BS EN 61970-301:2014

BSI Standards Publication

Energy management system application program interface (EMS-API)

Part 301: Common information model (CIM) base

bsi.

...making excellence a habit.™

National foreword

This British Standard is the UK implementation of EN 61970-301:2014. It is identical to IEC 61970-301:2013. It supersedes BS EN 61970-301:2013 which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee PEL/57, Power systems management and associated information exchange.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Published by BSI Standards Limited 2014

ISBN 978 0 580 81664 2
ICS 33.200

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 August 2014.

Amendments/corrigenda issued since publication

Date	Text affected

EUROPEAN STANDARD
NORME EUROPÉENNE
EUROPÄISCHE NORM

EN 61970-301

July 2014

ICS 33.200

Supersedes EN 61970-301:2013

English Version

Energy management system application program interface (EMS-API) - Part 301: Common information model (CIM) base (IEC 61970-301:2013)

Interface de programmation d'application pour système de gestion d'énergie (EMS-API) - Part 301: Base de modèle d'information commun (CIM)
(CEI 61970-301:2013)

Schnittstelle für Anwendungsprogramme für Netzführungssysteme (EMS-API) - Teil 301: Allgemeines Informationsmodell (CIM)
(IEC 61970-301:2013)

This European Standard was approved by CENELEC on 2014-01-17. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

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

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

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

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

CEN-CENELEC Management Centre: Avenue Marnix 17, B-1000 Brussels

Ref. No. EN 61970-301:2014 E

Foreword

The text of document 57/1395/FDIS, future edition 5 of IEC 61970-301, prepared by IEC TC 57 "Power systems management and associated information exchange" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN 61970-301:2014.

The following dates are fixed:

• latest date by which the document has to be implemented at national level by publication of an identical national standard or by endorsement	(dop)	2015-01-18
• latest date by which the national standards conflicting with the document have to be withdrawn	(dow)	2017-01-17

This document supersedes EN 61970-301:2013.

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

This document has been prepared under a mandate given to CENELEC by the European Commission and the European Free Trade Association.

Endorsement notice

The text of the International Standard IEC 61970-301:2013 was approved by CENELEC as a European Standard without any modification.

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

IEC 61968-11	NOTE Harmonised in EN 61968-11.
IEC 61970-501	NOTE Harmonised as EN 61970-501.

Annex ZA
(normative)

Normative references to international publications with their corresponding European publications

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.

NOTE When an international publication has been modified by common modifications, indicated by (mod), the relevant EN/HD applies.

Publication	Year	Title	EN/HD	Year
IEC 60050	(Series)	International Electrotechnical Vocabulary (IEV)	-	-
IEC 60870-6	(Series)	Telecontrol equipment and systems - Part 6-802: Telecontrol protocols compatible with ISO standards and ITU-T recommendations - TASE.2 Object models	EN 60870-6	(Series)
IEC 61850	(Series)	Communication networks and systems for power utility automation	EN 61850	(Series)
IEC 61850-7-3	2010	Communication networks and systems for power utility automation - Part 7-3: Basic communication structure - Common data classes	EN 61850-7-3	2011
IEC 61850-7-4	2010	Communication networks and systems for power utility automation - Part 7-4: Basic communication structure - Compatible logical node classes and data object classes	EN 61850-7-4	2010
IEC 61968	(Series)	Application integration at electric utilities - System interfaces for distribution management	EN 61968	(Series)
IEC/TS 61970-2		Energy management system application program interface (EMS-API) - Part 2: Glossary	CLC/TS 61970-2	
IEC 62325	(Series)	Framework for energy market communications	EN 62325	(Series)
UML 2.0 - Object Group Management		Object Management Group: UML 2.0 Specification	-	-

CONTENTS

INTRODUCTION

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

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

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

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

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

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

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

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

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

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

ICL

Wenlock Way

West Gorton

Manchester

M12 5DR

United Kingdom (U.K.)

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

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

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

Part 301: Common information model (CIM) base

1 Scope

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

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

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

2 Normative references

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

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

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

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

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

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

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

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

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

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

3 Terms and definitions

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

3.1
energy management system
EMS
computer system comprising a software platform providing basic support services and a set of applications providing the functionality needed for the effective operation of electrical generation and transmission facilities so as to ensure adequate security of energy supply at minimum cost

3.2
application program interface
API
the set of public functions provided by an executable application component for use by other executable application components

3.3
unified modeling language
UML
formal and comprehensive descriptive language with diagramming techniques used to represent software systems, from requirements analysis, through design and implementation, to documentation

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

3.4
profile
subset of DCIM classes, associations and attributes needed to accomplish a specific type of interface

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

4 CIM specification

4.1 CIM modeling notation

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

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

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

4.2 CIM packages

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

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

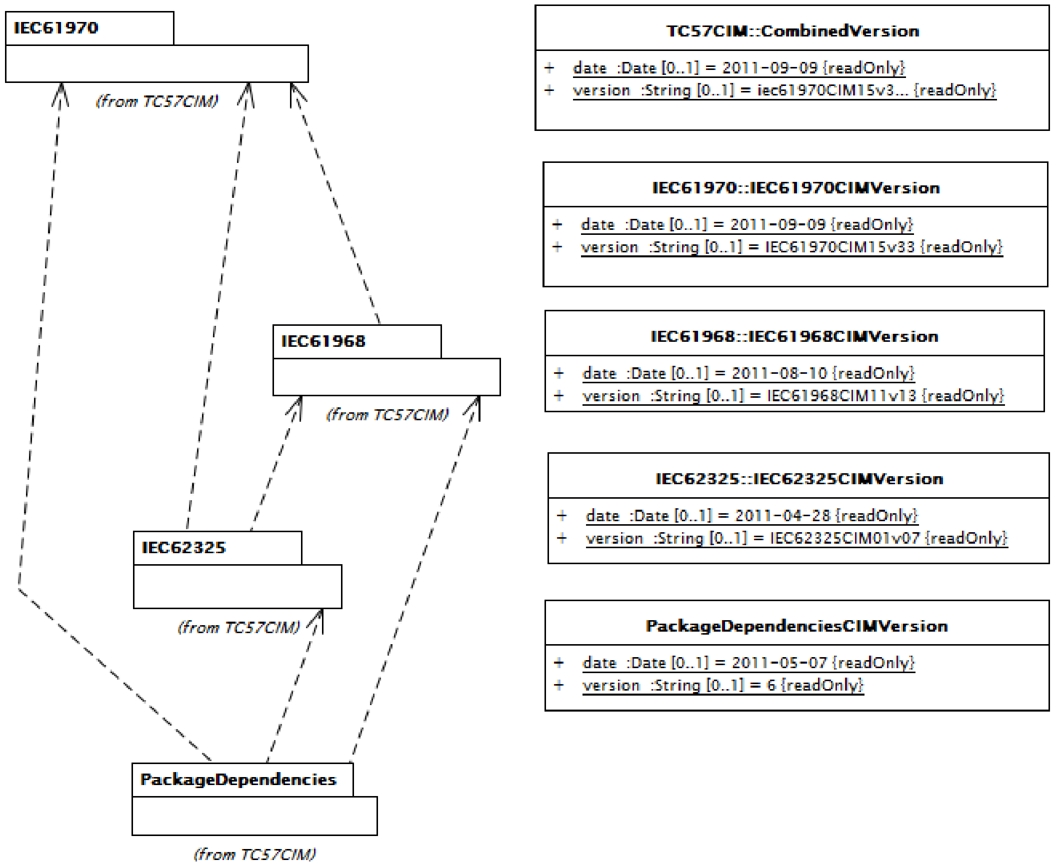

Figure 1 – Example of working group package dependencies

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

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

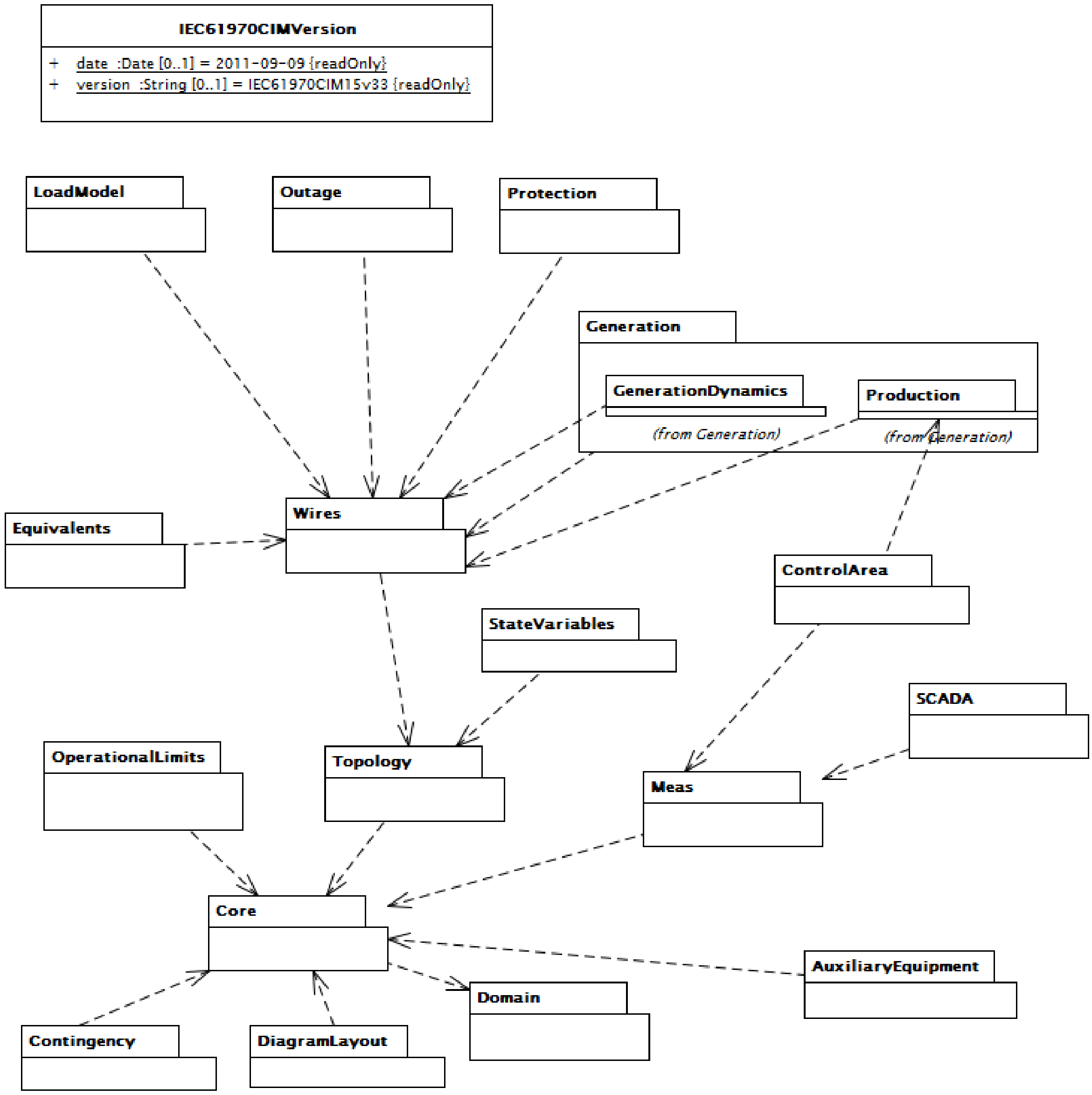

Figure 2 – CIM IEC 61970-301 package diagram

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

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

4.3 CIM classes and relationships

4.3.1 Classes

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

Classes and objects model what is in a power system that needs to be represented in a common way to power system applications. A class is a description of an object found in the real world, such as a power transformer, generator, or load that needs to be represented as part of the overall power system model in, for example, an EMS. Other types of objects include things such as schedules and measurements that EMS applications also need to process, analyze, and store. Such objects need a common representation to achieve the purposes of the EMS-API standard for plug-compatibility and interoperability. A particular object in a power system with a unique identity is modeled as an instance of the class to which it belongs.

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

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

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

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

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

4.3.2 Generalization

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

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

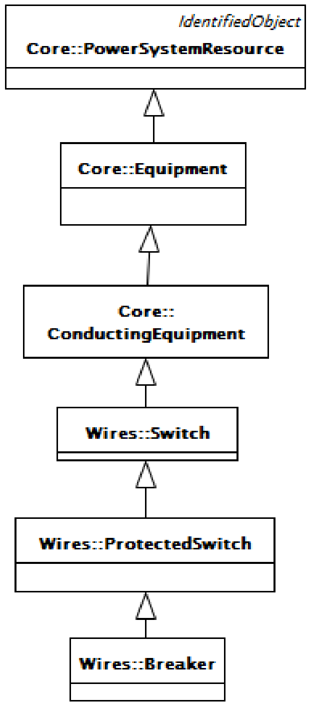

Figure 3 – Example of generalization

4.3.3 Simple association

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

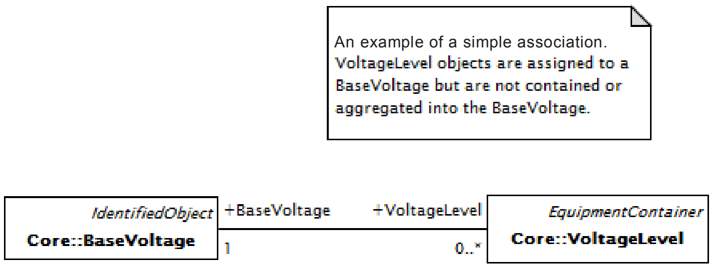

Figure 4 – Example of simple association

4.3.4 Aggregation

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

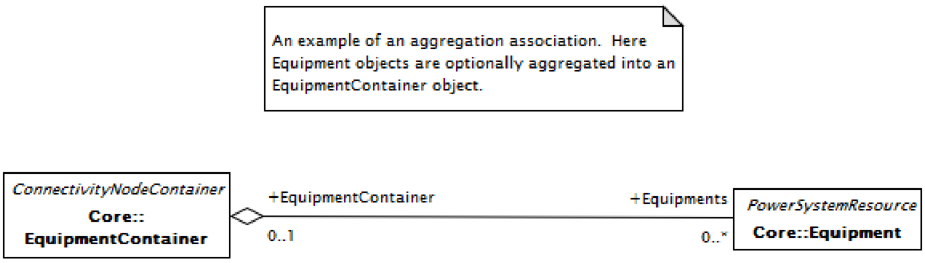

Figure 5 – Example of aggregation

4.4 CIM model concepts and examples

4.4.1 Concepts

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

4.4.2 Containment, equipment hierarchies and naming

4.4.2.1 Containment structure

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

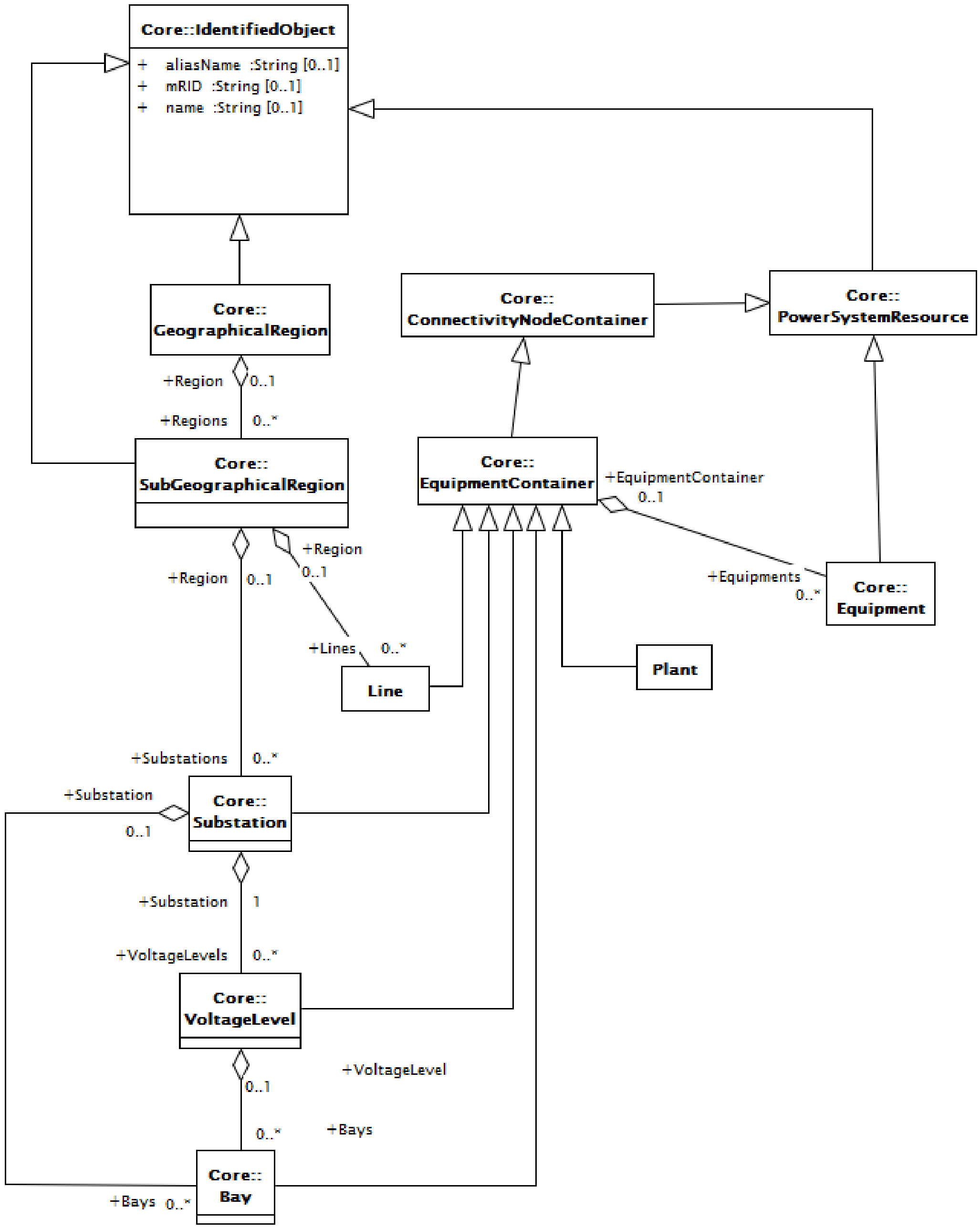

Figure 6 – Equipment containers

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

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

4.4.2.2 IdentifiedObject class

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

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

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

4.4.3 Names model

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

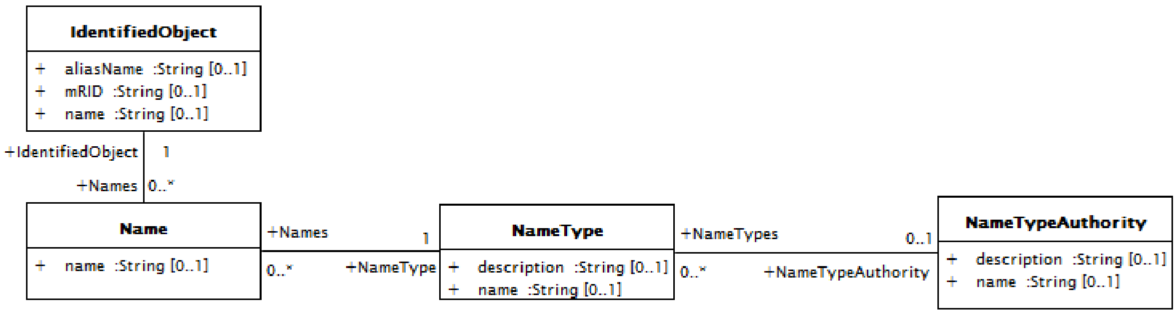

Figure 7 – Names

Table 1 – NameType class naming conventions

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

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

name: The name attribute value.

description: The use of the name type.

