulatingCondEq	156
Table 205 – Association ends of Wires::RegulatingCondEq with other classes.....	157
Table 206 – Attributes of Wires::RegulatingControl	157
Table 207 – Association ends of Wires::RegulatingControl with other classes.....	158
Table 208 – Attributes of Wires::RegulationSchedule	158
Table 209 – Association ends of Wires::RegulationSchedule with other classes.....	159
Table 210 – Attributes of Wires::Resistor	159
Table 211 – Association ends of Wires::Resistor with other classes.....	160

Table 212 – Attributes of Wires::SeriesCompensator	160
Table 213 – Association ends of Wires::SeriesCompensator with other classes	161
Table 214 – Attributes of Wires::ShuntCompensator	161
Table 215 – Association ends of Wires::ShuntCompensator with other classes	162
Table 216 – Attributes of Wires::StaticVarCompensator	163
Table 217 – Association ends of Wires::StaticVarCompensator with other classes	164
Table 218 – Literals of Wires::SVCControlMode	164
Table 219 – Attributes of Wires::Switch	165
Table 220 – Association ends of Wires::Switch with other classes	165
Table 221 – Attributes of Wires::SynchronousMachine	166
Table 222 – Association ends of Wires::SynchronousMachine with other classes	167
Table 223 – Literals of Wires::SynchronousMachineOperatingMode	168
Table 224 – Literals of Wires::SynchronousMachineType	168
Table 225 – Attributes of Wires::TapChanger	169
Table 226 – Association ends of Wires::TapChanger with other classes	169
Table 227 – Literals of Wires::TapChangerKind	170
Table 228 – Attributes of Wires::TapSchedule	170
Table 229 – Association ends of Wires::TapSchedule with other classes	171
Table 230 – Literals of Wires::TransformerControlMode	171
Table 231 – Attributes of Wires::TransformerWinding	171
Table 232 – Association ends of Wires::TransformerWinding with other classes	172
Table 233 – Attributes of Wires::VoltageControlZone	173
Table 234 – Association ends of Wires::VoltageControlZone with other classes	173
Table 235 – Literals of Wires::WindingConnection	174
Table 236 – Attributes of Wires::WindingTest	174
Table 237 – Association ends of Wires::WindingTest with other classes	175
Table 238 – Literals of Wires::WindingType	175
Table 239 – Literals of GenerationDynamics::BoilerControlMode	177
Table 240 – Attributes of GenerationDynamics::BWRSteamSupply	177
Table 241 – Association ends of GenerationDynamics::BWRSteamSupply with other classes	178
Table 242 – Attributes of GenerationDynamics::CombustionTurbine	179
Table 243 – Association ends of GenerationDynamics::CombustionTurbine with other classes	179
Table 244 – Attributes of GenerationDynamics::CTTempActivePowerCurve	180
Table 245 – Association ends of GenerationDynamics::CTTempActivePowerCurve with other classes	180
Table 246 – Attributes of GenerationDynamics::DrumBoiler	181
Table 247 – Association ends of GenerationDynamics::DrumBoiler with other classes	182
Table 248 – Attributes of GenerationDynamics::FossilSteamSupply	182
Table 249 – Association ends of GenerationDynamics::FossilSteamSupply with other classes	183
Table 250 – Attributes of GenerationDynamics::HeatRecoveryBoiler	183
Table 251 – Association ends of GenerationDynamics::HeatRecoveryBoiler with other classes	184

Table 252 – Attributes of GenerationDynamics::HydroTurbine	185
Table 253 – Association ends of GenerationDynamics::HydroTurbine with other classes	185
Table 254 – Attributes of GenerationDynamics::PrimeMover	186
Table 255 – Association ends of GenerationDynamics::PrimeMover with other classes	186
Table 256 – Attributes of GenerationDynamics::PWRSteamSupply	186
Table 257 – Association ends of GenerationDynamics::PWRSteamSupply with other classes	187
Table 258 – Attributes of GenerationDynamics::SteamSupply	188
Table 259 – Association ends of GenerationDynamics::SteamSupply with other classes	188
Table 260 – Attributes of GenerationDynamics::SteamTurbine	188
Table 261 – Association ends of GenerationDynamics::SteamTurbine with other classes	189
Table 262 – Attributes of GenerationDynamics::Subcritical	189
Table 263 – Association ends of GenerationDynamics::Subcritical with other classes	190
Table 264 – Attributes of GenerationDynamics::Supercritical	191
Table 265 – Association ends of GenerationDynamics::Supercritical with other classes	192
Table 266 – Literals of GenerationDynamics::TurbineType	192
Table 267 – Literals of Production::HydroEnergyConversionKind	197
Table 268 – Attributes of Production::NuclearGeneratingUnit	198
Table 269 – Association ends of Production::NuclearGeneratingUnit with other classes	199
Table 270 – Attributes of Production::WindGeneratingUnit	200
Table 271 – Association ends of Production::WindGeneratingUnit with other classes	201
Table 272 – Attributes of Production::AirCompressor	202
Table 273 – Association ends of Production::AirCompressor with other classes	202
Table 274 – Attributes of Production::CAESPlant	202
Table 275 – Association ends of Production::CAESPlant with other classes	203
Table 276 – Attributes of Production::Classification	203
Table 277 – Attributes of Production::CogenerationPlant	203
Table 278 – Association ends of Production::CogenerationPlant with other classes	204
Table 279 – Attributes of Production::CombinedCyclePlant	204
Table 280 – Association ends of Production::CombinedCyclePlant with other classes	205
Table 281 – Attributes of Production::CostPerHeatUnit	205
Table 282 – Attributes of Production::Emission	205
Table 283 – Attributes of Production::EmissionAccount	206
Table 284 – Association ends of Production::EmissionAccount with other classes	206
Table 285 – Attributes of Production::EmissionCurve	206
Table 286 – Association ends of Production::EmissionCurve with other classes	207
Table 287 – Literals of Production::EmissionType	207
Table 288 – Literals of Production::EmissionValueSource	208
Table 289 – Attributes of Production::FossilFuel	208
Table 290 – Association ends of Production::FossilFuel with other classes	209
Table 291 – Attributes of Production::FuelAllocationSchedule	209
Table 292 – Association ends of Production::FuelAllocationSchedule with other classes	210

Table 293 – Literals of Production::FuelType	210
Table 294 – Attributes of Production::GeneratingUnit	210
Table 295 – Association ends of Production::GeneratingUnit with other classes	212
Table 296 – Literals of Production::GeneratorControlMode	213
Table 297 – Literals of Production::GeneratorControlSource	213
Table 298 – Literals of Production::GeneratorOperatingMode	214
Table 299 – Attributes of Production::GenUnitOpCostCurve	214
Table 300 – Association ends of Production::GenUnitOpCostCurve with other classes	215
Table 301 – Attributes of Production::GenUnitOpSchedule	215
Table 302 – Association ends of Production::GenUnitOpSchedule with other classes	215
Table 303 – Attributes of Production::GrossToNetActivePowerCurve	216
Table 304 – Association ends of Production::GrossToNetActivePowerCurve with other classes	216
Table 305 – Attributes of Production::HeatInputCurve	217
Table 306 – Association ends of Production::HeatInputCurve with other classes	217
Table 307 – Attributes of Production::HeatRate	218
Table 308 – Attributes of Production::HeatRateCurve	218
Table 309 – Association ends of Production::HeatRateCurve with other classes	218
Table 310 – Attributes of Production::HydroGeneratingEfficiencyCurve	219
Table 311 – Association ends of Production::HydroGeneratingEfficiencyCurve with other classes	219
Table 312 – Attributes of Production::HydroGeneratingUnit	220
Table 313 – Association ends of Production::HydroGeneratingUnit with other classes	221
Table 314 – Literals of Production::HydroPlantType	222
Table 315 – Attributes of Production::HydroPowerPlant	222
Table 316 – Association ends of Production::HydroPowerPlant with other classes	223
Table 317 – Attributes of Production::HydroPump	223
Table 318 – Association ends of Production::HydroPump with other classes	224
Table 319 – Attributes of Production::HydroPumpOpSchedule	224
Table 320 – Association ends of Production::HydroPumpOpSchedule with other classes	225
Table 321 – Attributes of Production::IncrementalHeatRateCurve	225
Table 322 – Association ends of Production::IncrementalHeatRateCurve with other classes	226
Table 323 – Attributes of Production::InflowForecast	226
Table 324 – Association ends of Production::InflowForecast with other classes	226
Table 325 – Attributes of Production::LevelVsVolumeCurve	227
Table 326 – Association ends of Production::LevelVsVolumeCurve with other classes	227
Table 327 – Attributes of Production::PenstockLossCurve	227
Table 328 – Association ends of Production::PenstockLossCurve with other classes	228
Table 329 – Attributes of Production::Reservoir	228
Table 330 – Association ends of Production::Reservoir with other classes	229
Table 331 – Attributes of Production::ShutdownCurve	230
Table 332 – Association ends of Production::ShutdownCurve with other classes	230