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

4.4.4 Connectivity model

4.4.4.1 Connectivity description

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

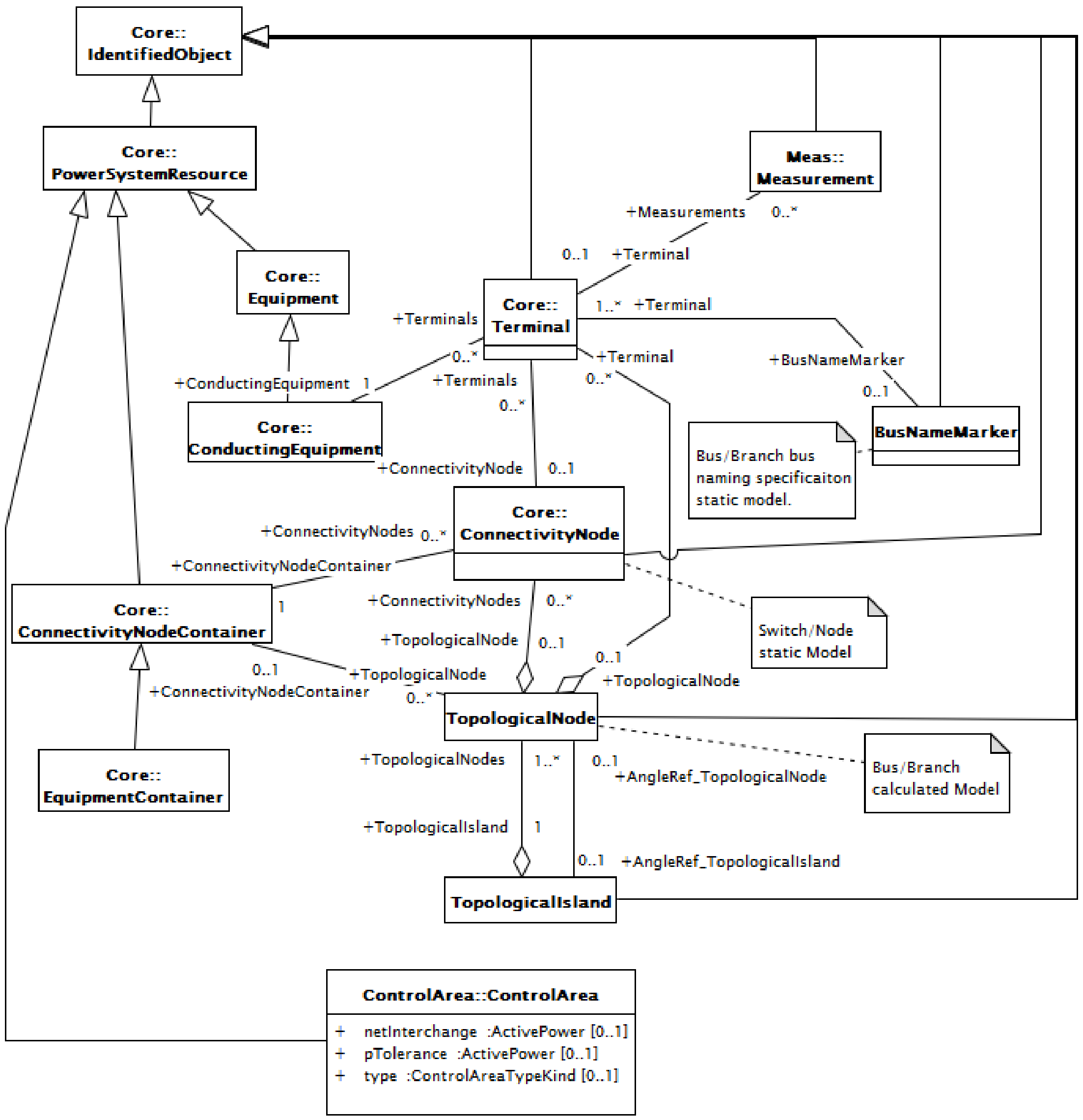

Figure 8 – Connectivity model

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

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

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

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

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

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

4.4.4.2 Connectivity and containment example

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

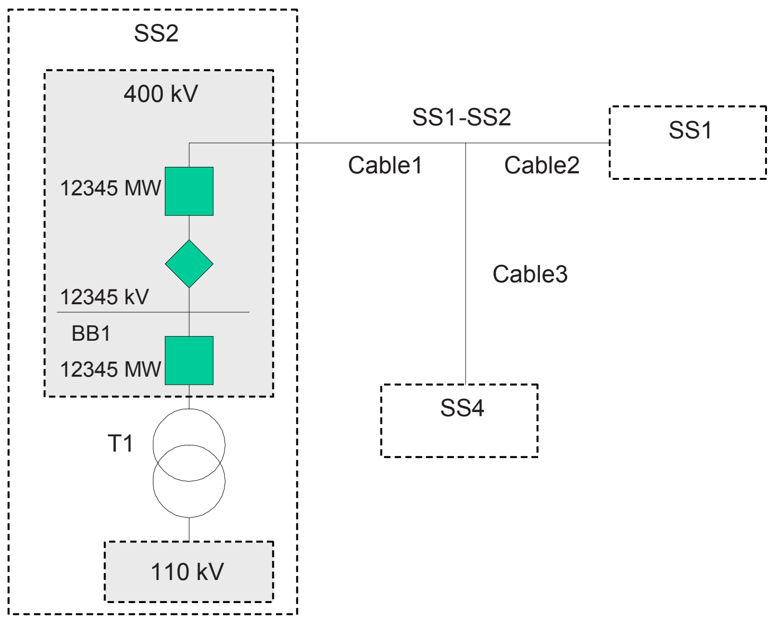

Figure 9 – Simple network example

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

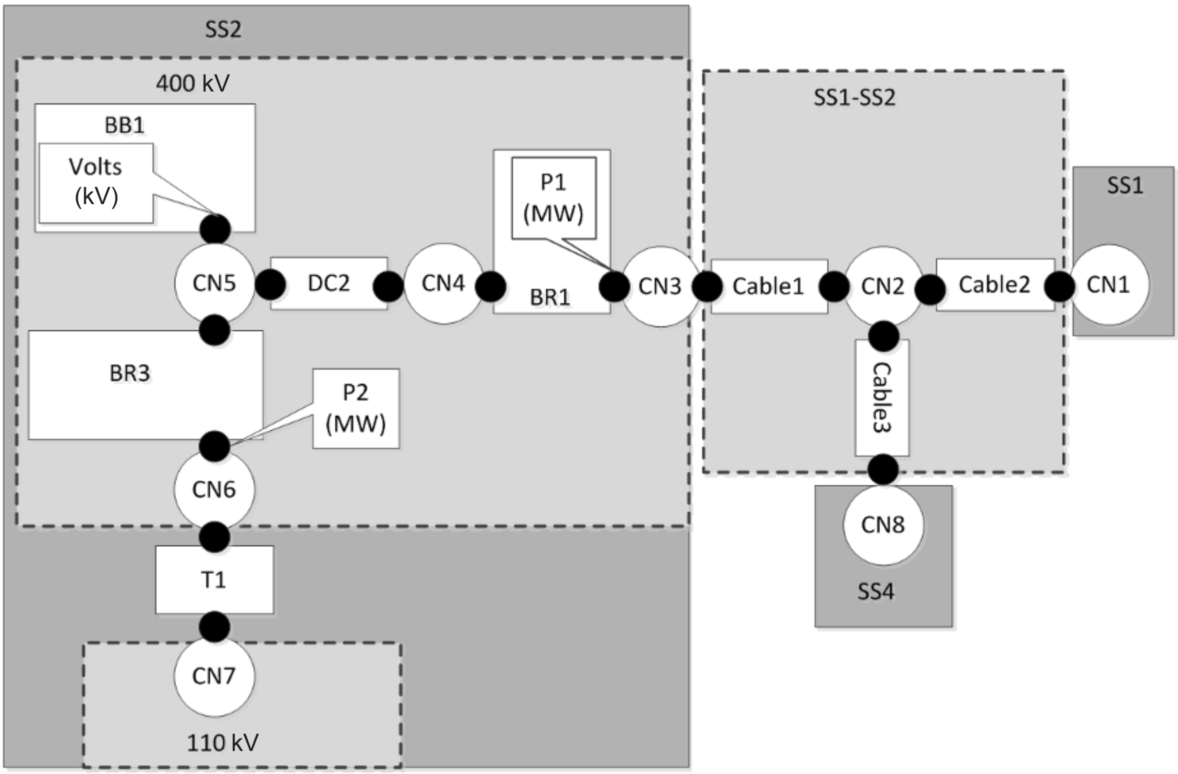

Figure 10 – Simple network connectivity modeled with CIM Topology

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

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

4.4.5 Inheritance hierarchy

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

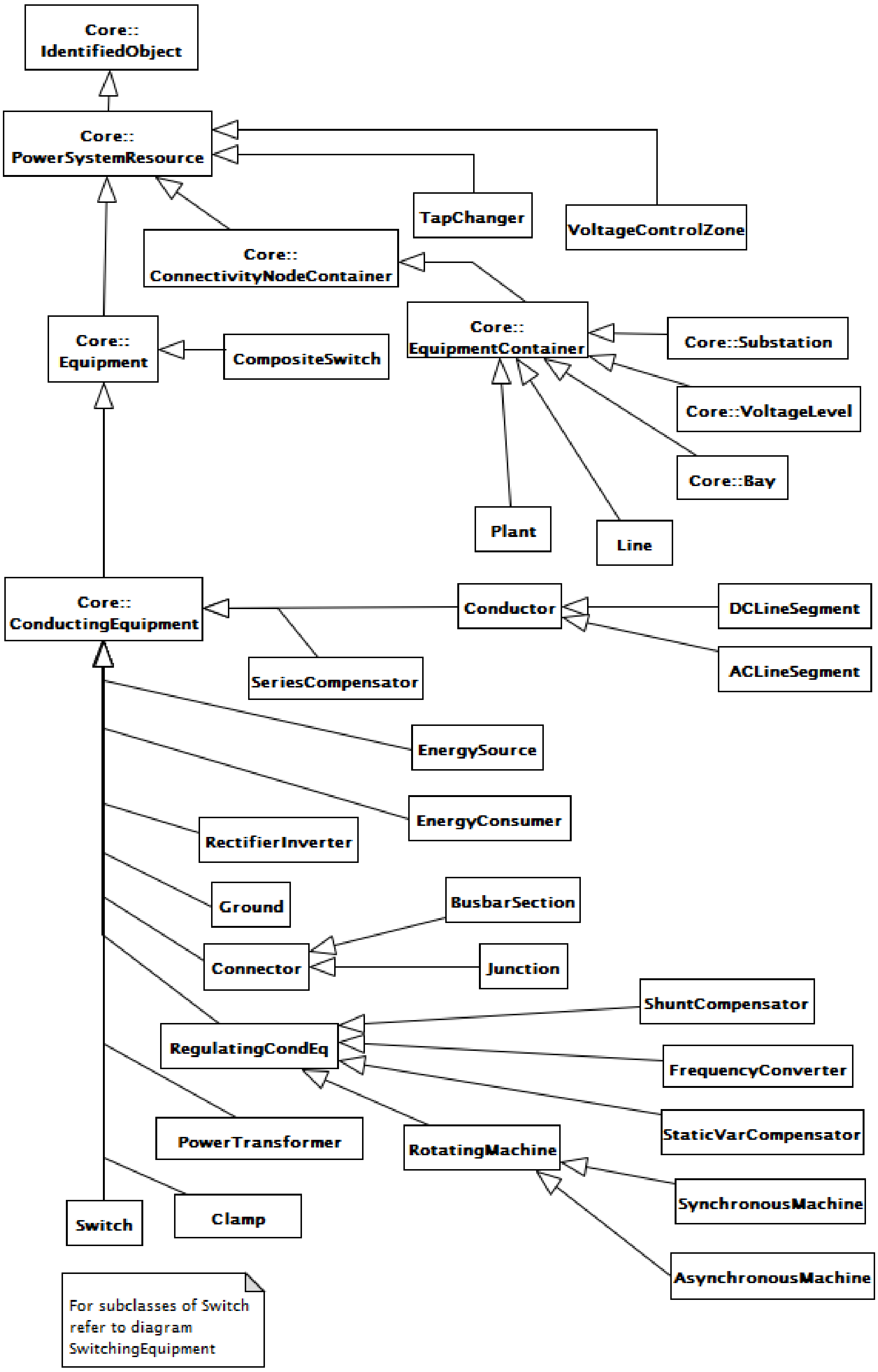

Figure 11 – Equipment inheritance hierarchy

4.4.6 Transformer model

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

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

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

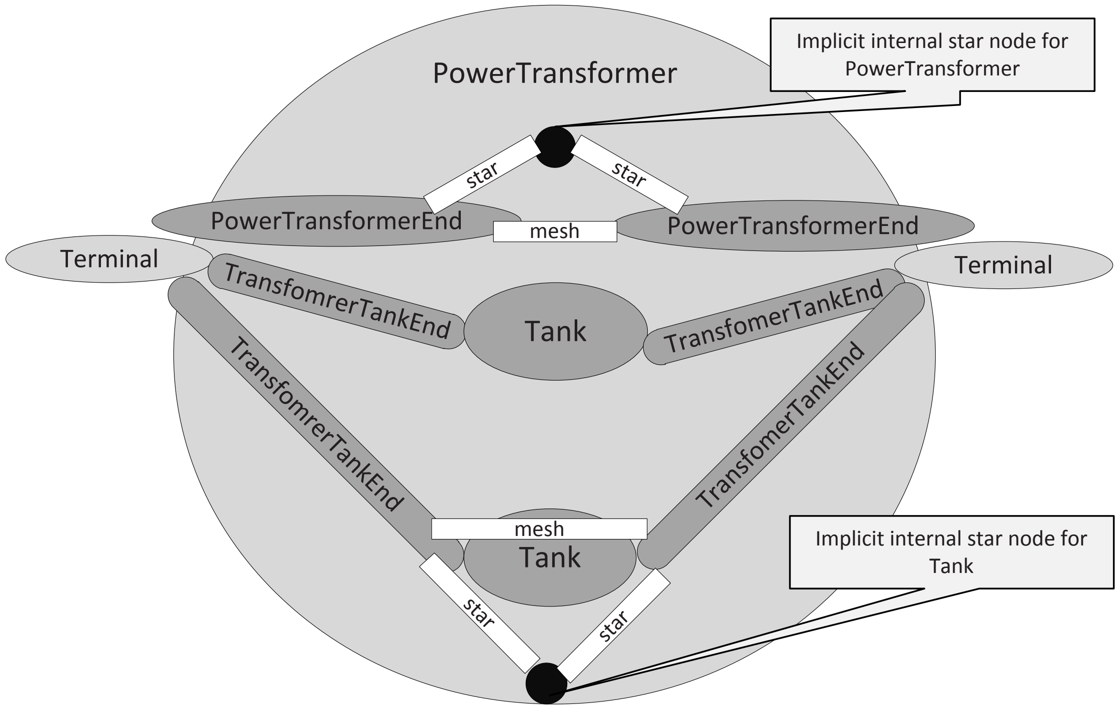

Figure 12 – Transformer and Tank model

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

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

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

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

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

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

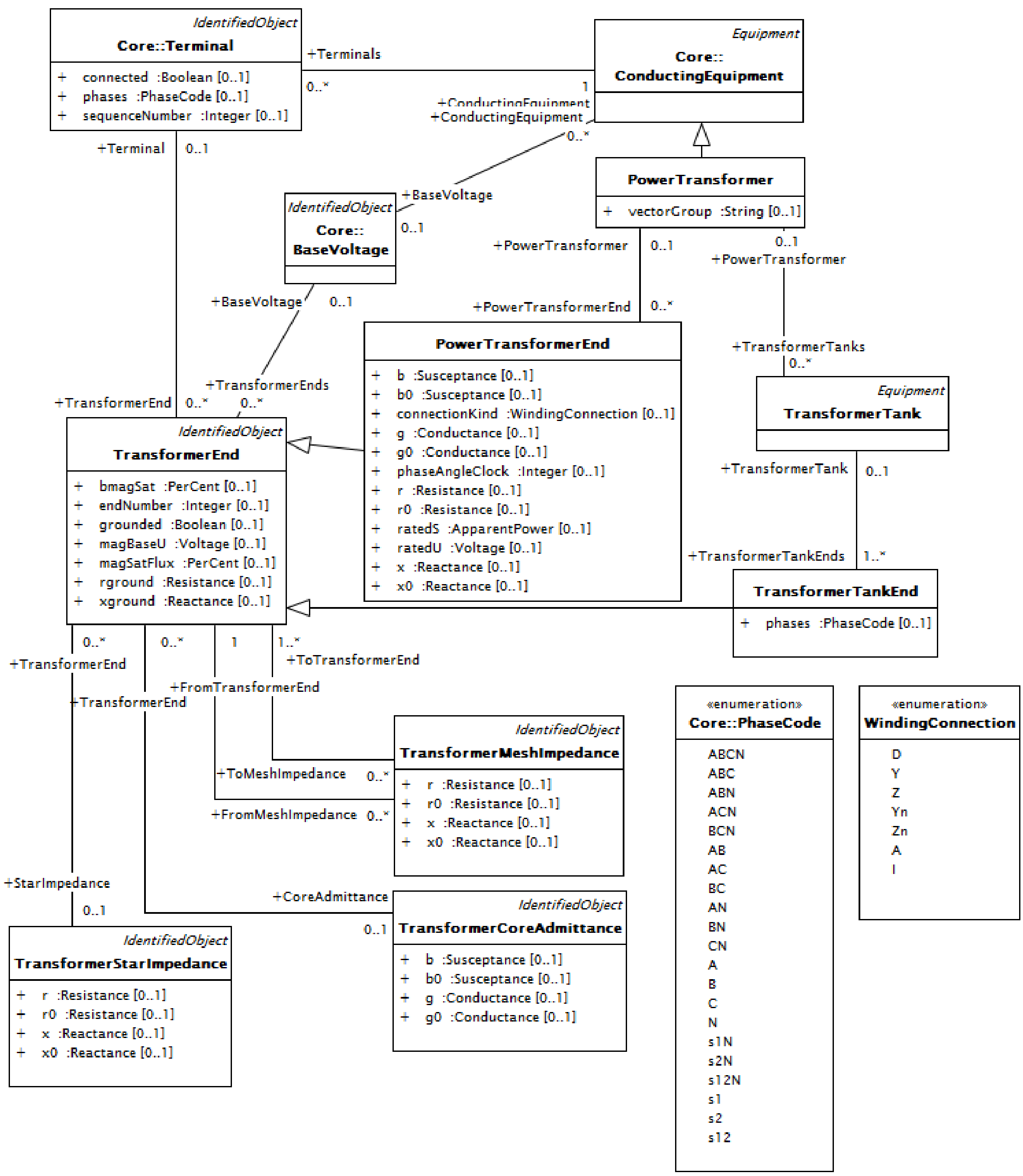

Figure 13 – Transformer model

4.4.7 Transformer tap modeling

4.4.7.1 Generalized tap modeling

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

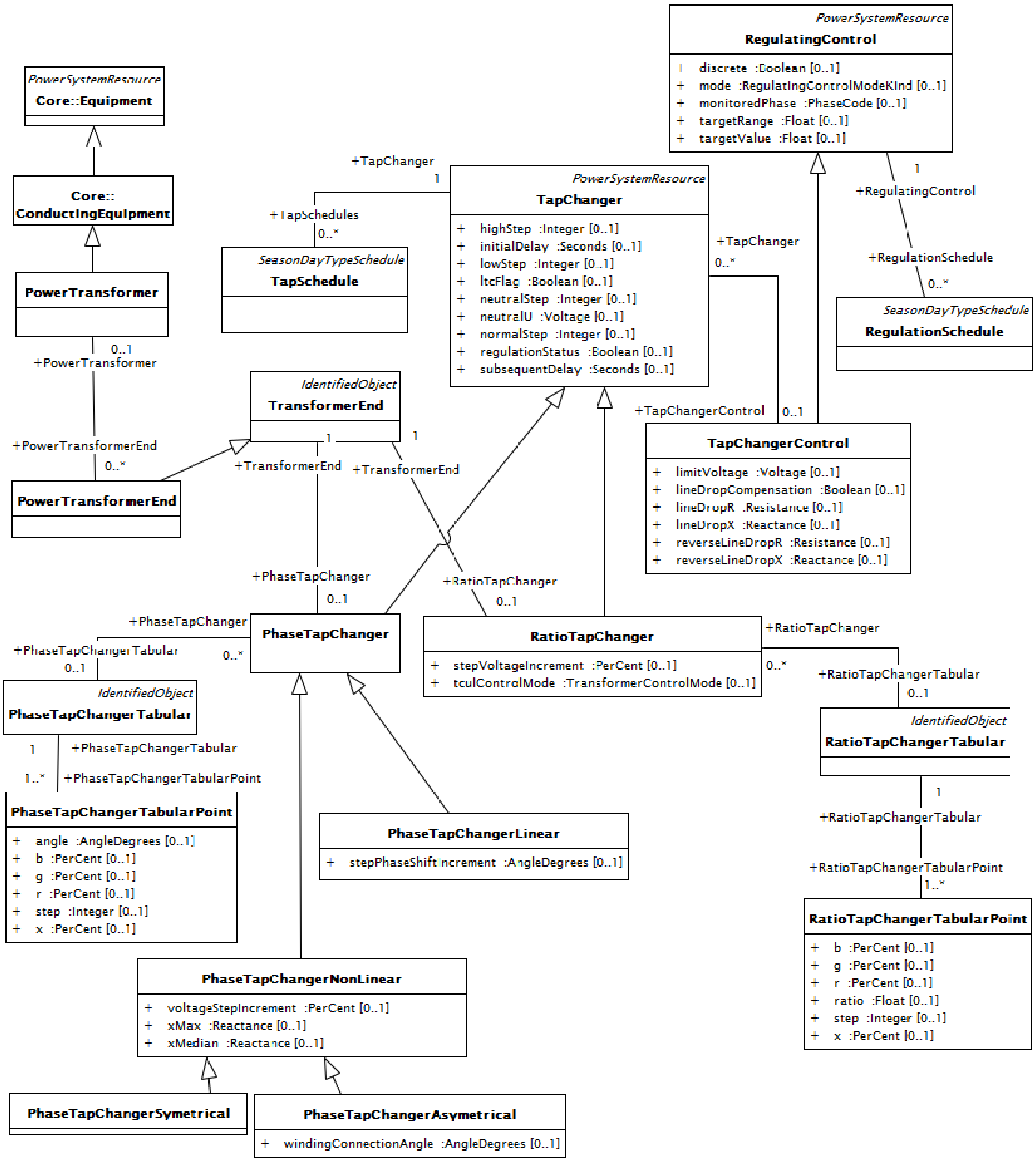

Figure 14 – Transformer tap model

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

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

4.4.7.2 Voltage ratio transformer modeling

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

4.4.7.3 Voltage ratio transformer tabular modeling

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

4.4.7.4 Linear phase shifter transformer modeling

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

4.4.7.5 Symmetrical phase shifter modeling

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

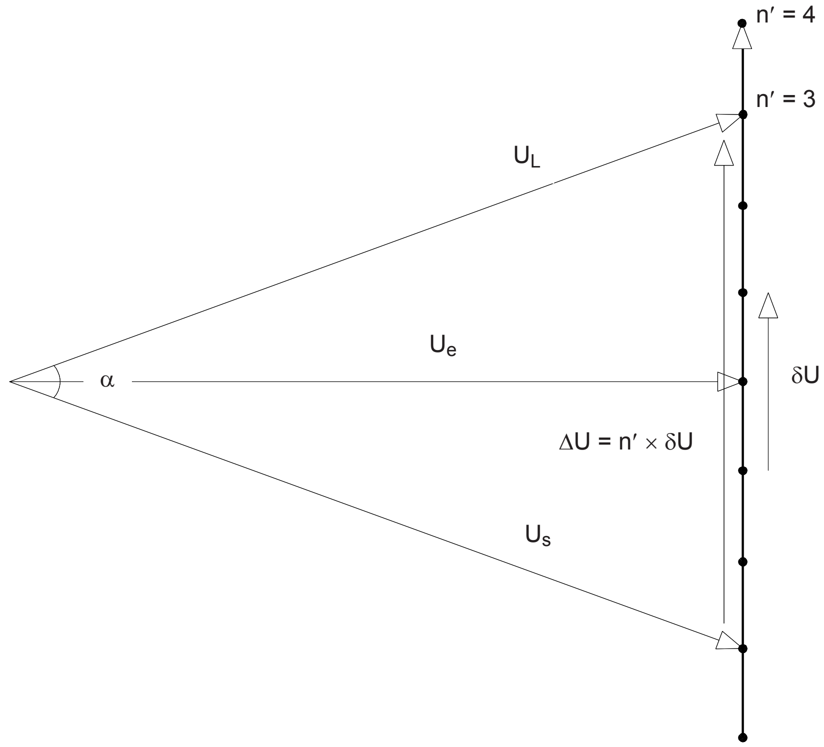

Figure 15 – Symmetrical Phase Shifter

The phase shift angle can be represented as:

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

$\rho = 1$

Where

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

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

4.4.7.6 Asymmetrical phase shifter modeling

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

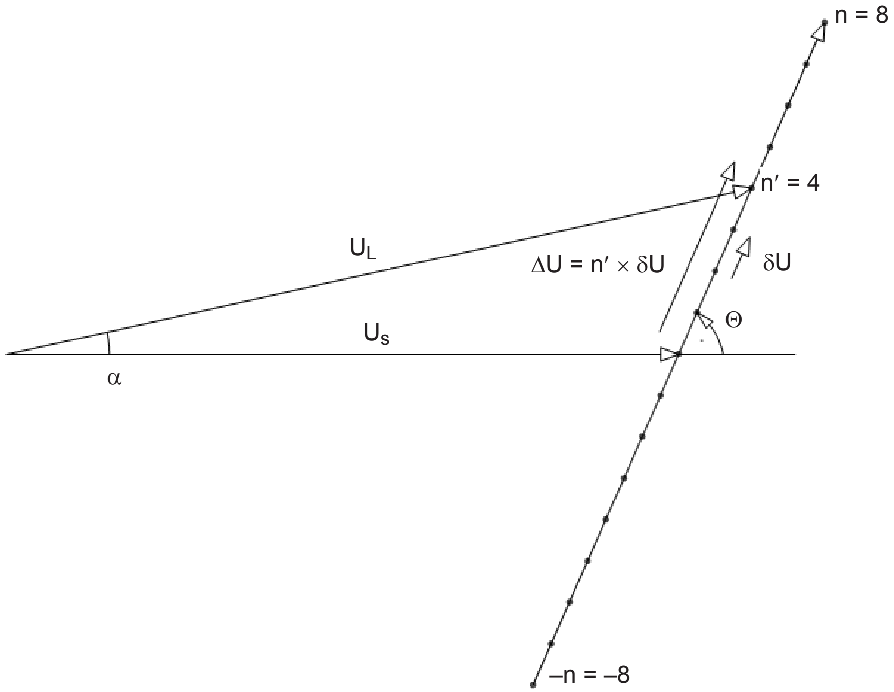

Figure 16 – Asymmetrical Phase Shifter

The phase shift angle can be represented as:

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

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

Where

α is the phase shift angle, local terminal relative to other nominal terminal;

n’ is the current tap position;

δU is the voltage change per tap position;

Θ is the winding connection angle;

ρ is the voltage magnitude transformation ratio, local terminal to other nominal terminal;

U_L is the local terminal voltage;

U_S is the system or other terminal voltage.