Table 333 – Attributes of Production::StartIgnFuelCurve.....	231
Table 334 – Association ends of Production::StartIgnFuelCurve with other classes	231
Table 335 – Attributes of Production::StartMainFuelCurve	231
Table 336 – Association ends of Production::StartMainFuelCurve with other classes	232
Table 337 – Attributes of Production::StartRampCurve	232
Table 338 – Association ends of Production::StartRampCurve with other classes.....	233
Table 339 – Attributes of Production::StartupModel.....	233
Table 340 – Association ends of Production::StartupModel with other classes	234
Table 341 – Attributes of Production::SteamSendoutSchedule	234
Table 342 – Association ends of Production::SteamSendoutSchedule with other classes.....	234
Table 343 – Attributes of Production::TailbayLossCurve	235
Table 344 – Association ends of Production::TailbayLossCurve with other classes.....	235
Table 345 – Attributes of Production::TargetLevelSchedule	235
Table 346 – Association ends of Production::TargetLevelSchedule with other classes	236
Table 347 – Attributes of Production::ThermalGeneratingUnit	236
Table 348 – Association ends of Production::ThermalGeneratingUnit with other classes	238
Table 349 – Attributes of LoadModel::ConformLoad	240
Table 350 – Association ends of LoadModel::ConformLoad with other classes.....	240
Table 351 – Attributes of LoadModel::ConformLoadGroup	241
Table 352 – Association ends of LoadModel::ConformLoadGroup with other classes.....	241
Table 353 – Attributes of LoadModel::ConformLoadSchedule	242
Table 354 – Association ends of LoadModel::ConformLoadSchedule with other classes	242
Table 355 – Attributes of LoadModel::DayType.....	242
Table 356 – Association ends of LoadModel::DayType with other classes	243
Table 357 – Attributes of LoadModel::EnergyArea.....	243
Table 358 – Association ends of LoadModel::EnergyArea with other classes	243
Table 359 – Attributes of LoadModel::LoadArea.....	243
Table 360 – Association ends of LoadModel::LoadArea with other classes	244
Table 361 – Attributes of LoadModel::LoadGroup	244
Table 362 – Association ends of LoadModel::LoadGroup with other classes	244
Table 363 – Attributes of LoadModel::LoadResponseCharacteristic	245
Table 364 – Association ends of LoadModel::LoadResponseCharacteristic with other classes.....	246
Table 365 – Attributes of LoadModel::NonConformLoad	246
Table 366 – Association ends of LoadModel::NonConformLoad with other classes.....	246
Table 367 – Attributes of LoadModel::NonConformLoadGroup	247
Table 368 – Association ends of LoadModel::NonConformLoadGroup with other classes.....	247
Table 369 – Attributes of LoadModel::NonConformLoadSchedule	248
Table 370 – Association ends of LoadModel::NonConformLoadSchedule with other classes.....	248
Table 371 – Attributes of LoadModel::PowerCutZone	248
Table 372 – Association ends of LoadModel::PowerCutZone with other classes.....	249

Table 373 – Attributes of LoadModel::Season.....	249
Table 374 – Association ends of LoadModel::Season with other classes	249
Table 375 – Attributes of LoadModel::SeasonDayTypeSchedule.....	250
Table 376 – Association ends of LoadModel::SeasonDayTypeSchedule with other classes.....	250
Table 377 – Literals of LoadModel::SeasonName	250
Table 378 – Attributes of LoadModel::StationSupply.....	251
Table 379 – Association ends of LoadModel::StationSupply with other classes	251
Table 380 – Attributes of LoadModel::SubLoadArea	252
Table 381 – Association ends of LoadModel::SubLoadArea with other classes.....	252
Table 382 – Attributes of Outage::ClearanceTag.....	254
Table 383 – Association ends of Outage::ClearanceTag with other classes	254
Table 384 – Attributes of Outage::ClearanceTagType.....	255
Table 385 – Association ends of Outage::ClearanceTagType with other classes	255
Table 386 – Attributes of Outage::OutageSchedule	255
Table 387 – Association ends of Outage::OutageSchedule with other classes.....	256
Table 388 – Attributes of Outage::SwitchingOperation.....	256
Table 389 – Association ends of Outage::SwitchingOperation with other classes	256
Table 390 – Literals of Outage::SwitchState	257
Table 391 – Attributes of Protection::CurrentRelay.....	257
Table 392 – Association ends of Protection::CurrentRelay with other classes	258
Table 393 – Attributes of Protection::FaultIndicator	259
Table 394 – Association ends of Protection::FaultIndicator with other classes.....	259
Table 395 – Attributes of Protection::ProtectionEquipment.....	259
Table 396 – Association ends of Protection::ProtectionEquipment with other classes	260
Table 397 – Attributes of Protection::RecloseSequence.....	260
Table 398 – Association ends of Protection::RecloseSequence with other classes	261
Table 399 – Attributes of Protection::SurgeProtector.....	261
Table 400 – Association ends of Protection::SurgeProtector with other classes.....	261
Table 401 – Attributes of Protection::SynchrocheckRelay	262
Table 402 – Association ends of Protection::SynchrocheckRelay with other classes	262
Table 403 – Attributes of Equivalents::EquivalentInjection	264
Table 404 – Association ends of Equivalents::EquivalentInjection with other classes	264
Table 405 – Attributes of Equivalents::EquivalentBranch	265
Table 406 – Association ends of Equivalents::EquivalentBranch with other classes	265
Table 407 – Attributes of Equivalents::EquivalentEquipment	266
Table 408 – Association ends of Equivalents::EquivalentEquipment with other classes.....	266
Table 409 – Attributes of Equivalents::EquivalentNetwork.....	267
Table 410 – Association ends of Equivalents::EquivalentNetwork with other classes	267
Table 411 – Attributes of Equivalents::EquivalentShunt	267
Table 412 – Association ends of Equivalents::EquivalentShunt with other classes.....	268
Table 413 – Attributes of Meas::Accumulator.....	274
Table 414 – Association ends of Meas::Accumulator with other classes	275

Table 415 – Attributes of Meas::AccumulatorLimit.....	275
Table 416 – Association ends of Meas::AccumulatorLimit with other classes	275
Table 417 – Attributes of Meas::AccumulatorLimitSet	276
Table 418 – Association ends of Meas::AccumulatorLimitSet with other classes.....	276
Table 419 – Attributes of Meas::AccumulatorValue	276
Table 420 – Association ends of Meas::AccumulatorValue with other classes	277
Table 421 – Attributes of Meas::Analog	277
Table 422 – Association ends of Meas::Analog with other classes	277
Table 423 – Attributes of Meas::AnalogLimit.....	278
Table 424 – Association ends of Meas::AnalogLimit with other classes	278
Table 425 – Attributes of Meas::AnalogLimitSet.....	278
Table 426 – Association ends of Meas::AnalogLimitSet with other classes	279
Table 427 – Attributes of Meas::AnalogValue	279
Table 428 – Association ends of Meas::AnalogValue with other classes	279
Table 429 – Attributes of Meas::Command.....	280
Table 430 – Association ends of Meas::Command with other classes.....	280
Table 431 – Attributes of Meas::Control	280
Table 432 – Association ends of Meas::Control with other classes.....	281
Table 433 – Attributes of Meas::ControlType	281
Table 434 – Association ends of Meas::ControlType with other classes.....	281
Table 435 – Attributes of Meas::CurrentTransformer	282
Table 436 – Association ends of Meas::CurrentTransformer with other classes.....	282
Table 437 – Attributes of Meas::Discrete.....	283
Table 438 – Association ends of Meas::Discrete with other classes	283
Table 439 – Attributes of Meas::DiscreteValue	283
Table 440 – Association ends of Meas::DiscreteValue with other classes	284
Table 441 – Attributes of Meas::Limit.....	284
Table 442 – Attributes of Meas::LimitSet.....	285
Table 443 – Attributes of Meas::Measurement.....	285
Table 444 – Association ends of Meas::Measurement with other classes.....	286
Table 445 – Attributes of Meas::MeasurementValue.....	286
Table 446 – Association ends of Meas::MeasurementValue with other classes	286
Table 447 – Attributes of Meas::MeasurementValueQuality.....	287
Table 448 – Association ends of Meas::MeasurementValueQuality with other classes	287
Table 449 – Attributes of Meas::MeasurementValueSource.....	288
Table 450 – Association ends of Meas::MeasurementValueSource with other classes	288
Table 451 – Attributes of Meas::PotentialTransformer	288
Table 452 – Association ends of Meas::PotentialTransformer with other classes.....	289
Table 453 – Attributes of Meas::Quality61850.....	289
Table 454 – Attributes of Meas::SetPoint	290
Table 455 – Association ends of Meas::SetPoint with other classes	290
Table 456 – Attributes of Meas::StringMeasurement.....	291
Table 457 – Association ends of Meas::StringMeasurement with other classes	291