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

4.4.7.7 Phase shifter transformer tabular modeling

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

4.4.8 Phase wire modeling

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

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

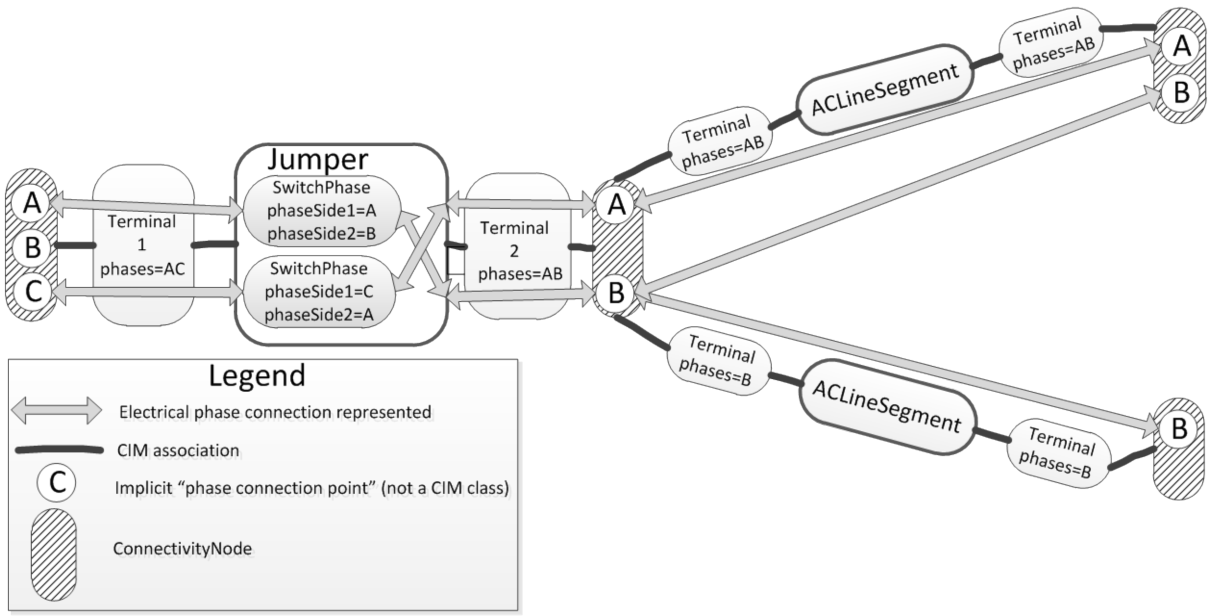

Figure 17 – Phase connectivity

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

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

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

4.4.9 Cuts, clamps and jumpers model

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

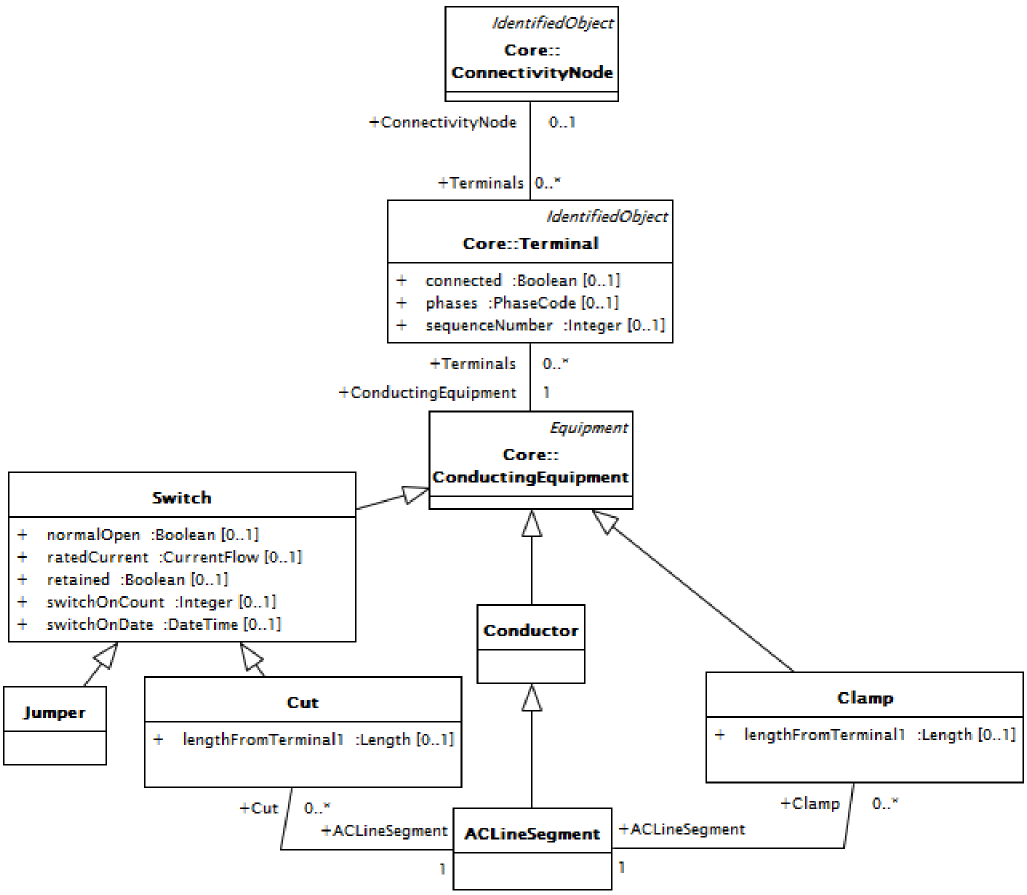

Figure 18 – Cuts, clamps, and jumpers UML model

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

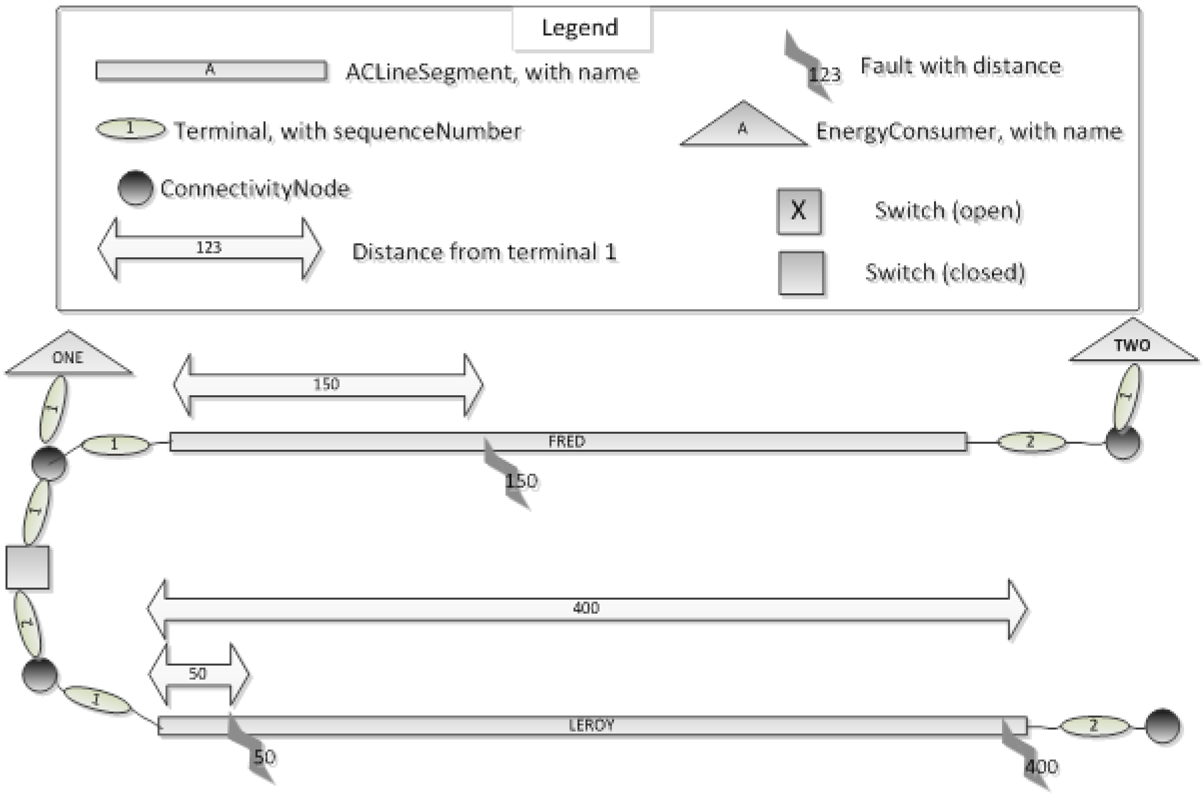

Figure 19 – Example before cuts and jumpers applied

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

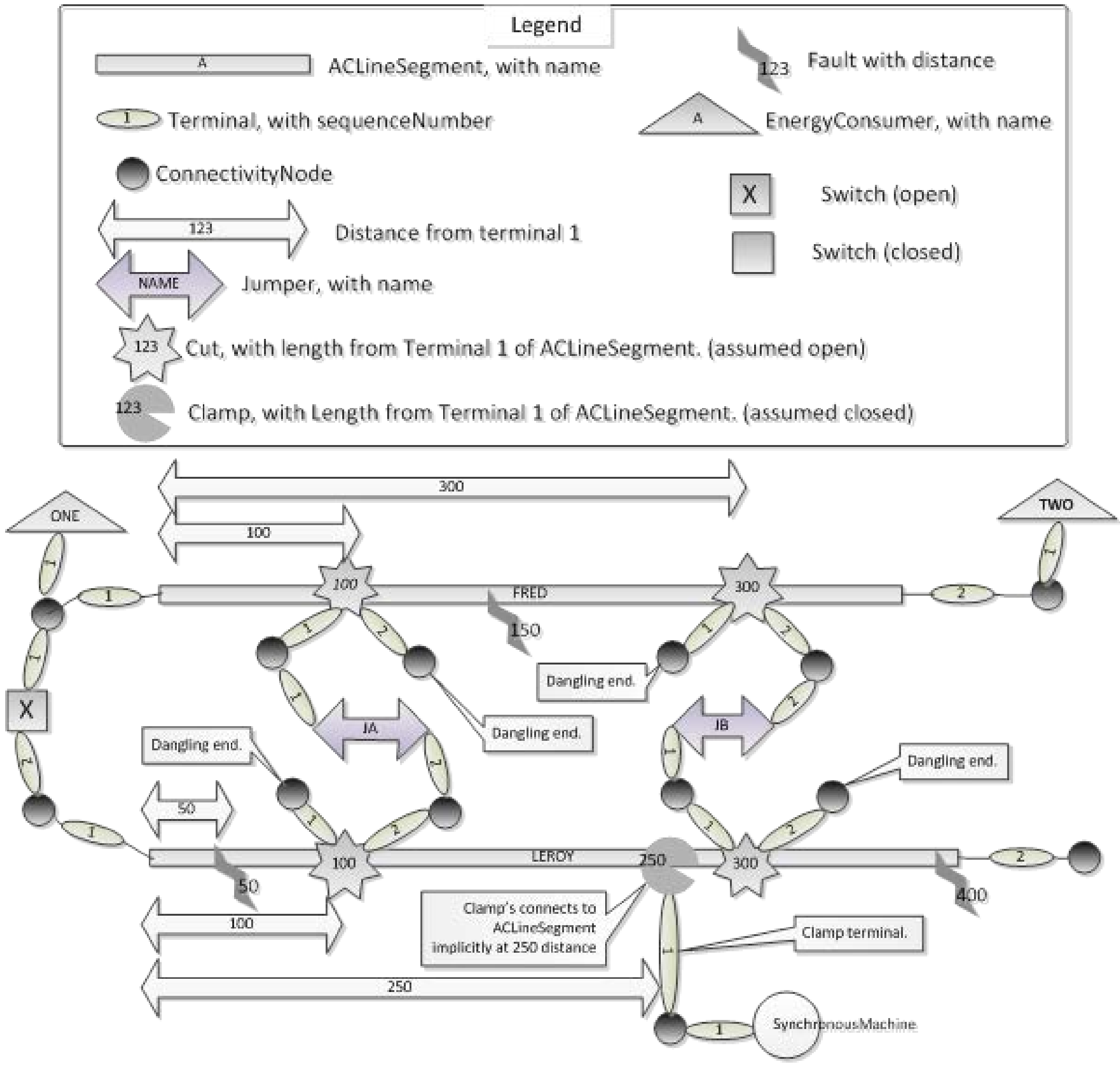

Figure 20 – Example after cuts and jumpers applied

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

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

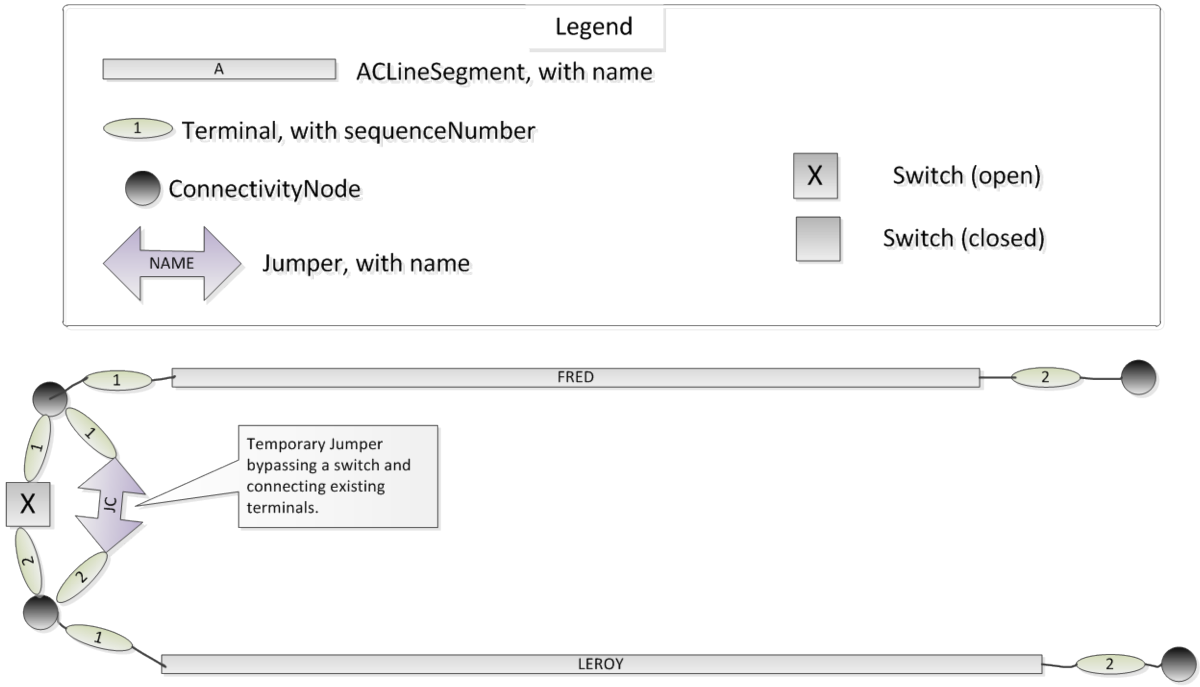

Figure 21 – Example of jumper without cut or clamp

4.4.10 Measurements and controls

4.4.10.1 Measurement overview

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

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

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

4.4.10.2 Control overview

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

4.4.10.3 Use of measurement-related classes

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

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

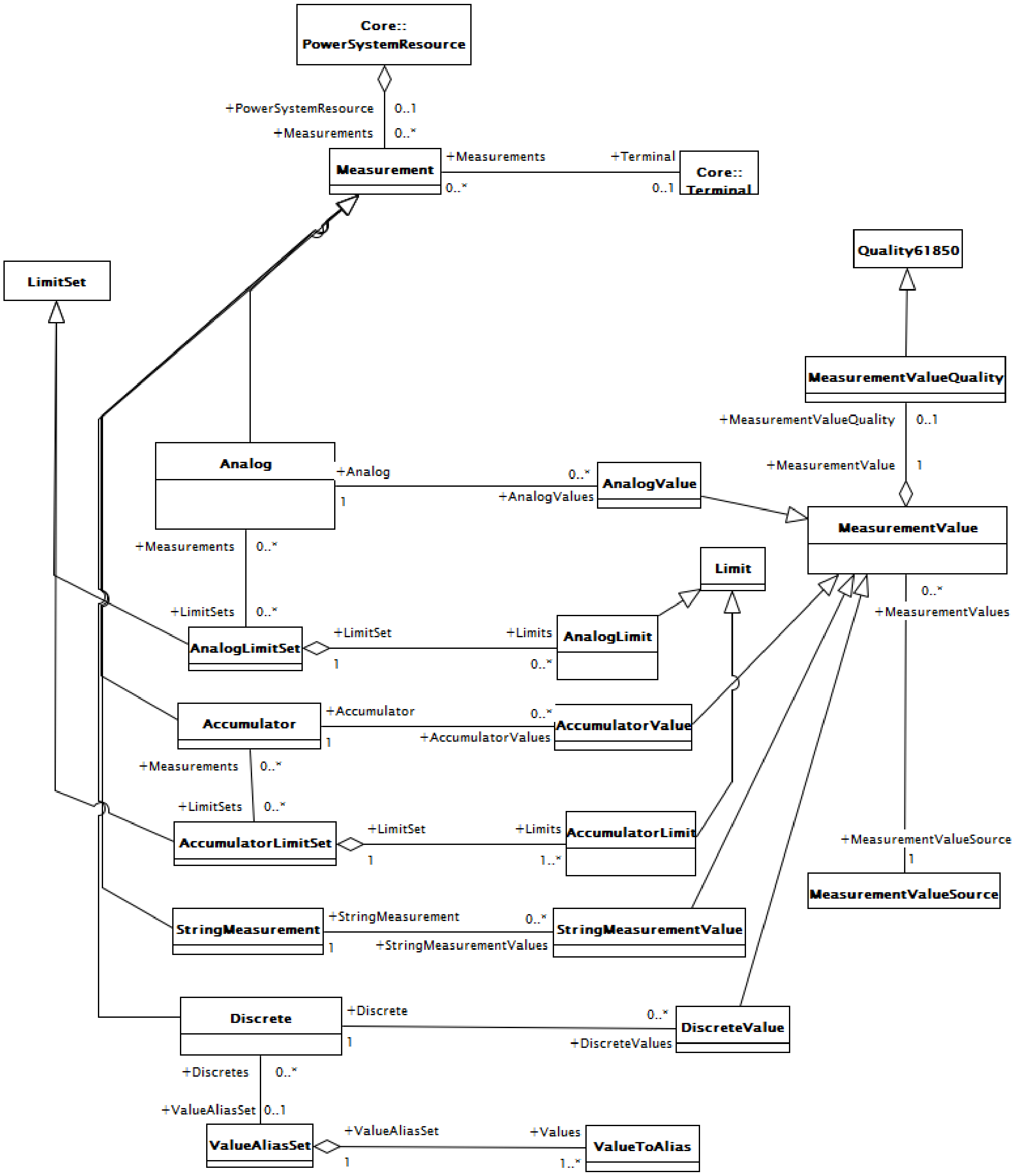

Figure 22 – Navigating from PSR to MeasurementValue

Table 2 – measurementType naming conventions

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

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

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

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

Table 3 – MeasurementValueSource naming conventions

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

Following these conventions:

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

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

4.4.10.4 Attachment of measurements

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

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

Figure 23 shows two examples of the placement of Measurements.

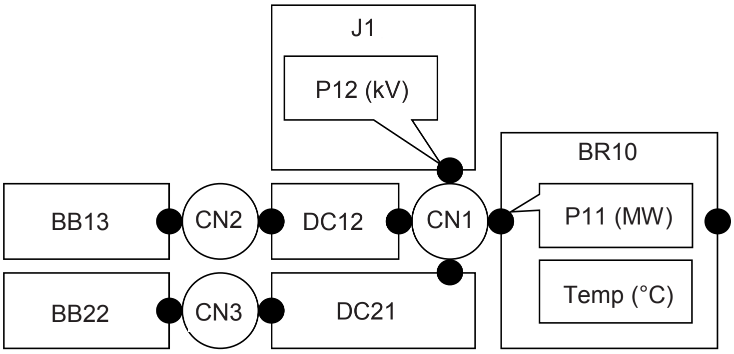

Figure 23 – Measurement placement

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

4.4.10.5 ICCP measurements

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

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

4.4.11 Regulating control models

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

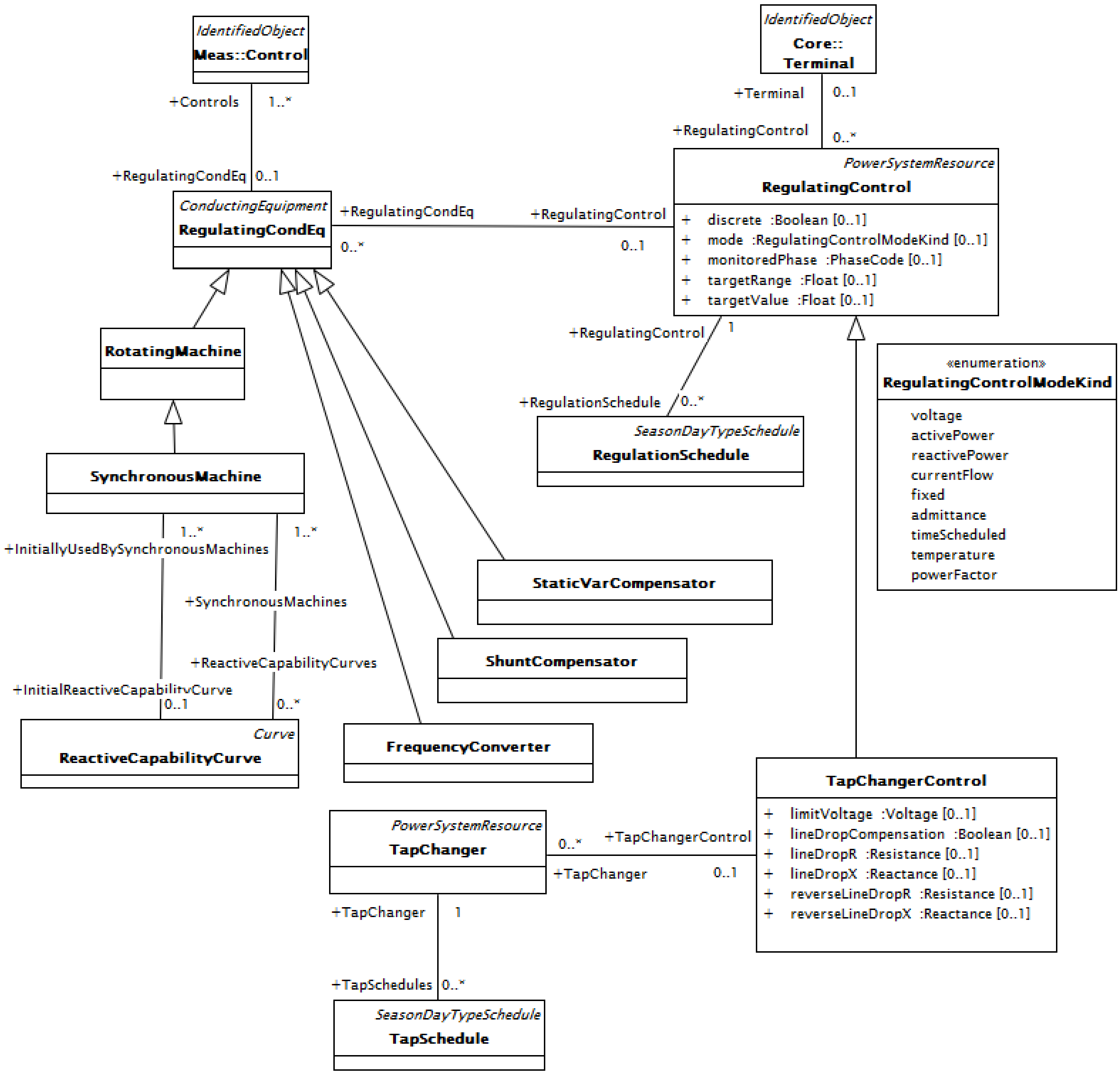

Figure 24 – Regulating control models

4.5 Modeling guidelines

4.5.1 Modeling for change

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

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

4.5.2 Process for amendments to the CIM

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

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

4.5.3 Changes to the CIM UML model

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

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

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

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

4.5.4 Changes to the CIM standards documents

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

4.5.5 Deprecations

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

4.5.6 CIM profiles

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

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

4.6 Modeling tools

This CIM release was constructed using Sparx Systems Enterprise Architect product. The entire CIM UML model exists as an Enterprise Architect[1] project file and is viewable with that tool, including the class diagrams and descriptions of classes, attributes, types, and relationships. Viewing the CIM in this fashion provides a graphical navigation interface that permits all CIM specification data to be viewed via point-and-click from the class diagram in each package. A free viewer is available through Sparx Systems.

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

Clause 6 of this standard was auto-generated using the open source tool jCleanCim, available at http://cimug.ucaiug.org/MTEI/Shared%20Documents/jCleanCim, which uses the automation interfaces of Microsoft Word™[2] and Enterprise Architect and the standard IEC format and styles.

4.7 User implementation conventions

4.7.1 Conventions beyond UML

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

4.7.2 Number of Terminals for ConductingEquipment objects

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

4.8 CIM modeling examples

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

---

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

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

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

It should be noted that an RDF schema version of the CIM is still metadata rather than an instantiation of an actual network. However, complete network model files with descriptions of all network elements and their electrical connectivity can be generated by system suppliers using proprietary export tools, and then imported by other systems via a similar import tool, which is used to populate a local network engineering tool database. Examples of such CIM XML models include the Siemens 100 bus model, the Alstom 60 bus model, and the ABB 40 bus model files[3)] used for CIM XML interoperability testing.

5 Detailed model

5.1 Overview

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

5.2 Context

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

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

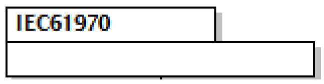

Figure 25 – CIM top level packages

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

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

Table 4 – Attributes of Package1::Class1

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

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

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

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

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

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

Table 6 – Literals of Package1::Enumeration1

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

6 Top package IEC61970

6.1 General

Top package for IEC 61970.

Figure 26 shows class diagram IEC61970Dependencies.

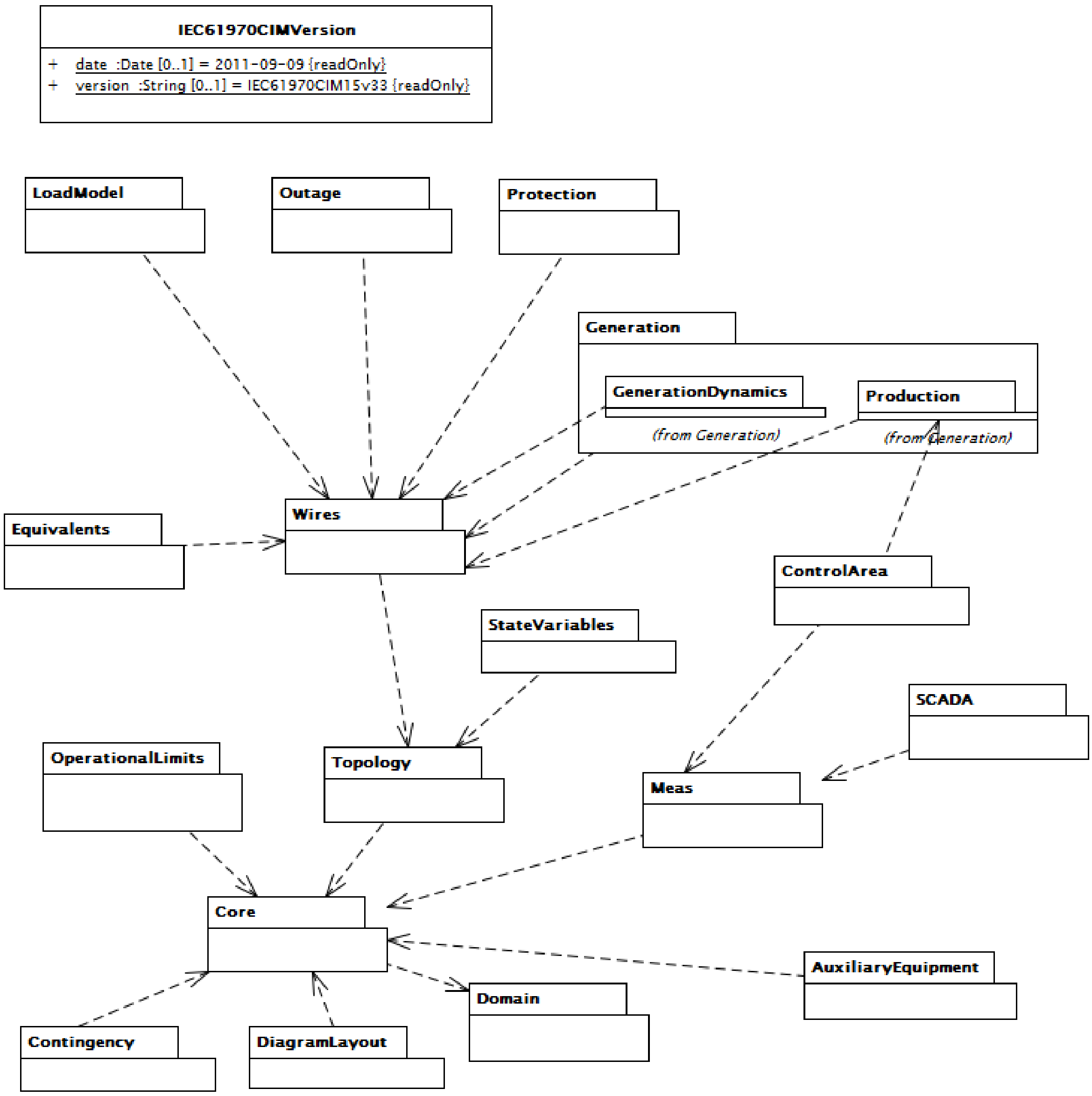

Figure 26 – Class diagram IEC61970::IEC61970Dependencies

This diagram shows all 61970 packages and their logical dependencies.

6.2 IEC61970CIMVersion root class

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

Table 7 shows all attributes of IEC61970CIMVersion.

Table 7 – Attributes of IEC61970::IEC61970CIMVersion

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

6.3 Package Domain

6.3.1 General

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

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

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

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

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

Figure 27 shows class diagram BasicDatatypes.

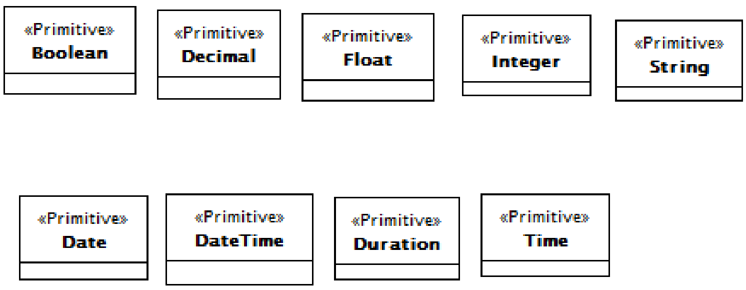

Figure 27 – Class diagram Domain::BasicDatatypes

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

Figure 28 shows class diagram ElectricityDatatypes.

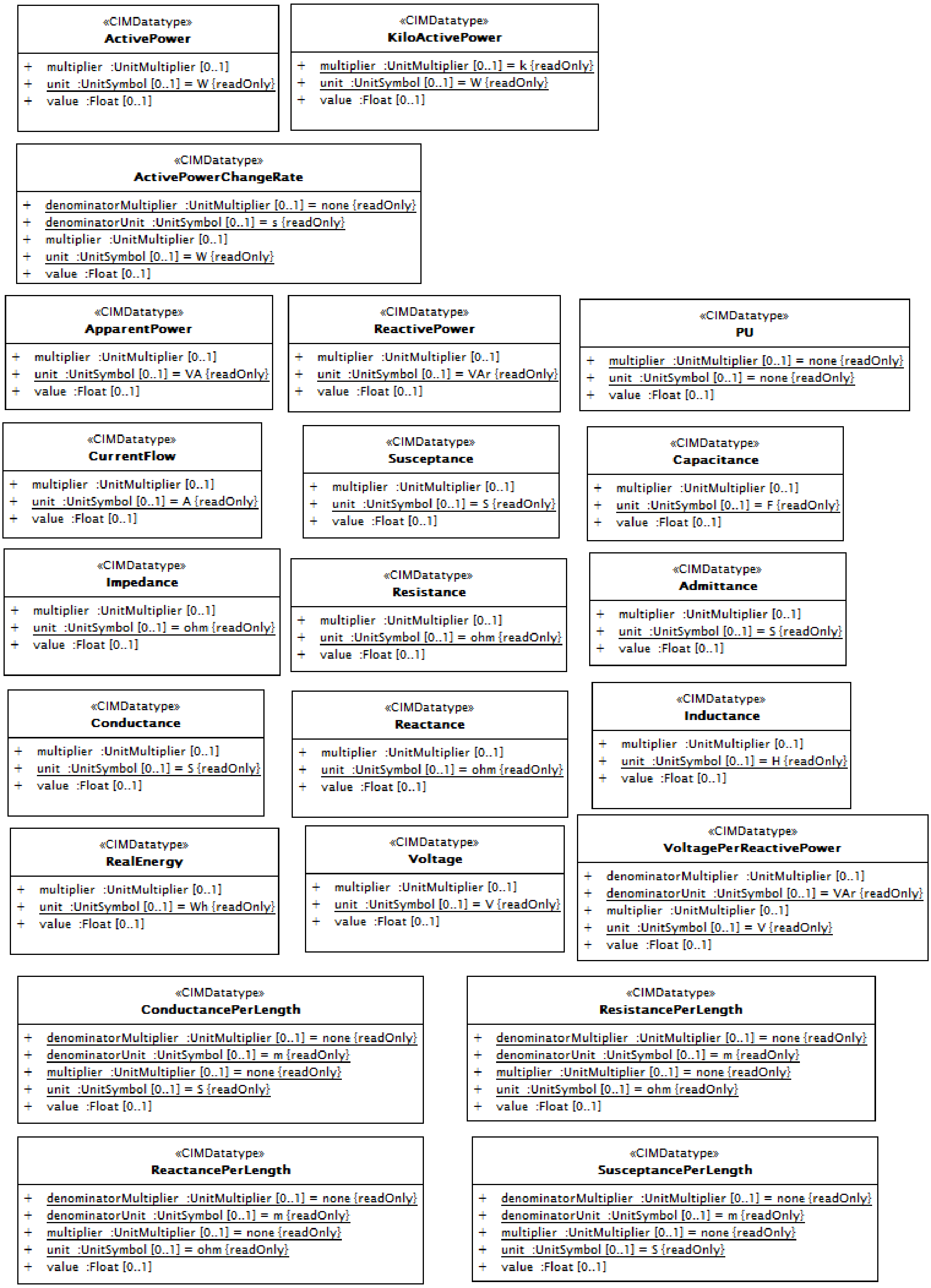

Figure 28 – Class diagram Domain::ElectricityDatatypes

This diagram shows electricity related data types.

Figure 29 shows class diagram EnumeratedUnitDatatypes.

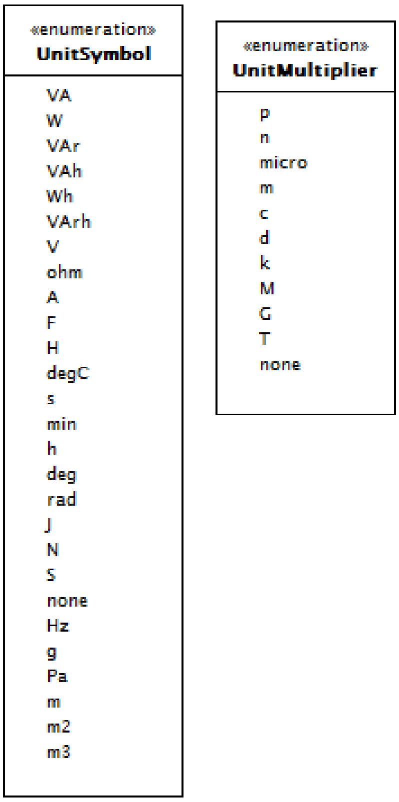

Figure 29 – Class diagram Domain::EnumeratedUnitDatatypes

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

Figure 30 shows class diagram GeneralDatatypes.

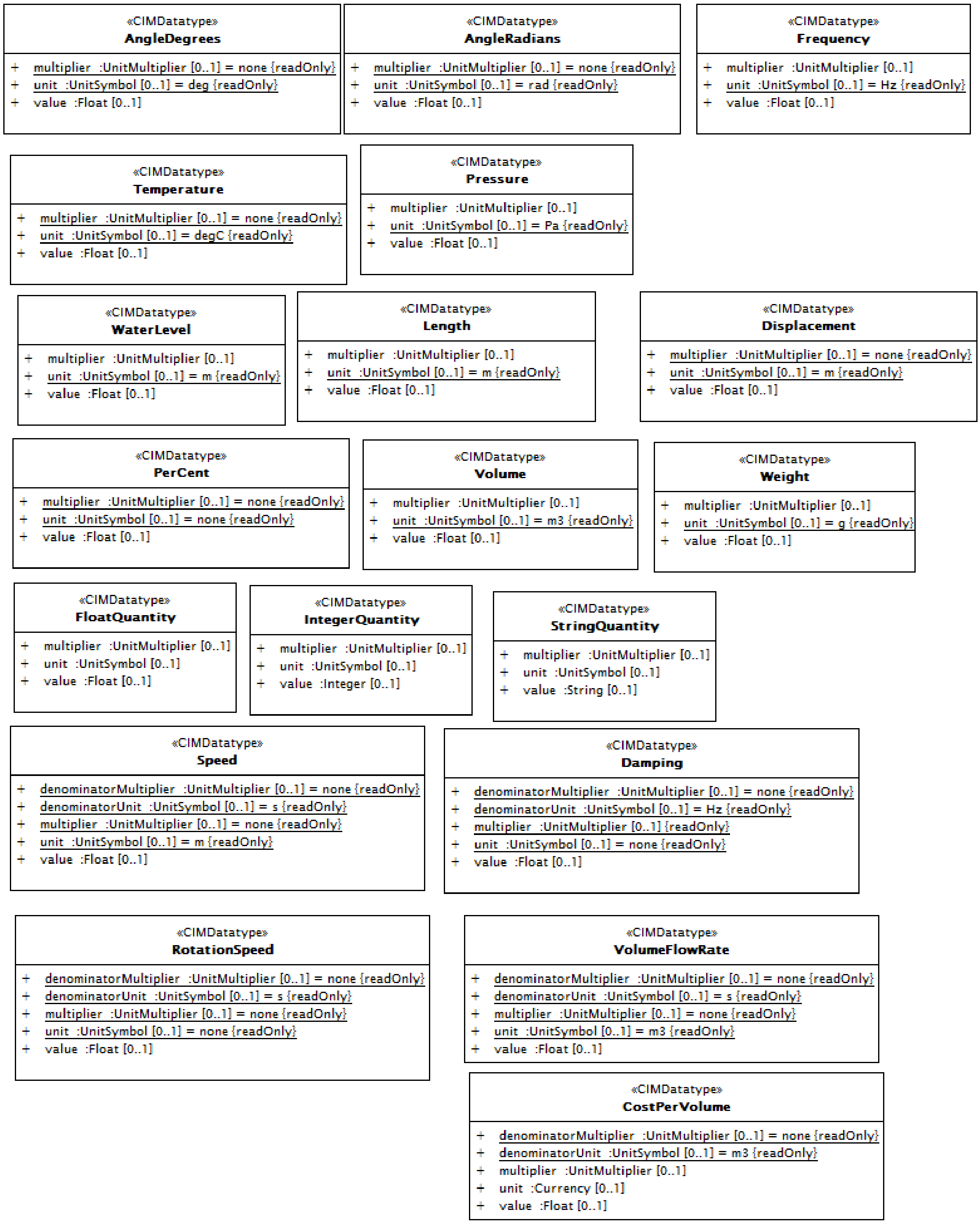

Figure 30 – Class diagram Domain::GeneralDatatypes

This diagram shows general data types.

Figure 31 shows class diagram MonetaryDatatypes.

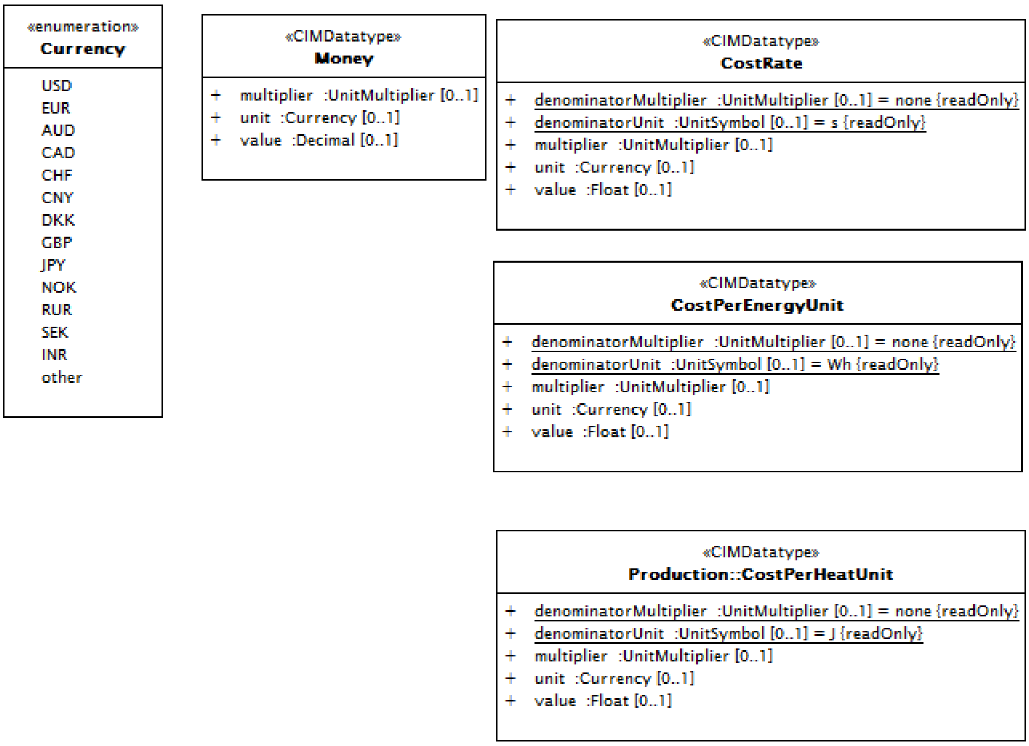

Figure 31 – Class diagram Domain::MonetaryDatatypes

This diagram shows monetary related data types.

Figure 32 shows class diagram TimeDatatypes.

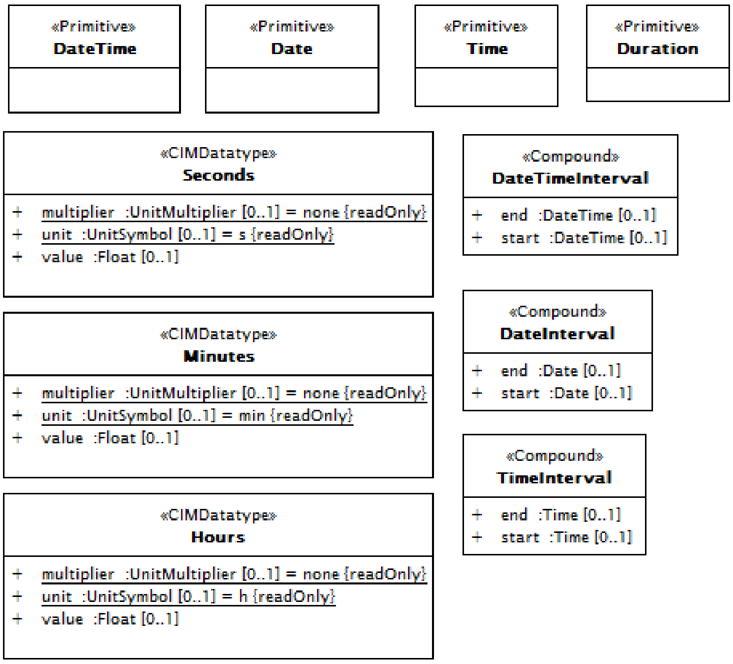

Figure 32 – Class diagram Domain::TimeDatatypes

This diagram shows time related data types.

6.3.2 ActivePower datatype

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

Table 8 shows all attributes of ActivePower.

Table 8 – Attributes of Domain::ActivePower

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

6.3.3 ActivePowerChangeRate datatype

Rate of change of active power per time.

Table 9 shows all attributes of ActivePowerChangeRate.

Table 9 – Attributes of Domain::ActivePowerChangeRate

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

6.3.4 Admittance datatype

Ratio of current to voltage.

Table 10 shows all attributes of Admittance.

Table 10 – Attributes of Domain::Admittance

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

6.3.5 AngleDegrees datatype

Measurement of angle in degrees.

Table 11 shows all attributes of AngleDegrees.

Table 11 – Attributes of Domain::AngleDegrees

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

6.3.6 AngleRadians datatype

Phase angle in radians.

Table 12 shows all attributes of AngleRadians.

Table 12 – Attributes of Domain::AngleRadians

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

6.3.7 ApparentPower datatype

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

Table 13 shows all attributes of ApparentPower.

Table 13 – Attributes of Domain::ApparentPower

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

6.3.8 Boolean primitive

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

6.3.9 Capacitance datatype

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

Table 14 shows all attributes of Capacitance.

Table 14 – Attributes of Domain::Capacitance

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

6.3.10 Conductance datatype

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

Table 15 shows all attributes of Conductance.

Table 15 – Attributes of Domain::Conductance

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

6.3.11 ConductancePerLength datatype

Real part of admittance per unit of length.

Table 16 shows all attributes of ConductancePerLength.