Table 458 – Attributes of Meas::StringMeasurementValue	291
Table 459 – Association ends of Meas::StringMeasurementValue with other classes	292
Table 460 – Literals of Meas::Validity	292
Table 461 – Attributes of Meas::ValueAliasSet	293
Table 462 – Association ends of Meas::ValueAliasSet with other classes	293
Table 463 – Attributes of Meas::ValueToAlias.....	293
Table 464 – Association ends of Meas::ValueToAlias with other classes	294
Table 465 – Attributes of SCADA::CommunicationLink	295
Table 466 – Association ends of SCADA::CommunicationLink with other classes	296
Table 467 – Attributes of SCADA::RemoteControl	296
Table 468 – Association ends of SCADA::RemoteControl with other classes	296
Table 469 – Attributes of SCADA::RemotePoint.....	297
Table 470 – Association ends of SCADA::RemotePoint with other classes.....	297
Table 471 – Attributes of SCADA::RemoteSource	297
Table 472 – Association ends of SCADA::RemoteSource with other classes.....	298
Table 473 – Attributes of SCADA::RemoteUnit	298
Table 474 – Association ends of SCADA::RemoteUnit with other classes	298
Table 475 – Literals of SCADA::RemoteUnitType.....	299
Table 476 – Literals of SCADA::Source	299
Table 477 – Attributes of ControlArea::AltGeneratingUnitMeas.....	302
Table 478 – Association ends of ControlArea::AltGeneratingUnitMeas with other classes.....	302
Table 479 – Attributes of ControlArea::AltTieMeas	302
Table 480 – Association ends of ControlArea::AltTieMeas with other classes.....	302
Table 481 – Attributes of ControlArea::ControlArea.....	303
Table 482 – Association ends of ControlArea::ControlArea with other classes	303
Table 483 – Association ends of ControlArea::ControlAreaGeneratingUnit with other classes.....	304
Table 484 – Literals of ControlArea::ControlAreaTypeKind.....	304
Table 485 – Attributes of ControlArea::TieFlow.....	304
Table 486 – Association ends of ControlArea::TieFlow with other classes	304
Table 487 – Attributes of Contingency::ContingencyEquipment	306
Table 488 – Association ends of Contingency::ContingencyEquipment with other classes.....	306
Table 489 – Literals of Contingency::ContingencyEquipmentStatusKind.....	306
Table 490 – Attributes of Contingency::Contingency.....	307
Table 491 – Association ends of Contingency::Contingency with other classes	307
Table 492 – Attributes of Contingency::ContingencyElement.....	307
Table 493 – Association ends of Contingency::ContingencyElement with other classes	307
Table 494 – Attributes of StateVariables::SvShortCircuit.....	309
Table 495 – Association ends of StateVariables::SvShortCircuit with other classes	309
Table 496 – Attributes of StateVariables::SvInjection	309
Table 497 – Association ends of StateVariables::SvInjection with other classes	309
Table 498 – Attributes of StateVariables::SvPowerFlow.....	

Table 499 – Association ends of StateVariables::SvPowerFlow with other classes	310
Table 500 – Attributes of StateVariables::SvShuntCompensatorSections.....	310
Table 501 – Association ends of StateVariables::SvShuntCompensatorSections with other classes	310
Table 502 – Attributes of StateVariables::SvStatus	311
Table 503 – Association ends of StateVariables::SvStatus with other classes	311
Table 504 – Attributes of StateVariables::SvTapStep	311
Table 505 – Association ends of StateVariables::SvTapStep with other classes	311
Table 506 – Attributes of StateVariables::SvVoltage.....	311
Table 507 – Association ends of StateVariables::SvVoltage with other classes	312
Table 508 – Attributes of StateVariables::TopologicalIsland	312
Table 509 – Association ends of StateVariables::TopologicalIsland with other classes.....	312

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 fourth edition cancels and replaces the third edition, published in 2011. This fourth edition constitutes a technical revision.

Major changes from the third edition include the following:

- extensions have been added to support UCTE exchange;
- several classes have been moved from IEC 61970 to the Assets package in IEC 61968;
- the transformer regulation model has been modified and expanded to support phase shifting transformer models needed by ENTSO-E;
- zero and negative sequence impedance terms have been added where missing;
- new StateVariables package has been added to support exchange of network model solutions from power flow, state estimation, etc.;
- additional classes that have been added included:

- PhaseTapChanger
- RatioTapChanger
- ImpedanceVariationCurve
- RatioVariationCurve
- TapSchedule
- SwitchSchedule
- PhaseVariationCurve
- EquivalentInjection added to the Equivalent package
- WindGeneratingUnit and NuclearGeneratingUnit added as subtypes of GeneratingUnit
- classes that were removed included:
 - Company
 - HeatExchanger
 - MeasurementType class removed and replaced with attribute Measurement.measurementType.
 - Datatypes ShortLength and LongLength were removed and replaced with Length.
 - Load, CustomerLoad, and InductionMotorLoad.
 - Subtypes of ConformLoad and NonConFormLoad
- various editorial changes to cleanup the UML model.

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

FDIS	Report on voting
57/XX/FDIS	57/XX/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.

The National Committees are requested to note that for this publication the stability date is 2013.

THIS TEXT IS INCLUDED FOR THE INFORMATION OF THE NATIONAL COMMITTEES AND WILL BE DELETED AT THE PUBLICATION STAGE.

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 several parts of the IEC 61970 series which define an application program interface (API) for an energy management system (EMS). This part of IEC 61970 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 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.

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

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

ISO 8601, *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 given in IEC 61970-2 and the following apply.

NOTE For general glossary definitions, see the International Electrotechnical Vocabulary, IEC 60050.

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

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 the 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 CIM 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 the IEC.

The CIM package in IEC 61970-301 (this document) is overseen by the same working group that handles the following packages:

- Domain
- Core
- OperationalLimits
- Topology
- Wires
- Generation
- Generation.GenerationDynamics
- Generation.Production
- LoadModel
- Outage
- Protection
- Equivalents
- Meas
- SCADA
- ControlArea
- Contingency
- StateVariables

The 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.

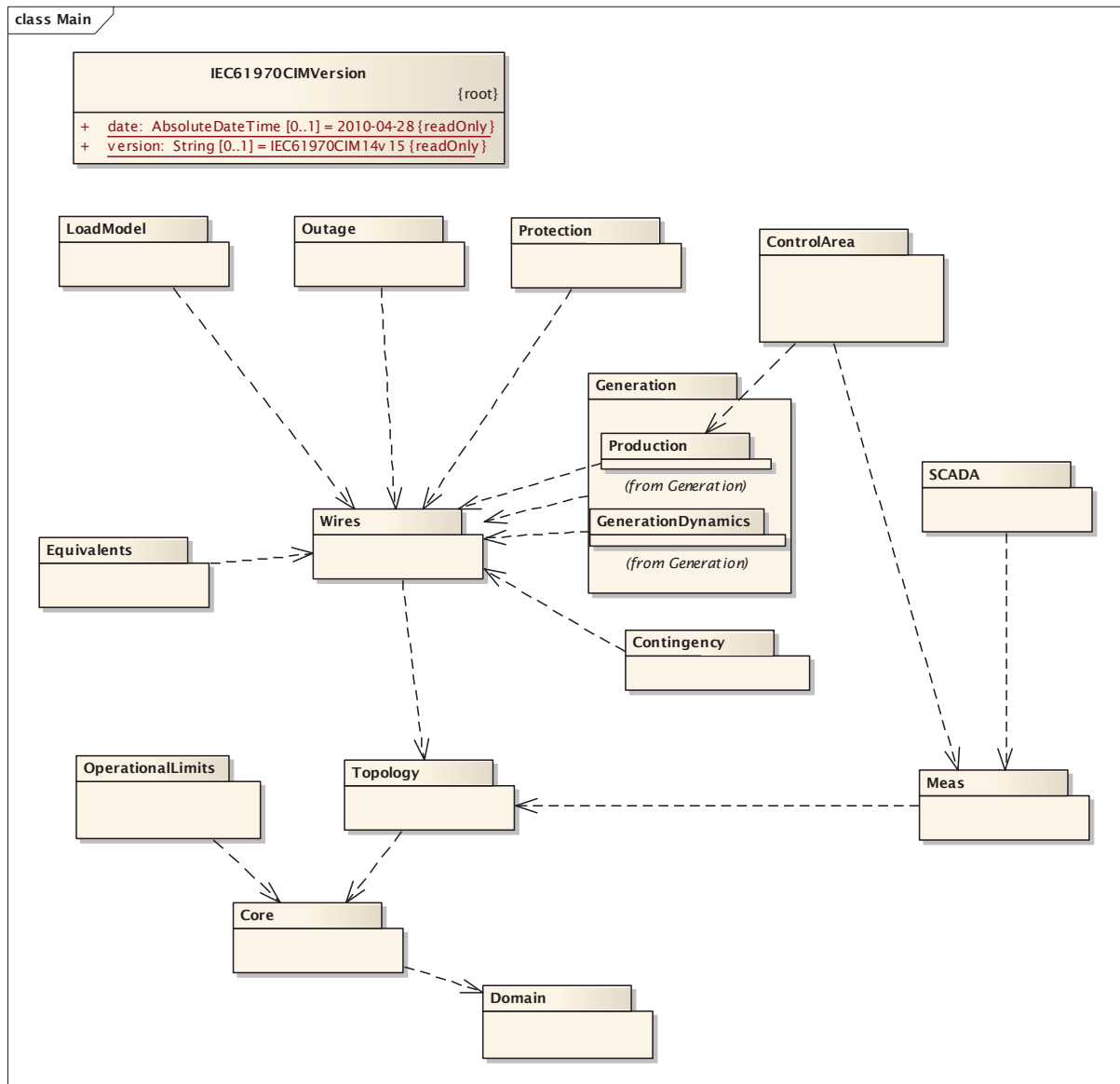


Figure 1 – CIM IEC 61970-301 package diagram

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 IEC61970CIM14v15, 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 are also shown.

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. For a specific interface, a profile is defined consisting of a subset 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, 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 4.2. 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 4.3.2, 4.3.3 and 4.3.4.

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.

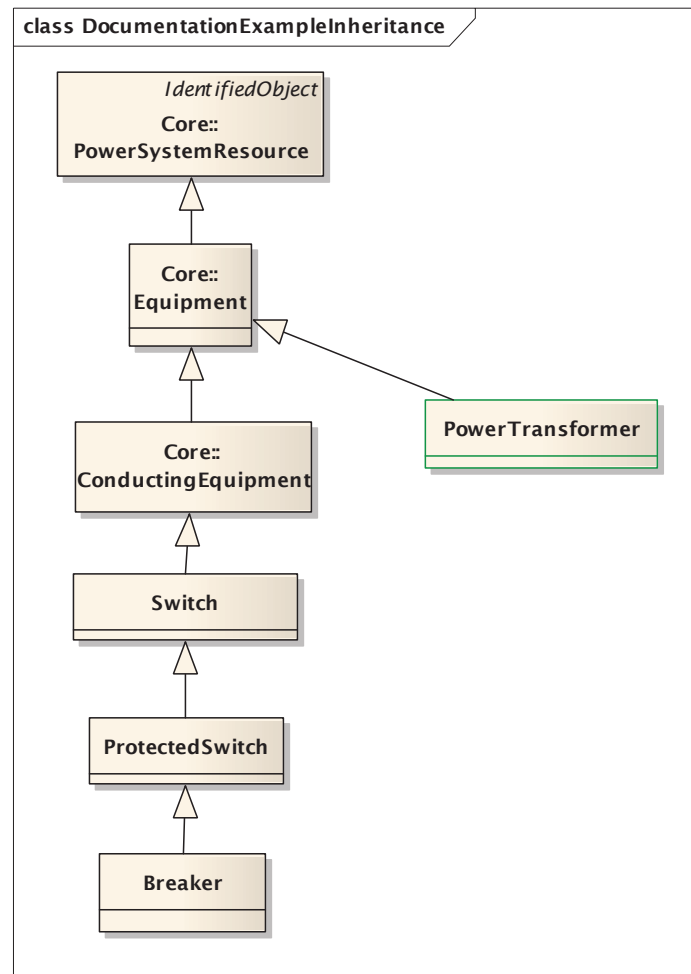


Figure 2 – Example of generalization

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.

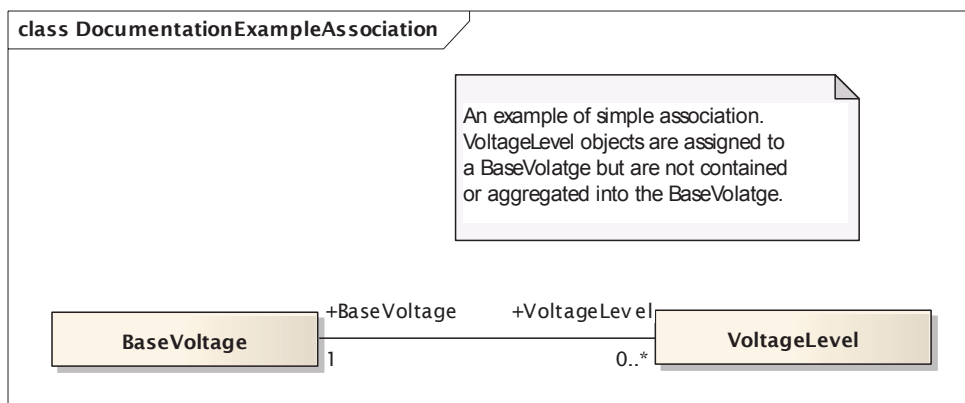


Figure 3 – Example of simple association

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.

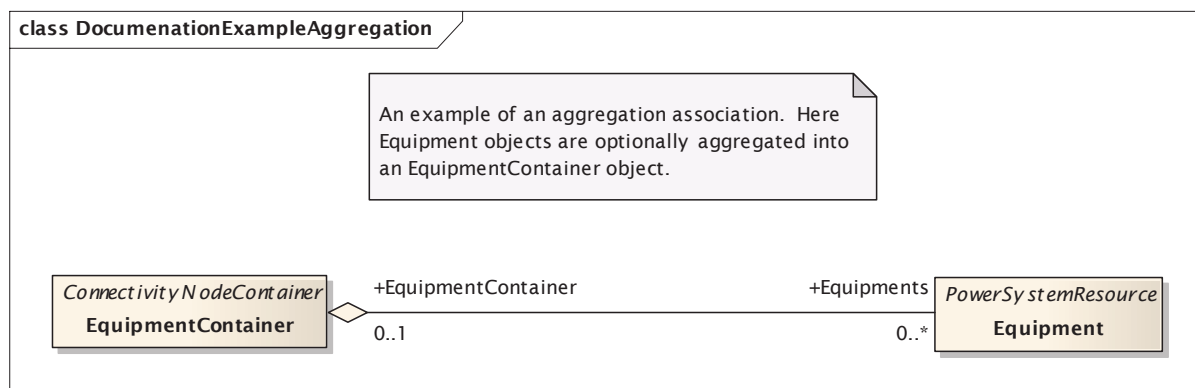


Figure 4 – Example of aggregation

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 4.4.2 to 4.4.8.

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.

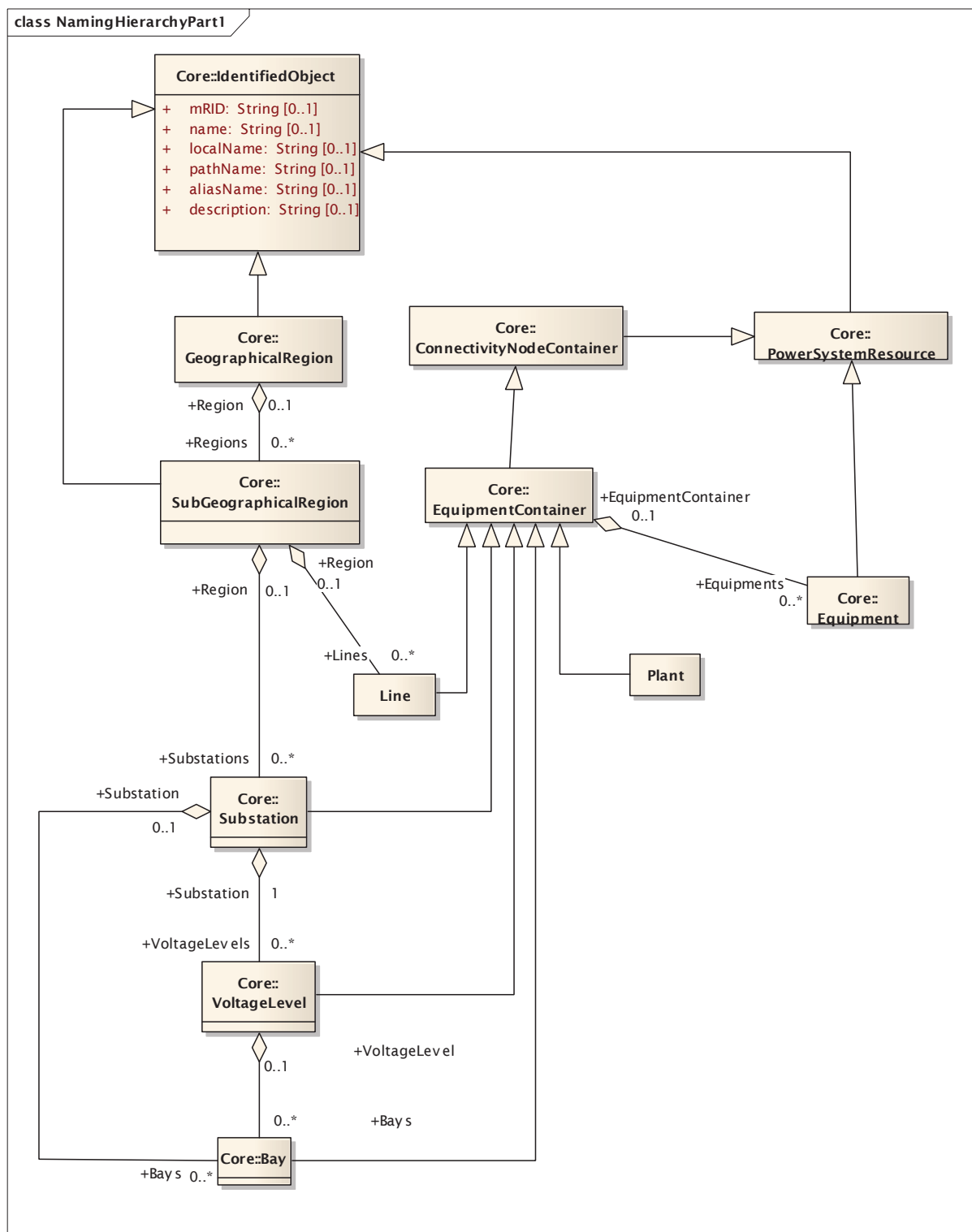


Figure 5 – Equipment containers

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.