Table 16 – Attributes of Domain::ConductancePerLength

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

6.3.12 CostPerEnergyUnit datatype

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

Table 17 shows all attributes of CostPerEnergyUnit.

Table 17 – Attributes of Domain::CostPerEnergyUnit

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

6.3.13 CostPerVolume datatype

Cost per unit volume.

Table 18 shows all attributes of CostPerVolume.

Table 18 – Attributes of Domain::CostPerVolume

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

6.3.14 CostRate datatype

Cost, in units of currency, per elapsed time.

Table 19 shows all attributes of CostRate.

Table 19 – Attributes of Domain::CostRate

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

6.3.15 Currency enumeration

Monetary currencies. Apologies for this list not being exhaustive.

Table 20 shows all literals of Currency.

Table 20 – Literals of Domain::Currency

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

6.3.16 CurrentFlow datatype

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

Table 21 shows all attributes of CurrentFlow.

Table 21 – Attributes of Domain::CurrentFlow

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

6.3.17 Damping datatype

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

Table 22 shows all attributes of Damping.

Table 22 – Attributes of Domain::Damping

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

6.3.18 Date primitive

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

6.3.19 DateInterval compound

Interval between two dates.

Table 23 shows all attributes of DateInterval.

Table 23 – Attributes of Domain::DateInterval

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

6.3.20 DateTime primitive

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

6.3.21 DateTimeInterval compound

Interval between two date and time points.

Table 24 shows all attributes of DateTimeInterval.

Table 24 – Attributes of Domain::DateTimeInterval

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

6.3.22 Decimal primitive

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

6.3.23 Displacement datatype

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

Table 25 shows all attributes of Displacement.

Table 25 – Attributes of Domain::Displacement

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

6.3.24 Duration primitive

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

6.3.25 Float primitive

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

6.3.26 FloatQuantity datatype

Quantity with float value and associated unit information.

Table 26 shows all attributes of FloatQuantity.

Table 26 – Attributes of Domain::FloatQuantity

name	type	description
value	Float	
unit	UnitSymbol	
multiplier	UnitMultiplier	

6.3.27 Frequency datatype

Cycles per second.

Table 27 shows all attributes of Frequency.

Table 27 – Attributes of Domain::Frequency

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

6.3.28 Hours datatype

Time specified in hours.

Table 28 shows all attributes of Hours.

Table 28 – Attributes of Domain::Hours

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

6.3.29 Impedance datatype

Ratio of voltage to current.

Table 29 shows all attributes of Impedance.

Table 29 – Attributes of Domain::Impedance

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

6.3.30 Inductance datatype

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

Table 30 shows all attributes of Inductance.

Table 30 – Attributes of Domain::Inductance

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

6.3.31 Integer primitive

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

6.3.32 IntegerQuantity datatype

Quantity with integer value and associated unit information.

Table 31 shows all attributes of IntegerQuantity.

Table 31 – Attributes of Domain::IntegerQuantity

name	type	description
value	Integer	
unit	UnitSymbol	
multiplier	UnitMultiplier	

6.3.33 KiloActivePower datatype

Active power in kilowatts.

Table 32 shows all attributes of KiloActivePower.

Table 32 – Attributes of Domain::KiloActivePower

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

6.3.34 Length datatype

Unit of length. Never negative.

Table 33 shows all attributes of Length.

Table 33 – Attributes of Domain::Length

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

6.3.35 Minutes datatype

Time in minutes.

Table 34 shows all attributes of Minutes.

Table 34 – Attributes of Domain::Minutes

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

6.3.36 Money datatype

Amount of money.

Table 35 shows all attributes of Money.

Table 35 – Attributes of Domain::Money

name	type	description
multiplier	UnitMultiplier	
unit	Currency	
value	Decimal	

6.3.37 PU datatype

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

Table 36 shows all attributes of PU.

Table 36 – Attributes of Domain::PU

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

6.3.38 PerCent datatype

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

Table 37 shows all attributes of PerCent.

Table 37 – Attributes of Domain::PerCent

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

6.3.39 Pressure datatype

Pressure in Pascal.

Table 38 shows all attributes of Pressure.

Table 38 – Attributes of Domain::Pressure

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

6.3.40 Reactance datatype

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

Table 39 shows all attributes of Reactance.

Table 39 – Attributes of Domain::Reactance

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

6.3.41 ReactancePerLength datatype

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

Table 40 shows all attributes of ReactancePerLength.

Table 40 – Attributes of Domain::ReactancePerLength

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

6.3.42 ReactivePower datatype

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

Table 41 shows all attributes of ReactivePower.

Table 41 – Attributes of Domain::ReactivePower

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

6.3.43 RealEnergy datatype

Real electrical energy.

Table 42 shows all attributes of RealEnergy.

Table 42 – Attributes of Domain::RealEnergy

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

6.3.44 Resistance datatype

Resistance (real part of impedance).

Table 43 shows all attributes of Resistance.

Table 43 – Attributes of Domain::Resistance

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

6.3.45 ResistancePerLength datatype

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

Table 44 shows all attributes of ResistancePerLength.

Table 44 – Attributes of Domain::ResistancePerLength

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

6.3.46 RotationSpeed datatype

Number of revolutions per second.

Table 45 shows all attributes of RotationSpeed.

Table 45 – Attributes of Domain::RotationSpeed

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

6.3.47 Seconds datatype

Time, in seconds.

Table 46 shows all attributes of Seconds.

Table 46 – Attributes of Domain::Seconds

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

6.3.48 Speed datatype

Distance per unit of time.

Table 47 shows all attributes of Speed.

Table 47 – Attributes of Domain::Speed

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

6.3.49 String primitive

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

6.3.50 StringQuantity datatype

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

Table 48 shows all attributes of StringQuantity.

Table 48 – Attributes of Domain::StringQuantity

name	type	description
value	String	
unit	UnitSymbol	
multiplier	UnitMultiplier	

6.3.51 Susceptance datatype

Imaginary part of admittance.

Table 49 shows all attributes of Susceptance.

Table 49 – Attributes of Domain::Susceptance

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

6.3.52 SusceptancePerLength datatype

Imaginary part of admittance per unit of length.

Table 50 shows all attributes of SusceptancePerLength.

Table 50 – Attributes of Domain::SusceptancePerLength

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

6.3.53 Temperature datatype

Value of temperature in degrees Celsius.

Table 51 shows all attributes of Temperature.

Table 51 – Attributes of Domain::Temperature

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

6.3.54 Time primitive

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

6.3.55 TimeInterval compound

Interval between two times.

Table 52 shows all attributes of TimeInterval.

Table 52 – Attributes of Domain::TimeInterval

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

6.3.56 UnitMultiplier enumeration

The unit multipliers defined for the CIM.

Table 53 shows all literals of UnitMultiplier.

Table 53 – Literals of Domain::UnitMultiplier

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

6.3.57 UnitSymbol enumeration

The units defiend for usage in the CIM.

Table 54 shows all literals of UnitSymbol.

Table 54 – Literals of Domain::UnitSymbol

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

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

6.3.58 Voltage datatype

Electrical voltage.

Table 55 shows all attributes of Voltage.

Table 55 – Attributes of Domain::Voltage

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

6.3.59 VoltagePerReactivePower datatype

Voltage variation with reactive power.

Table 56 shows all attributes of VoltagePerReactivePower.

Table 56 – Attributes of Domain::VoltagePerReactivePower

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

6.3.60 Volume datatype

Volume.

Table 57 shows all attributes of Volume.

Table 57 – Attributes of Domain::Volume

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

6.3.61 VolumeFlowRate datatype

Volume per time.

Table 58 shows all attributes of VolumeFlowRate.

Table 58 – Attributes of Domain::VolumeFlowRate

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

6.3.62 WaterLevel datatype

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

Table 59 shows all attributes of WaterLevel.

Table 59 – Attributes of Domain::WaterLevel

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

6.3.63 Weight datatype

The weight of an object.

Table 60 shows all attributes of Weight.

Table 60 – Attributes of Domain::Weight

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

6.4 Package Core

6.4.1 General

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

Figure 33 shows class diagram Main.

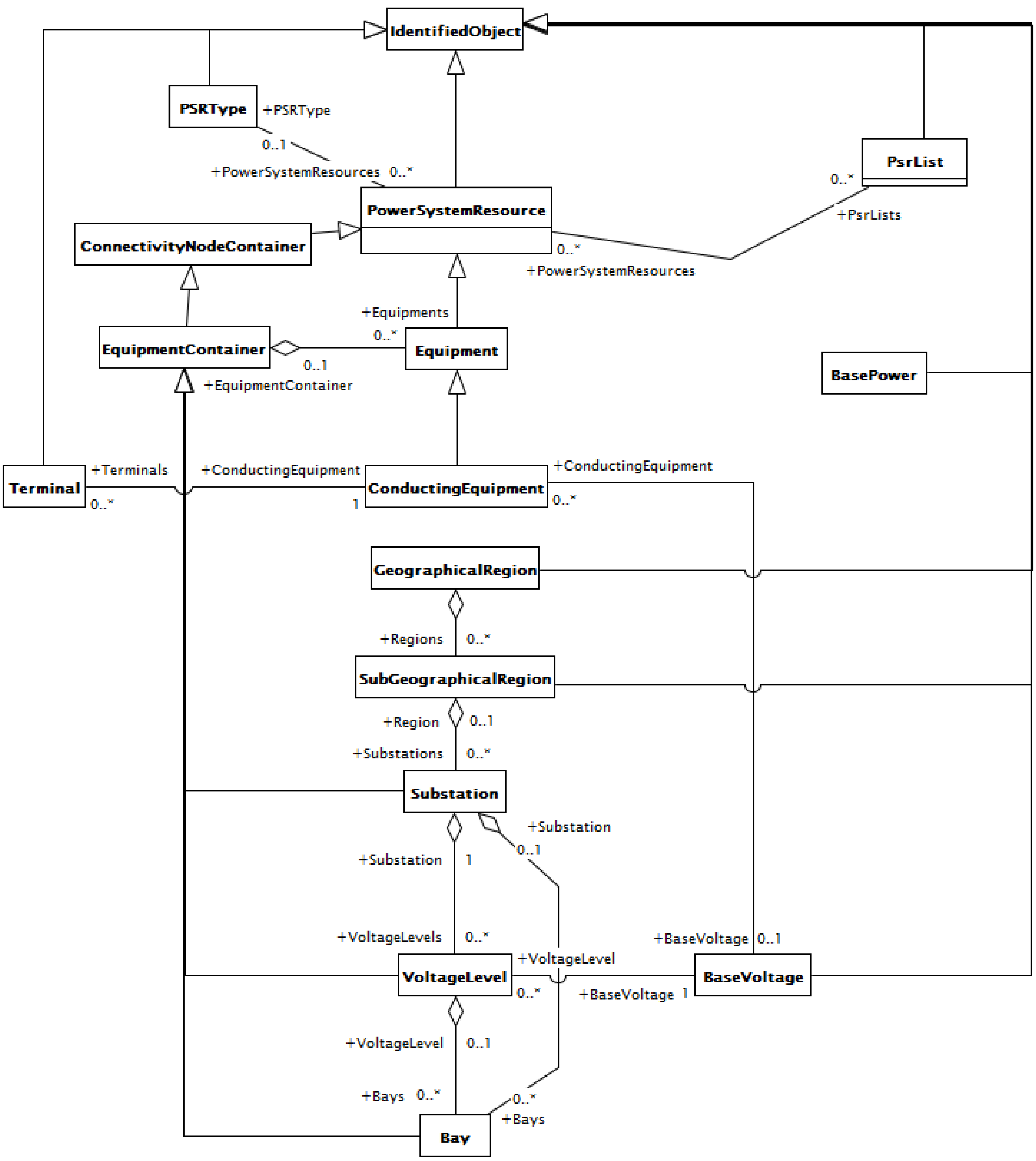

Figure 33 – Class diagram Core::Main

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

Figure 34 shows class diagram Names.

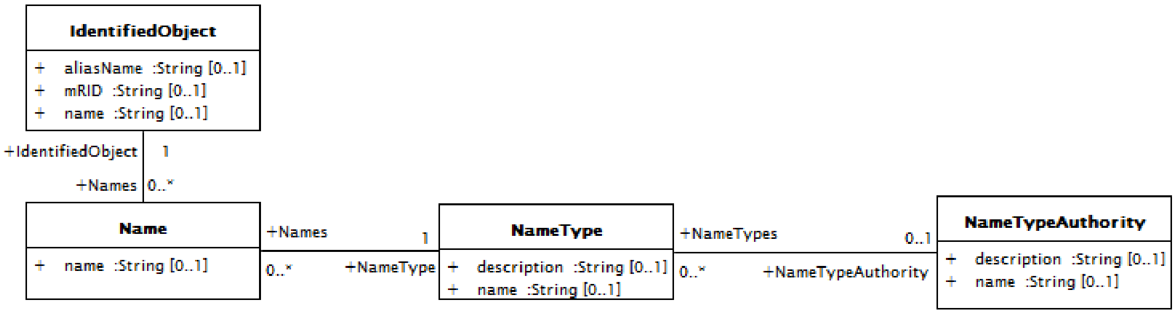

Figure 34 – Class diagram Core::Names

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

Figure 35 shows class diagram CurveSchedule.

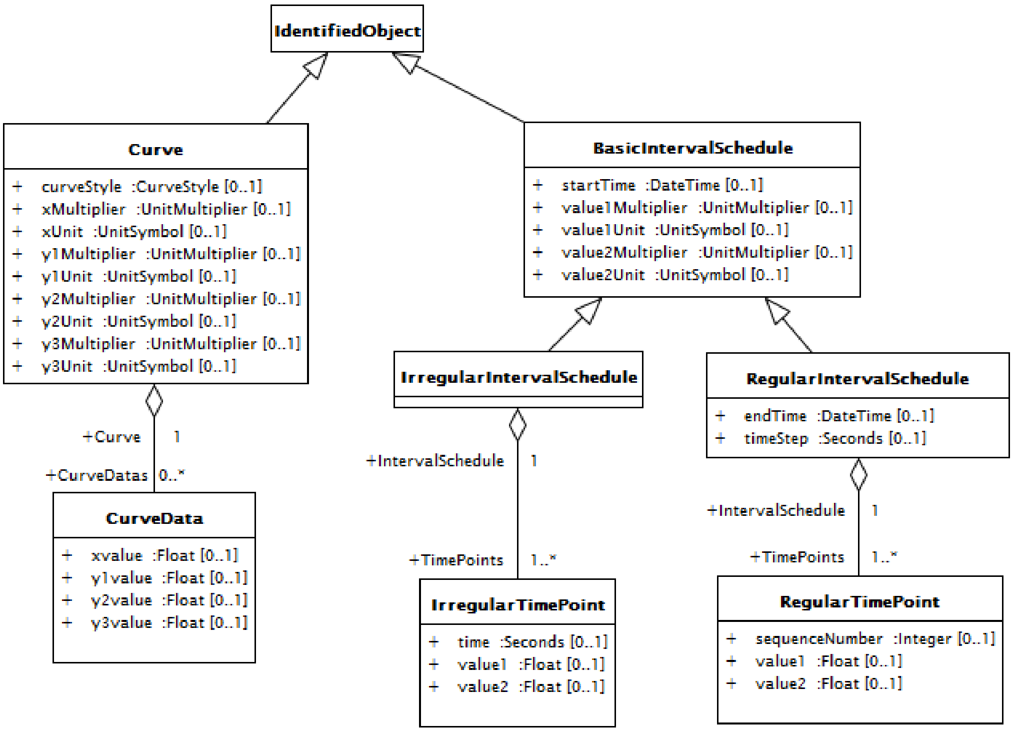

Figure 35 – Class diagram Core::CurveSchedule

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

Figure 36 shows class diagram Datatypes.

Figure 36 – Class diagram Core::Datatypes

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

Figure 37 shows class diagram Reporting.

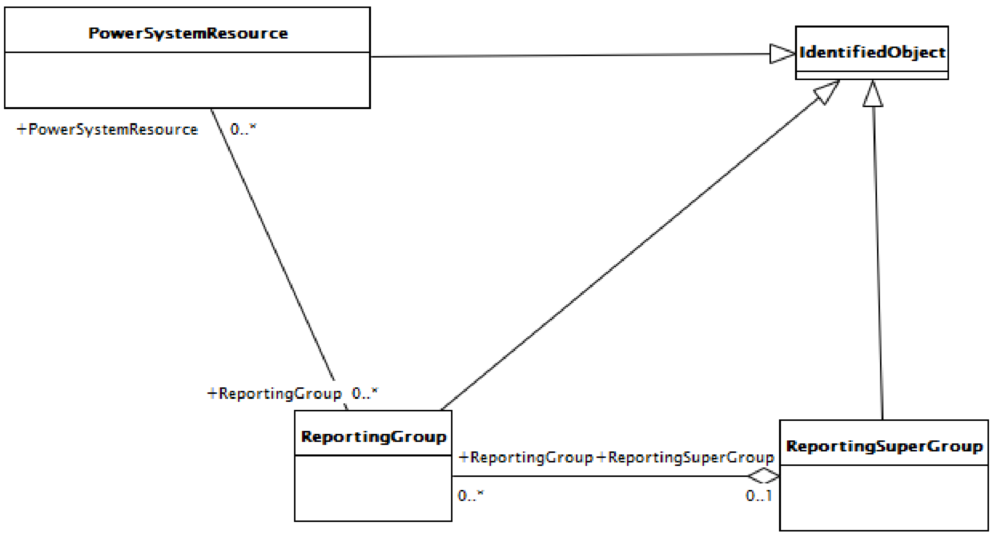

Figure 37 – Class diagram Core::Reporting

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

Figure 38 shows class diagram OperatingShare.

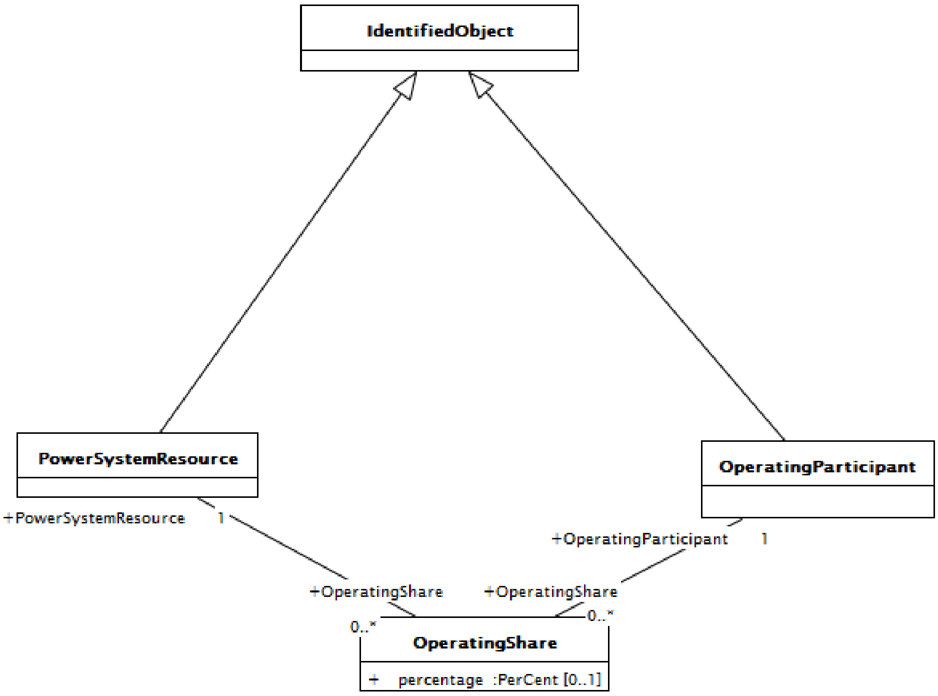

Figure 38 – Class diagram Core::OperatingShare

This diagram shows the common model for operating share.