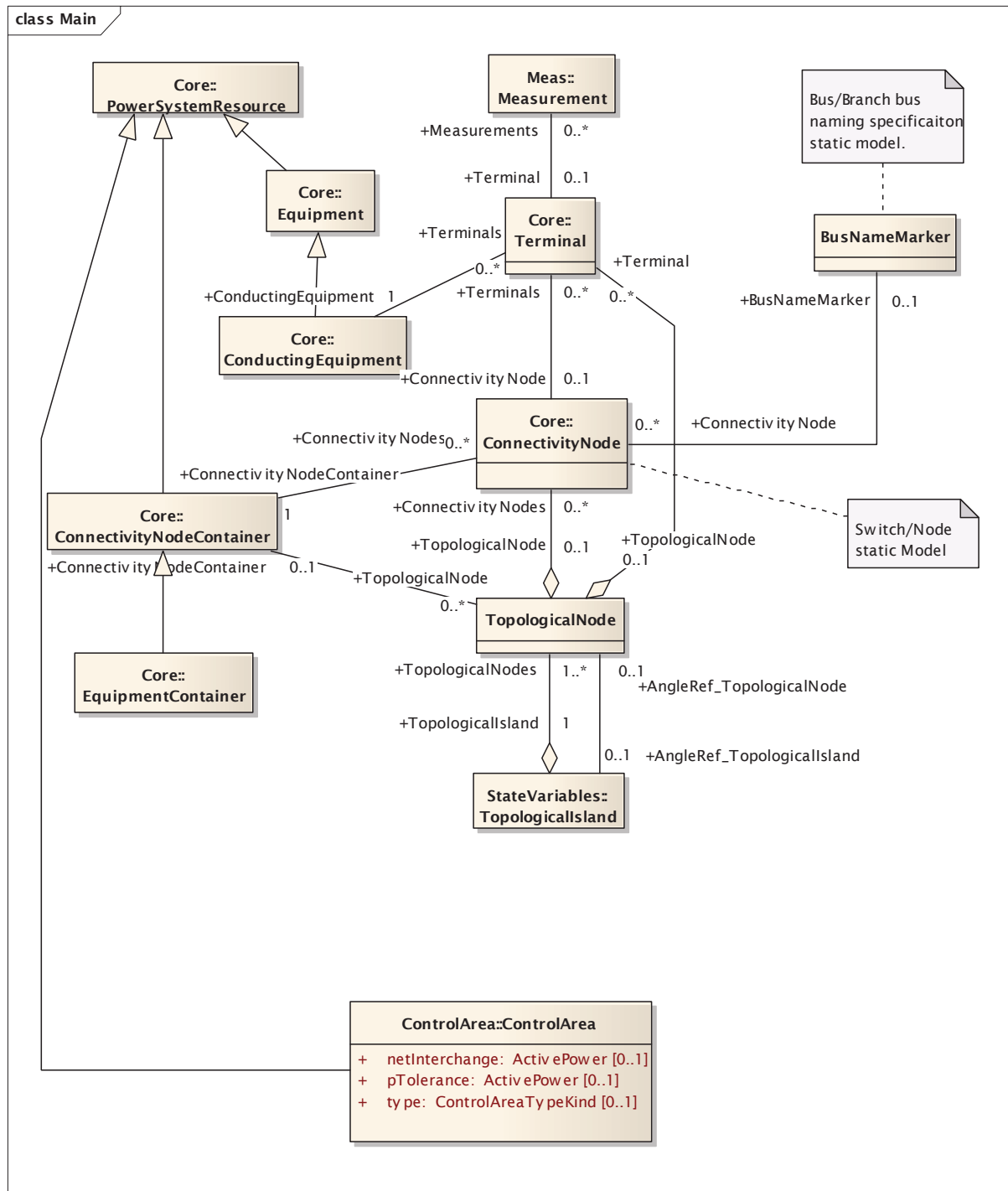


Figure 6 – Connectivity model

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 details.

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.

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.

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.

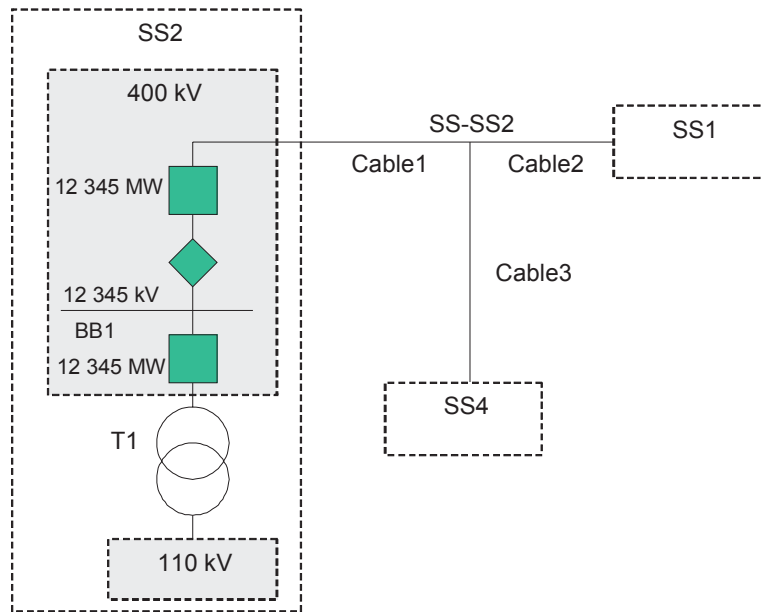


Figure 7 – Simple network example

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.

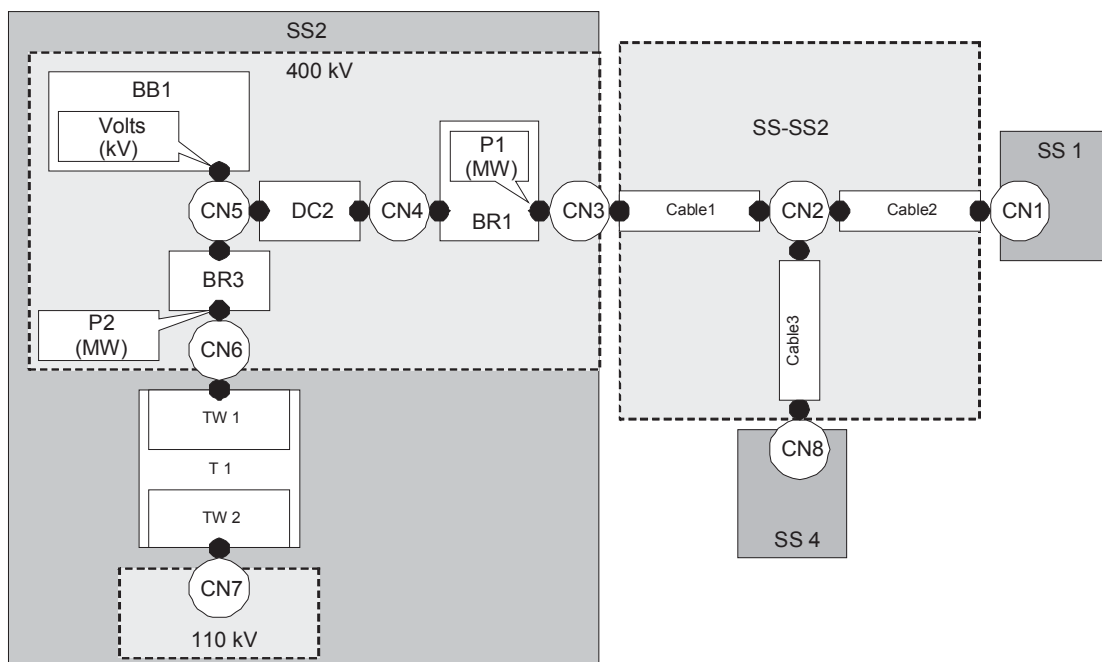


Figure 8 – Simple network connectivity modeled with CIM Topology

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 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.

4.4.4 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.

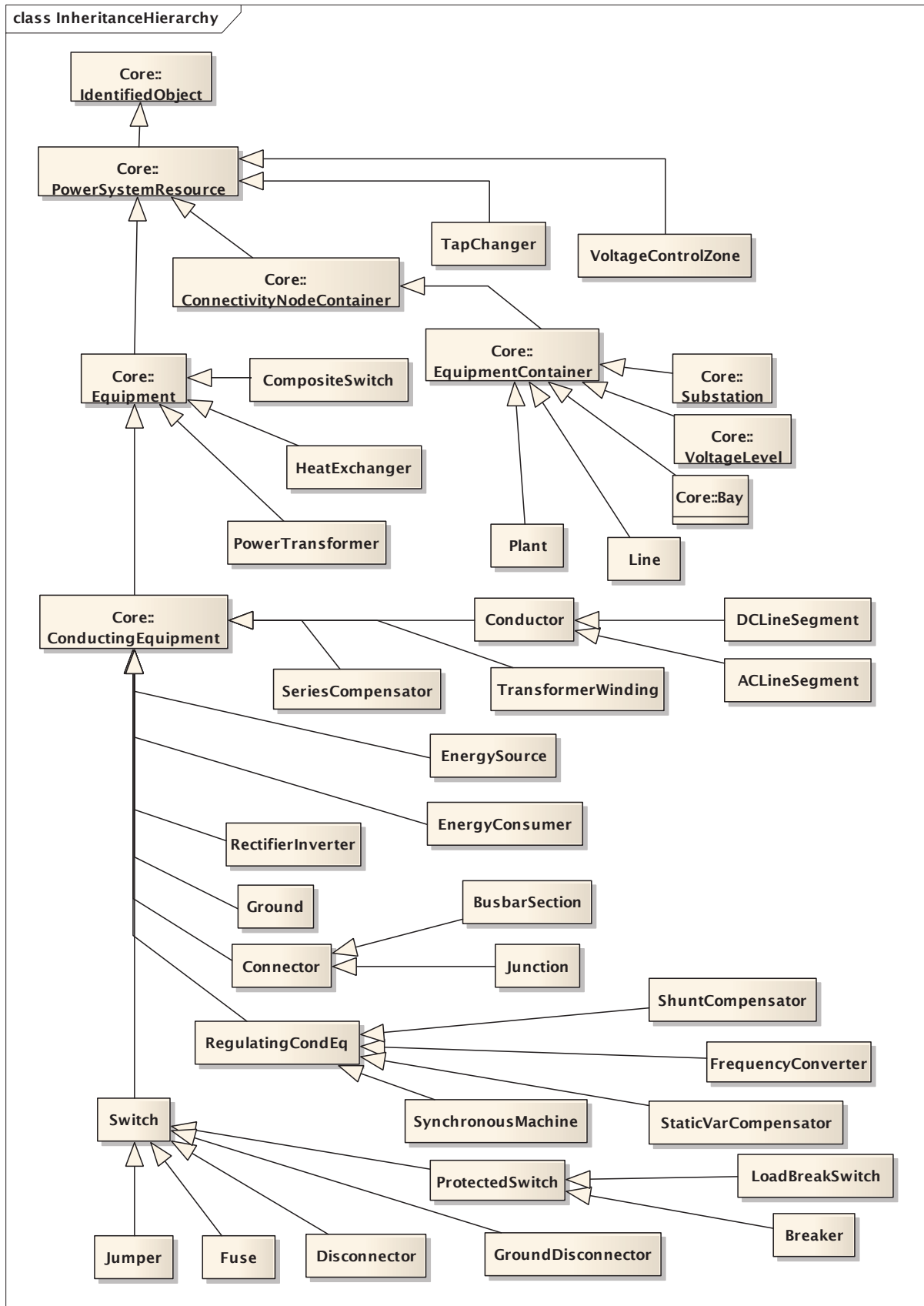


Figure 9 – Equipment inheritance hierarchy

4.4.5 Transformer model

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

As shown in Figure 10, a PowerTransformer is a specialized class of Equipment, which is a specialized class of a PowerSystemResource. 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 the 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 in Figure 10, a PowerTransformer may have (or contain) one or more TransformerWindings, but a TransformerWinding can only belong to (or be a member of) one PowerTransformer.

The TransformerWinding has other relationships as well:

- a generalization relationship with ConductingEquipment;
- two association relationships with the WindingTest class, such that a TransformerWinding object may be From from 0, 1, or more WindingTest objects and To 0, 1, or more WindingTest 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 10 along with the definition of all the attributes and relationships supported in each class.

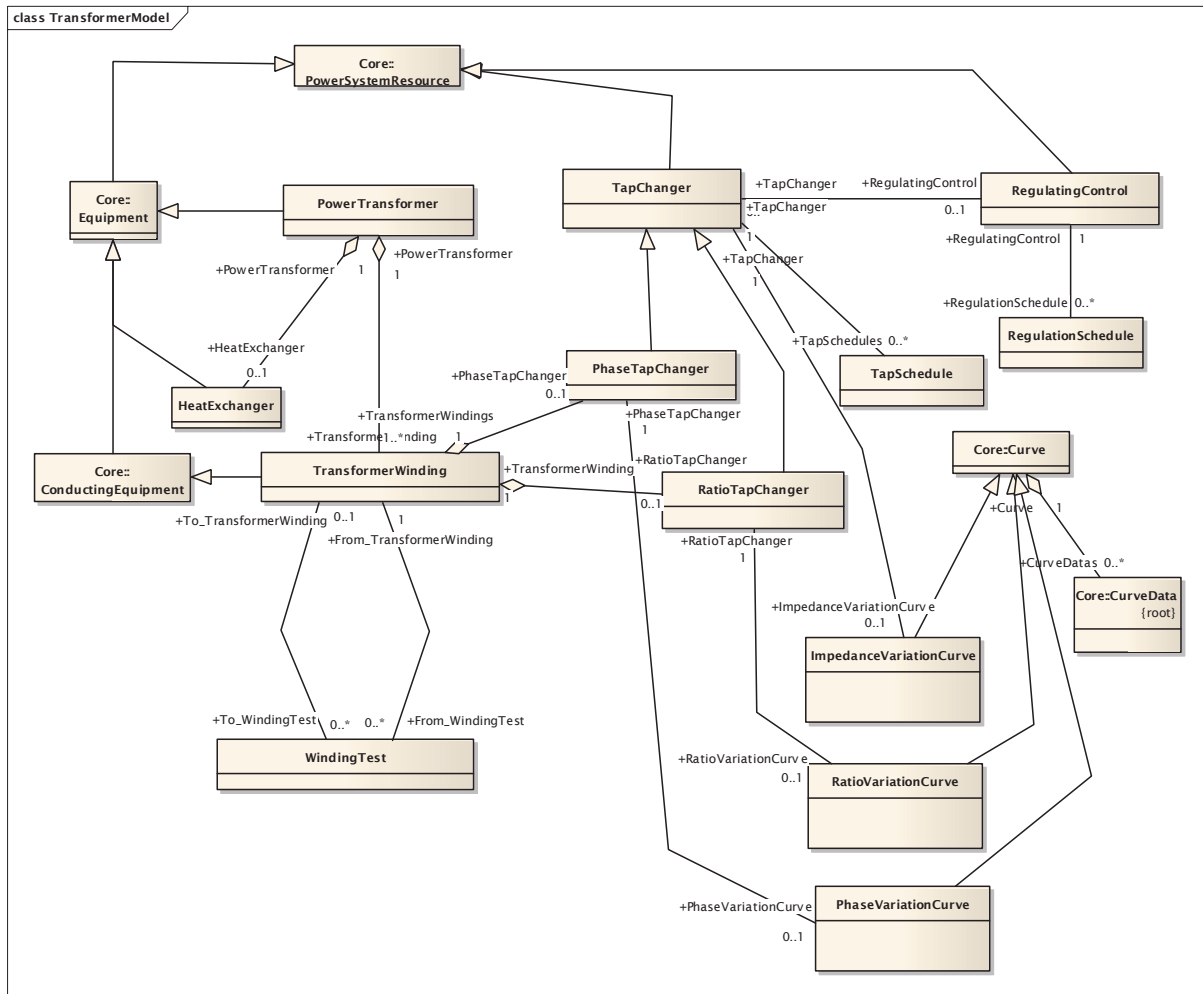


Figure 10 – Transformer model

4.4.6 Regulating transformer modeling

4.4.6.1 Regulating transformer modeling overview

Multiple types of regulating transformers are supported by the CIM model. Depending on the regulation capabilities, the effects of tap movement will be defined using either the RatioTapChanger class or the PhaseTapChanger class. Both of these classes are subtypes of the TapChanger class. For all the RatioTapChangers and PhaseTapChangers, the following attributes are required and have the same meaning.