6.4.2 BasePower

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

Table 61 shows all attributes of BasePower.

Table 61 – Attributes of Core::BasePower

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

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

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

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

6.4.3 BaseVoltage

Defines a system base voltage which is referenced.

Table 63 shows all attributes of BaseVoltage.

Table 63 – Attributes of Core::BaseVoltage

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

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

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

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

6.4.4 BasicIntervalSchedule

Schedule of values at points in time.

Table 65 shows all attributes of BasicIntervalSchedule.

Table 65 – Attributes of Core::BasicIntervalSchedule

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

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

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

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

6.4.5 Bay

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

Table 67 shows all attributes of Bay.

Table 67 – Attributes of Core::Bay

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

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

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

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

6.4.6 BreakerConfiguration enumeration

Switching arrangement for bay.

Table 69 shows all literals of BreakerConfiguration.

Table 69 – Literals of Core::BreakerConfiguration

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

6.4.7 BusbarConfiguration enumeration

Busbar layout for bay.

Table 70 shows all literals of BusbarConfiguration.

Table 70 – Literals of Core::BusbarConfiguration

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

6.4.8 ConductingEquipment

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

Table 71 shows all attributes of ConductingEquipment.

Table 71 – Attributes of Core::ConductingEquipment

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

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

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

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

6.4.9 ConnectivityNode

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

Table 73 shows all attributes of ConnectivityNode.

Table 73 – Attributes of Core::ConnectivityNode

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

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

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

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

6.4.10 ConnectivityNodeContainer

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

Table 75 shows all attributes of ConnectivityNodeContainer.

Table 75 – Attributes of Core::ConnectivityNodeContainer

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

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

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

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

6.4.11 Curve

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

Table 77 shows all attributes of Curve.

Table 77 – Attributes of Core::Curve

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

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

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

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

6.4.12 CurveData root class

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

Table 79 shows all attributes of CurveData.

Table 79 – Attributes of Core::CurveData

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

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

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

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

6.4.13 CurveStyle enumeration

Style or shape of curve.

Table 81 shows all literals of CurveStyle.

Table 81 – Literals of Core::CurveStyle

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

6.4.14 Equipment

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

Table 82 shows all attributes of Equipment.

Table 82 – Attributes of Core::Equipment

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

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

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

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

6.4.15 EquipmentContainer

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

Table 84 shows all attributes of EquipmentContainer.

Table 84 – Attributes of Core::EquipmentContainer

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

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

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

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

6.4.16 GeographicalRegion

A geographical region of a power system network model.

Table 86 shows all attributes of GeographicalRegion.

Table 86 – Attributes of Core::GeographicalRegion

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

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

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

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

6.4.17 IdentifiedObject root class

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

Table 88 shows all attributes of IdentifiedObject.

Table 88 – Attributes of Core::IdentifiedObject

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

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

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

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

6.4.18 IrregularIntervalSchedule

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

Table 90 shows all attributes of IrregularIntervalSchedule.

Table 90 – Attributes of Core::IrregularIntervalSchedule

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

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

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

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

6.4.19 IrregularTimePoint root class

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

Table 92 shows all attributes of IrregularTimePoint.

Table 92 – Attributes of Core::IrregularTimePoint

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

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

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

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

6.4.20 Name root class

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

Table 94 shows all attributes of Name.

Table 94 – Attributes of Core::Name

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

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

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

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

6.4.21 NameType root class

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

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

Table 96 shows all attributes of NameType.

Table 96 – Attributes of Core::NameType

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

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

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

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

6.4.22 NameTypeAuthority root class

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

Table 98 shows all attributes of NameTypeAuthority.

Table 98 – Attributes of Core::NameTypeAuthority

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

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

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

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

6.4.23 OperatingParticipant

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

Table 100 shows all attributes of OperatingParticipant.

Table 100 – Attributes of Core::OperatingParticipant

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

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

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

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

6.4.24 OperatingShare root class

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

Table 102 shows all attributes of OperatingShare.

Table 102 – Attributes of Core::OperatingShare

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

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

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

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

6.4.25 PhaseCode enumeration

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

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

Table 104 shows all literals of PhaseCode.

Table 104 – Literals of Core::PhaseCode

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

6.4.26 PowerSystemResource

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

Table 105 shows all attributes of PowerSystemResource.

Table 105 – Attributes of Core::PowerSystemResource

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

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

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

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

6.4.27 PsrList

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

Table 107 shows all attributes of PsrList.

Table 107 – Attributes of Core::PsrList

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

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

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

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

6.4.28 PSRType

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

Table 109 shows all attributes of PSRType.

Table 109 – Attributes of Core::PSRType

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

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

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

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

6.4.29 RegularIntervalSchedule

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

Table 111 shows all attributes of RegularIntervalSchedule.

Table 111 – Attributes of Core::RegularIntervalSchedule

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

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

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

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

6.4.30 RegularTimePoint root class

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

Table 113 shows all attributes of RegularTimePoint.

Table 113 – Attributes of Core::RegularTimePoint

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

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

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

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

6.4.31 ReportingGroup

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

Table 115 shows all attributes of ReportingGroup.

Table 115 – Attributes of Core::ReportingGroup

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

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

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

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

6.4.32 ReportingSuperGroup

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

Table 117 shows all attributes of ReportingSuperGroup.

Table 117 – Attributes of Core::ReportingSuperGroup

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

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

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

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

6.4.33 SubGeographicalRegion

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

Table 119 shows all attributes of SubGeographicalRegion.

Table 119 – Attributes of Core::SubGeographicalRegion

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

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

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

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

6.4.34 Substation

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

Table 121 shows all attributes of Substation.

Table 121 – Attributes of Core::Substation

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

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

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

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

6.4.35 Terminal

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

Table 123 shows all attributes of Terminal.

Table 123 – Attributes of Core::Terminal

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

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

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

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

6.4.36 VoltageLevel

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

Table 125 shows all attributes of VoltageLevel.

Table 125 – Attributes of Core::VoltageLevel

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

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

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

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

6.5 Package DiagramLayout

6.5.1 General

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

Figure 39 shows class diagram DiagramLayout.

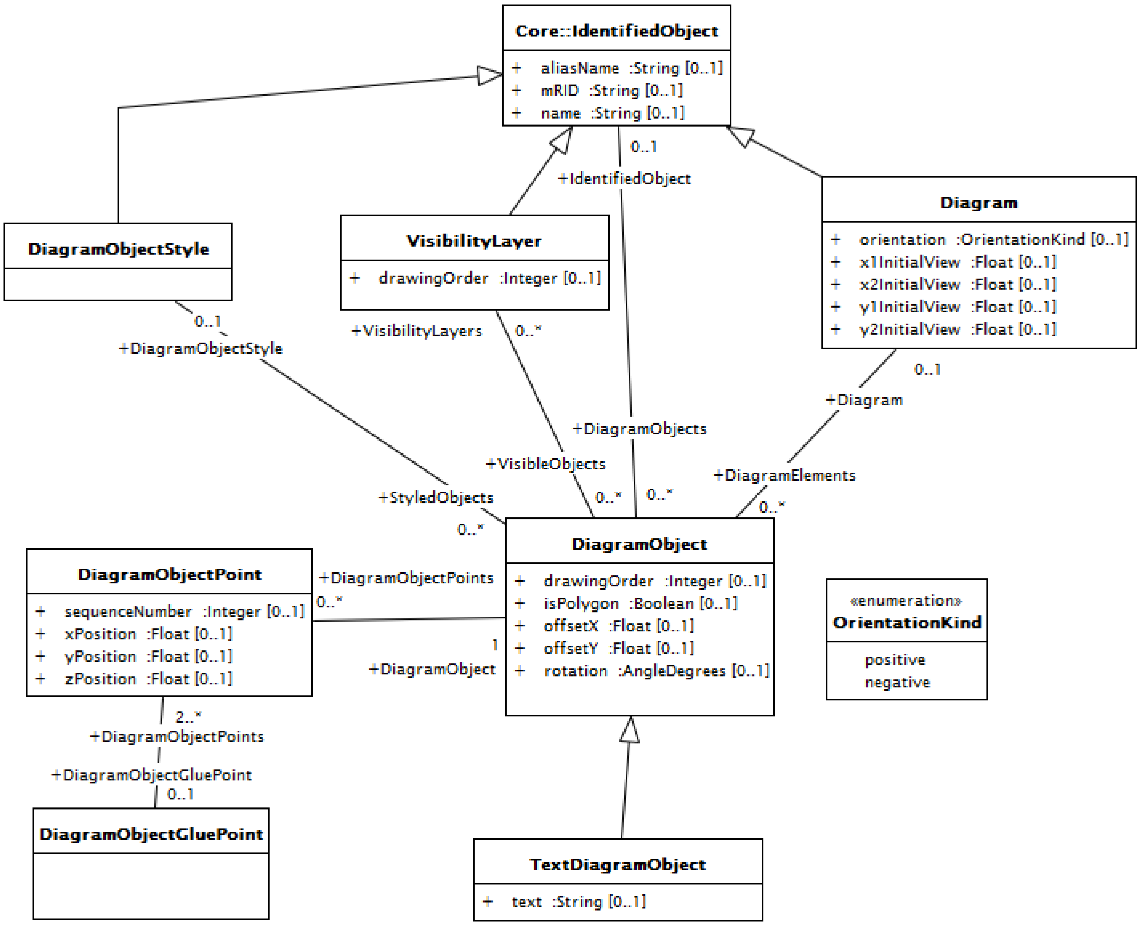

Figure 39 – Class diagram DiagramLayout::DiagramLayout

The diagram shows the diagram layout model.

6.5.2 Diagram

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

Table 127 shows all attributes of Diagram.

Table 127 – Attributes of DiagramLayout::Diagram

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

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

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

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

6.5.3 DiagramObject

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

Table 129 shows all attributes of DiagramObject.

Table 129 – Attributes of DiagramLayout::DiagramObject

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

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

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

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

6.5.4 DiagramObjectGluePoint root class

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

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

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

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

6.5.5 DiagramObjectPoint root class

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

Table 132 shows all attributes of DiagramObjectPoint.

Table 132 – Attributes of DiagramLayout::DiagramObjectPoint

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

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

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

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

6.5.6 DiagramObjectStyle

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

Table 134 shows all attributes of DiagramObjectStyle.

Table 134 – Attributes of DiagramLayout::DiagramObjectStyle

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

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

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

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

6.5.7 OrientationKind enumeration

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

Table 136 shows all literals of OrientationKind.

Table 136 – Literals of DiagramLayout::OrientationKind

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

6.5.8 TextDiagramObject

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

Table 137 shows all attributes of TextDiagramObject.

Table 137 – Attributes of DiagramLayout::TextDiagramObject

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

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

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

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

6.5.9 VisibilityLayer

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

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

Table 139 shows all attributes of VisibilityLayer.

Table 139 – Attributes of DiagramLayout::VisibilityLayer

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

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

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

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

6.6 Package OperationalLimits

6.6.1 General

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

Figure 40 shows class diagram OperationalLimits.

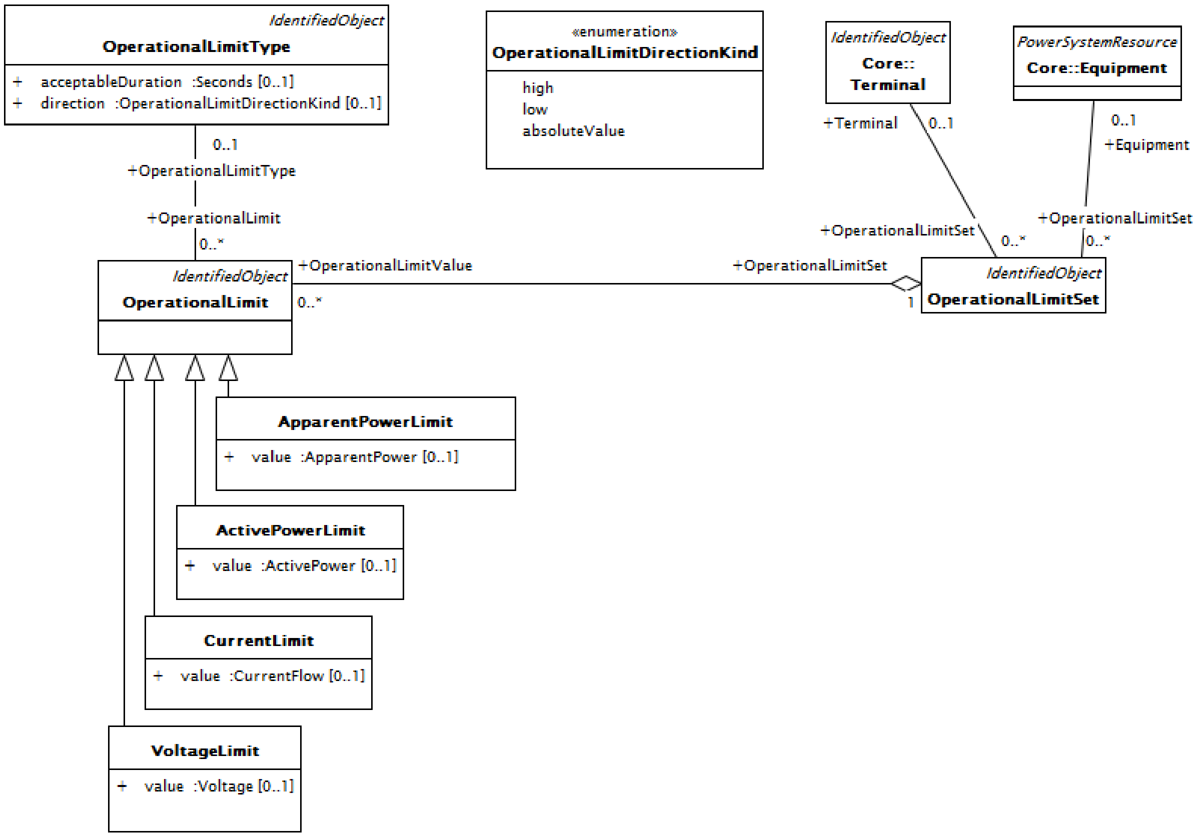

Figure 40 – Class diagram OperationalLimits::OperationalLimits

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

Figure 41 shows class diagram BranchGroup.

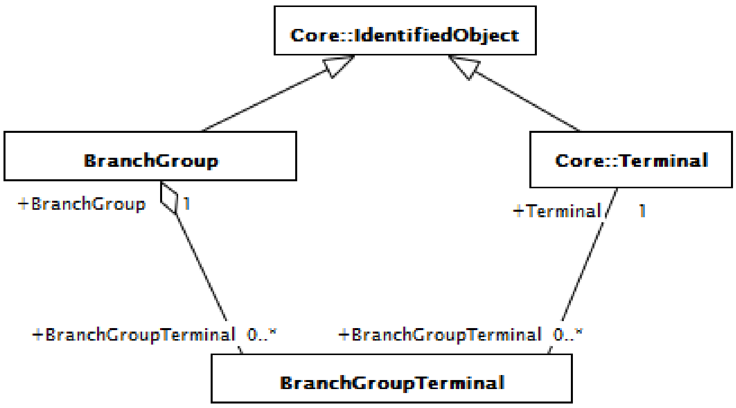

Figure 41 – Class diagram OperationalLimits::BranchGroup

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

6.6.2 ActivePowerLimit

Limit on active power flow.

Table 141 shows all attributes of ActivePowerLimit.

Table 141 – Attributes of OperationalLimits::ActivePowerLimit

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

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

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

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

6.6.3 ApparentPowerLimit

Apparent power limit.

Table 143 shows all attributes of ApparentPowerLimit.

Table 143 – Attributes of OperationalLimits::ApparentPowerLimit

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

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

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

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

6.6.4 BranchGroup

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

Table 145 shows all attributes of BranchGroup.

Table 145 – Attributes of OperationalLimits::BranchGroup

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

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

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

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

6.6.5 BranchGroupTerminal root class

A specific directed terminal flow for a branch group.

Table 147 shows all attributes of BranchGroupTerminal.

Table 147 – Attributes of OperationalLimits::BranchGroupTerminal

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

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

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

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

6.6.6 CurrentLimit

Operational limit on current.

Table 149 shows all attributes of CurrentLimit.

Table 149 – Attributes of OperationalLimits::CurrentLimit

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

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

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

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

6.6.7 OperationalLimit

A value associated with a specific kind of limit.

Table 151 shows all attributes of OperationalLimit.

Table 151 – Attributes of OperationalLimits::OperationalLimit

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

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

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

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

6.6.8 OperationalLimitDirectionKind enumeration

The direction of an operational limit.

Table 153 shows all literals of OperationalLimitDirectionKind.

Table 153 – Literals of OperationalLimits::OperationalLimitDirectionKind

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

6.6.9 OperationalLimitSet

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

Table 154 shows all attributes of OperationalLimitSet.

Table 154 – Attributes of OperationalLimits::OperationalLimitSet

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

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

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

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

6.6.10 OperationalLimitType

The operational meaning of a category of limits.

Table 156 shows all attributes of OperationalLimitType.

Table 156 – Attributes of OperationalLimits::OperationalLimitType

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

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

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

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

6.6.11 VoltageLimit

Operational limit applied to voltage.

Table 158 shows all attributes of VoltageLimit.

Table 158 – Attributes of OperationalLimits::VoltageLimit

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

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

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

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

6.7 Package Topology

6.7.1 General

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

Figure 42 shows class diagram TopologicalNodeTerminal.