TapChanger.highStep
 TapChanger.lowStep
 TapChanger.ltcFlag
 TapChanger.neutralStep
 TapChanger.neutralU
 TapChanger.normalStep
 TapChanger.regulationStatus

Other attributes are required and may have slightly different meaning depending on the type of transformer.

4.4.6.2 Voltage ratio transformer modeling

The RatioTapChanger class is used to define a transformer that changes the voltage ratio without impacting the phase angle across the transformer. The additional required attributes for a RatioTapChanger are:

TapChanger.stepVoltageIncrement
RatioTapChanger.tculControlMode

4.4.6.3 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 PhaseTapChanger class. The additional required attributes for a linear phase shifting transformer are:

PhaseTapChanger.stepPhaseShiftIncrement

4.4.6.4 Symmetrical phase shifter modeling

A symmetrical phase shifter construction causes the turns ratio and voltage magnitude transformation to remain constant while the voltage phase is shifted. The changes to voltage magnitude and phase over the full range of tap steps for a symmetrical phase shifter is shown in Figure 11 below.

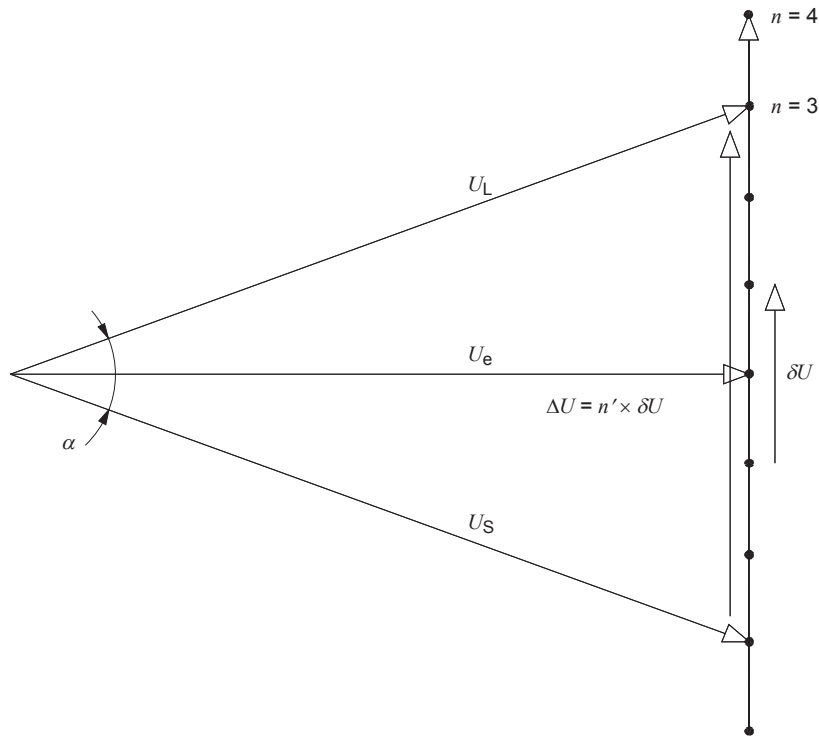


Figure 11 – Symmetrical Phase Shifter

The phase shift angle can be represented as:

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

$$\rho = 1$$

where

α is the phase shift angle;

n' is the current tap position;

ΔU is the voltage change;
 U_e is the voltage at neutral tap;
 δU is the PU voltage change;
 ρ is the transformation ratio.

For a symmetrical phase shifter, the PhaseTapChanger.phaseTapChangerType attribute shall be set to “symmetrical”. The PhaseTapChanger.windingConnectionAngle shall be either $+90^\circ$ or -90° . With $+90^\circ$ there is a positive phase angle increase as the tap position increases. With -90° the phase angle becomes a larger negative angle as the tap position increases. The additional required attributes for a symmetrical phase shifting transformer are:

TapChanger.stepVoltageIncrement
 PhaseTapChanger.phaseTapChangerType
 PhaseTapChanger.stepPhaseShiftIncrement
 PhaseTapChanger.windingConnectionAngle
 PhaseTapChanger.xStepMax
 PhaseTapChanger.xStepMin

4.4.6.5 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 12 below.

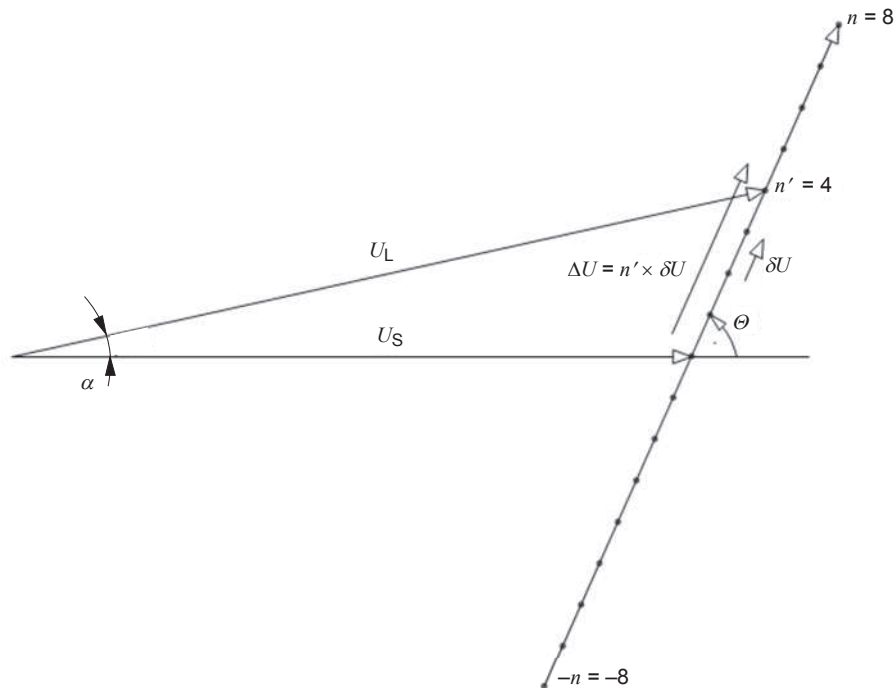


Figure 12 – Asymmetrical Phase Shifter

The phase shift angle can be represented as:

$$\alpha = \arctan\left(\frac{n' \delta U \sin \Theta}{1 + n' \delta U \cos \Theta}\right)$$

$$\rho = \frac{1}{\sqrt{(n' \delta u \sin \Theta)^2 + (1 + n' \delta U \cos \Theta)^2}}$$

where

α is the phase shift angle;

n' is the current tap position;

δU is the PU voltage change;

Θ is the winding connection angle;

ρ is the transformation ratio.

For an asymmetrical phase shifter, the PhaseTapChanger.phaseTapChangerType attribute shall be set to "asymmetrical". The PhaseTapChanger.windingConnectionAngle shall be a multiple of 30°: -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. The additional required attributes for an asymmetrical phase shifting transformer are:

```
TapChanger.stepVoltageIncrement
PhaseTapChanger.phaseTapChangerType
PhaseTapChanger.stepPhaseShiftIncrement
PhaseTapChanger.windingConnectionAngle
PhaseTapChanger.xStepMax
PhaseTapChanger.xStepMin
```

4.4.7 Measurements and controls

4.4.7.1 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.

4.4.7.2 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.

4.4.7.3 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 13):

- Measurements of a PowerSystemResource are classified by the attribute measurementType. The values to be used for Measurement.measurementType are specified in Table 1.
- MeasurementValues of a Measurement are classified by MeasurementValueSource.
- MeasurementValueSource 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 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 (Remote Terminal Unit) 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.

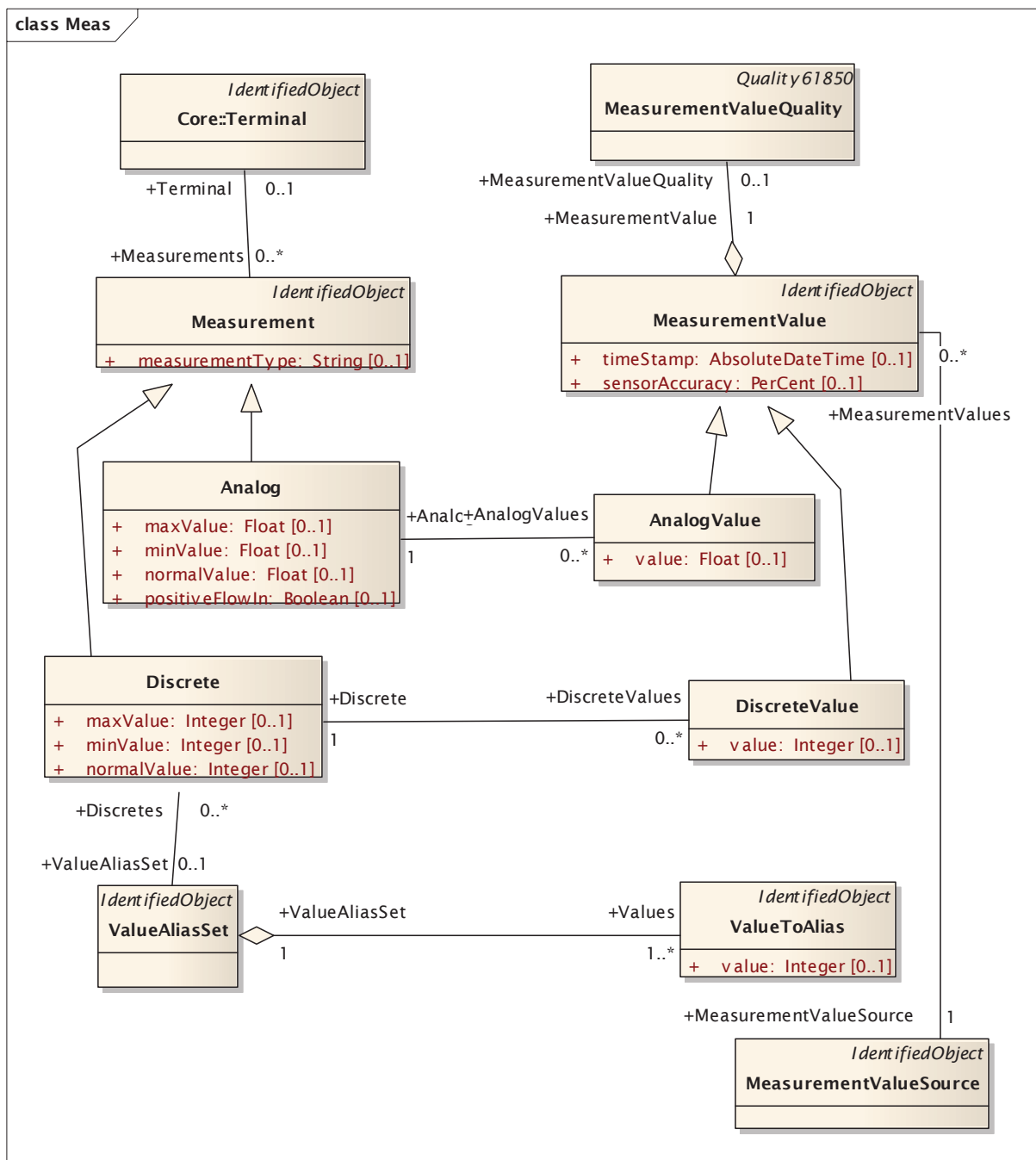


Figure 13 – Navigating from PSR to MeasurementValue

Table 1 – MeasurementType naming conventions

measurementType	IEC 61850 name	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 61850-7-4. The meaning of the columns in Table 1 are as follows:

- measurementType (Measurement.measurementType) is the IEC 61970 measurement type name;
- 61850 Name is the name assigned to the data 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 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:

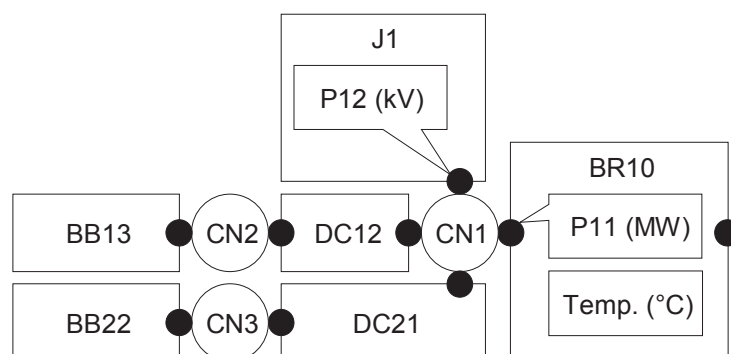
- each Measurement instance represents a technological quantity of a PowerSystemResource;
- each MeasurementValue of a Measurement represents a value for the technological quantity, as supplied from a single source;
- the source attribute in MeasurementValueQuality then indicates whether the source actually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

4.4.7.4 Attachment of measurements

As mentioned in 4.4.7.3 and as shown in Figure 13, 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 14 shows two examples of the placement of Measurements.

**Figure 14 – 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.

4.4.8 Regulating control models

Regulation control, such as automatic voltage control at generators or voltage tap control at transformers is modelled 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 15 shows how RegulatingCondEq and TapChanger classes can participate in regulation.

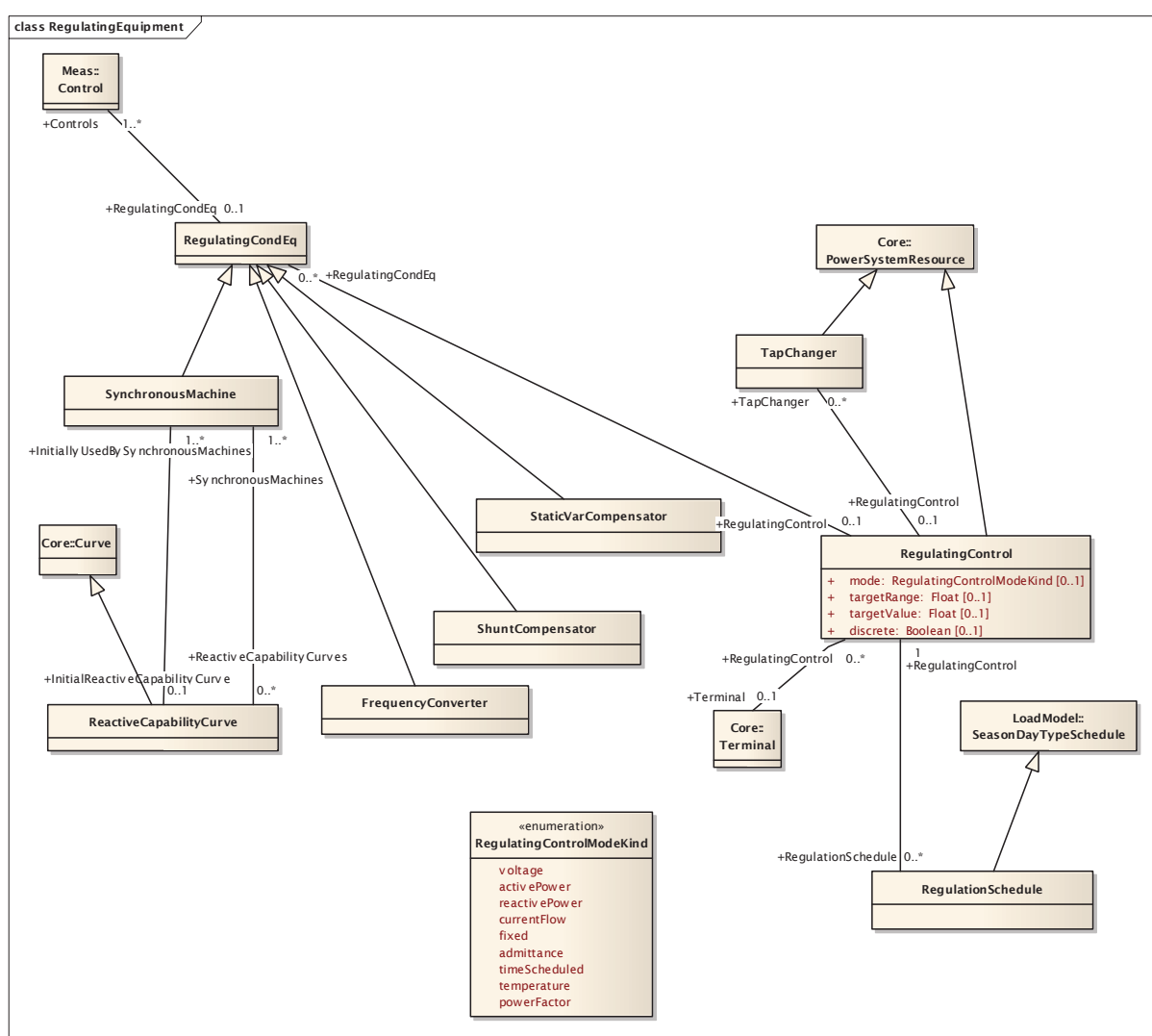


Figure 15 – Regulating control models

4.5 Modeling guidelines

4.5.1 Modeling for change

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 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 specific IEC 61970-3xx 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 IEC 61970-4xx 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 tools listed in the MTEI MTEI-wiki group available from <http://cimug.ucaiug.org>..

4.6 Modeling tools

Previous editions of CIM were created and maintained using IBM®'s Rational® ROSE¹, but this release was constructed using Sparx Systems Enterprise Architect² product. The entire CIM UML model exists as an Enterprise Architect 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 jCleanCim which uses the automation interfaces of Microsoft Word™³ and Enterprise Architect and using the standard IEC format and styles.

4.7 User implementation conventions

This subclause 4.7 provides recommended user conventions when using the CIM in actual system implementations.

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

¹ 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.

² 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.

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