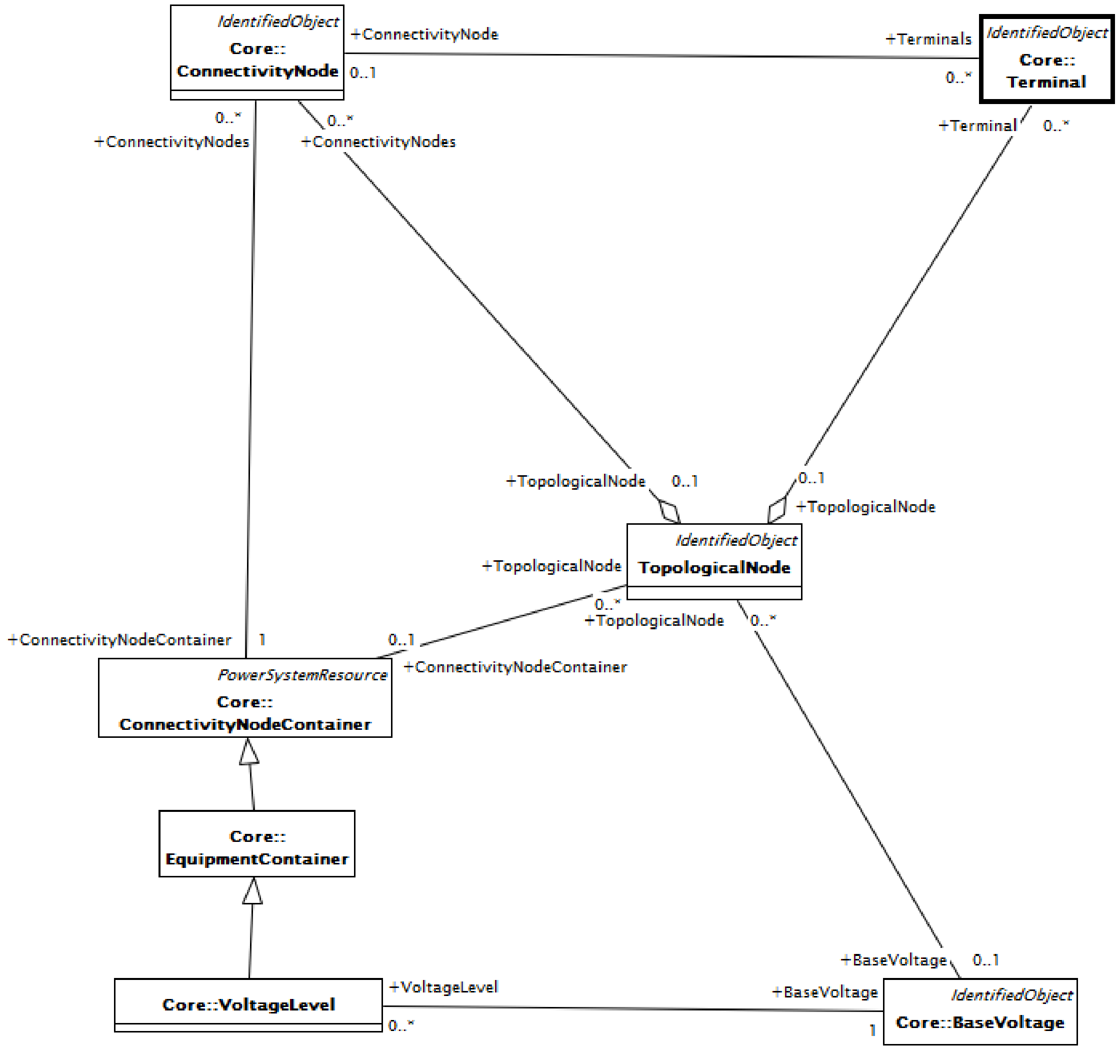

Figure 42 – Class diagram Topology::TopologicalNodeTerminal

Shows the relationships among topological nodes and terminals.

Figure 43 shows class diagram TopologyMeasRelations.

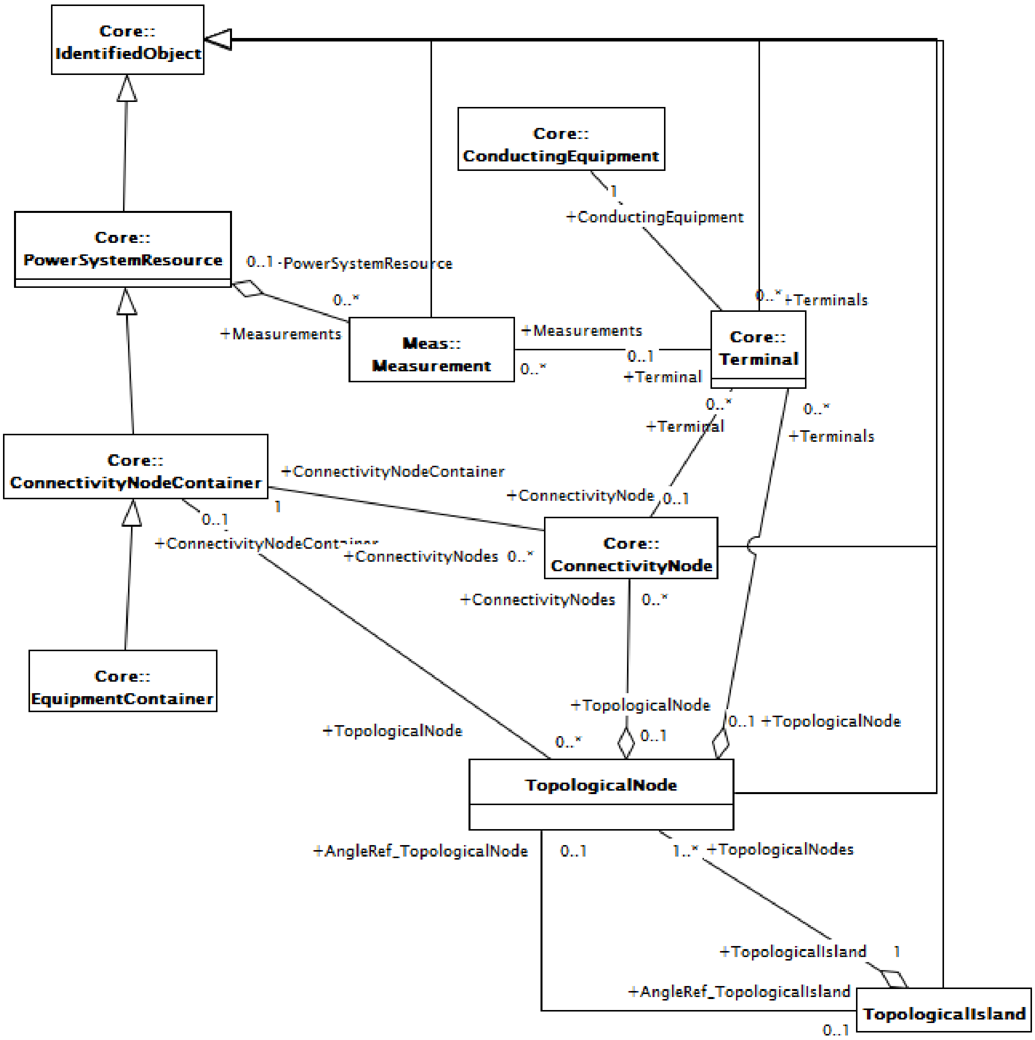

Figure 43 – Class diagram Topology::TopologyMeasRelations

Shows the relationship of measurements to terminals and topology.

Figure 44 shows class diagram TopologyReporting.

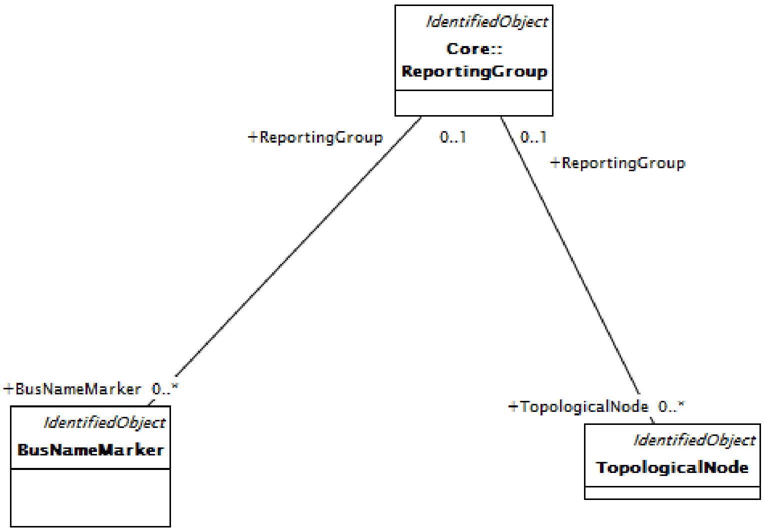

Figure 44 – Class diagram Topology::TopologyReporting

Shows topology reporting relationships.

Figure 45 shows class diagram Main.

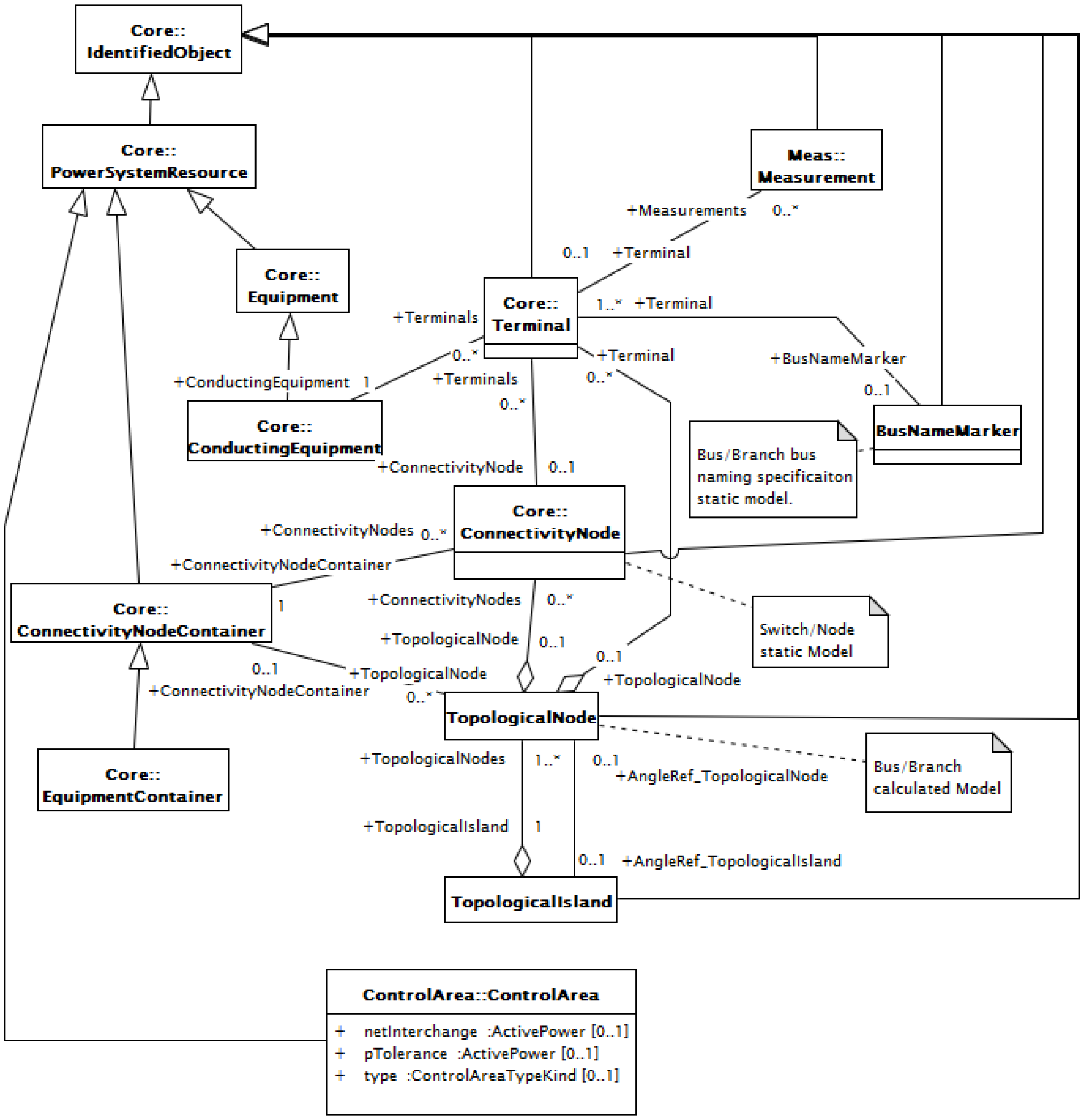

Figure 45 – Class diagram Topology::Main

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

6.7.2 BusNameMarker

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

Table 160 shows all attributes of BusNameMarker.

Table 160 – Attributes of Topology::BusNameMarker

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

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

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

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

6.7.3 TopologicalIsland

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

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

Table 162 shows all attributes of TopologicalIsland.

Table 162 – Attributes of Topology::TopologicalIsland

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

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

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

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

6.7.4 TopologicalNode

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

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

Table 164 shows all attributes of TopologicalNode.

Table 164 – Attributes of Topology::TopologicalNode

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

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

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

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

6.8 Package Wires

6.8.1 General

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

Figure 46 shows class diagram CutsAndJumpers.

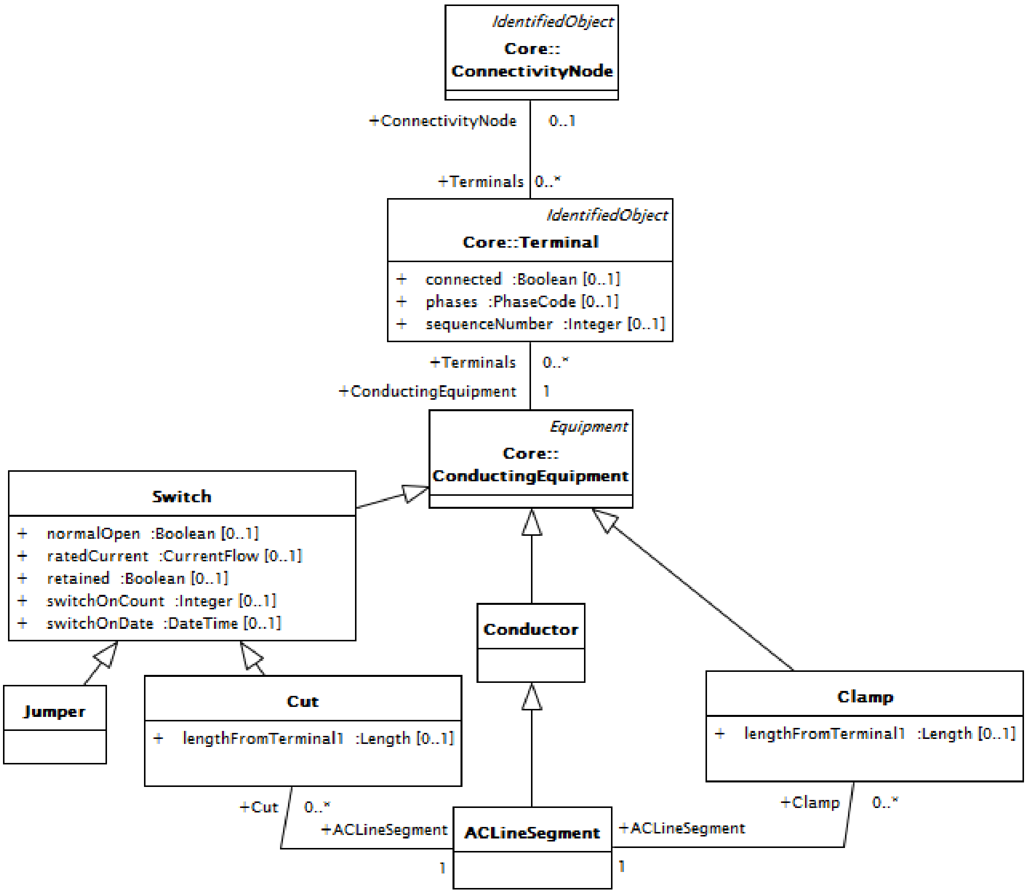

Figure 46 – Class diagram Wires::CutsAndJumpers

The diagram show the cuts and jumpers data model.

Figure 47 shows class diagram MutualCoupling.

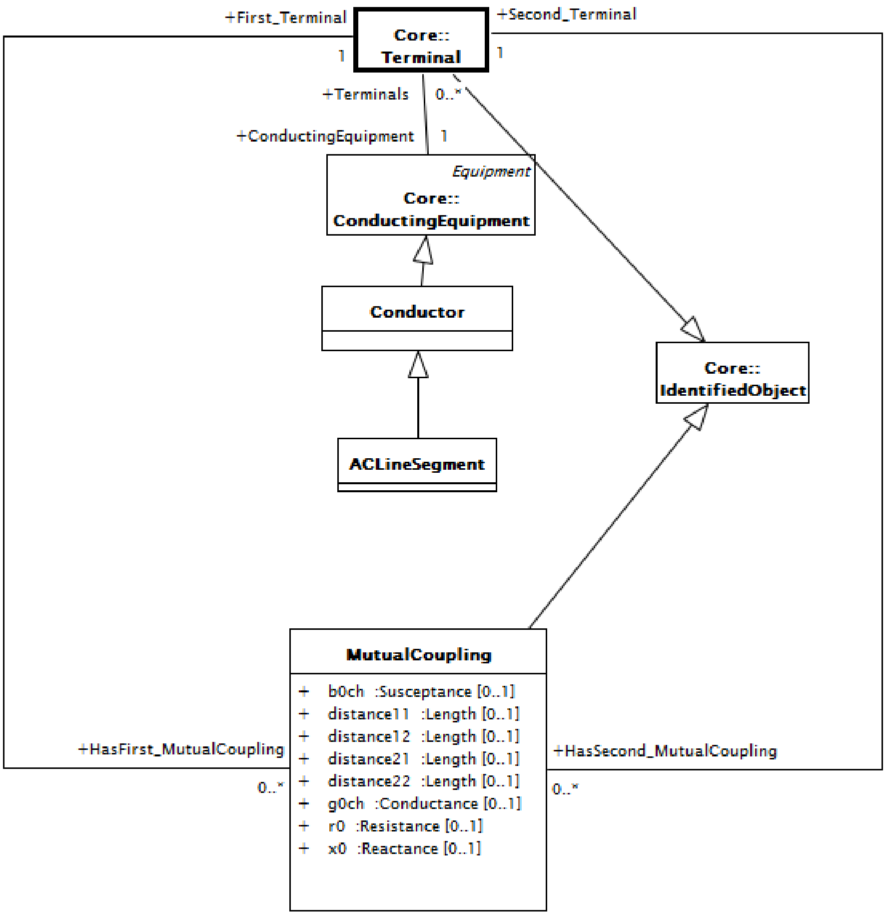

Figure 47 – Class diagram Wires::MutualCoupling

Shows mutual coupling models.

Figure 48 shows class diagram Schedules.

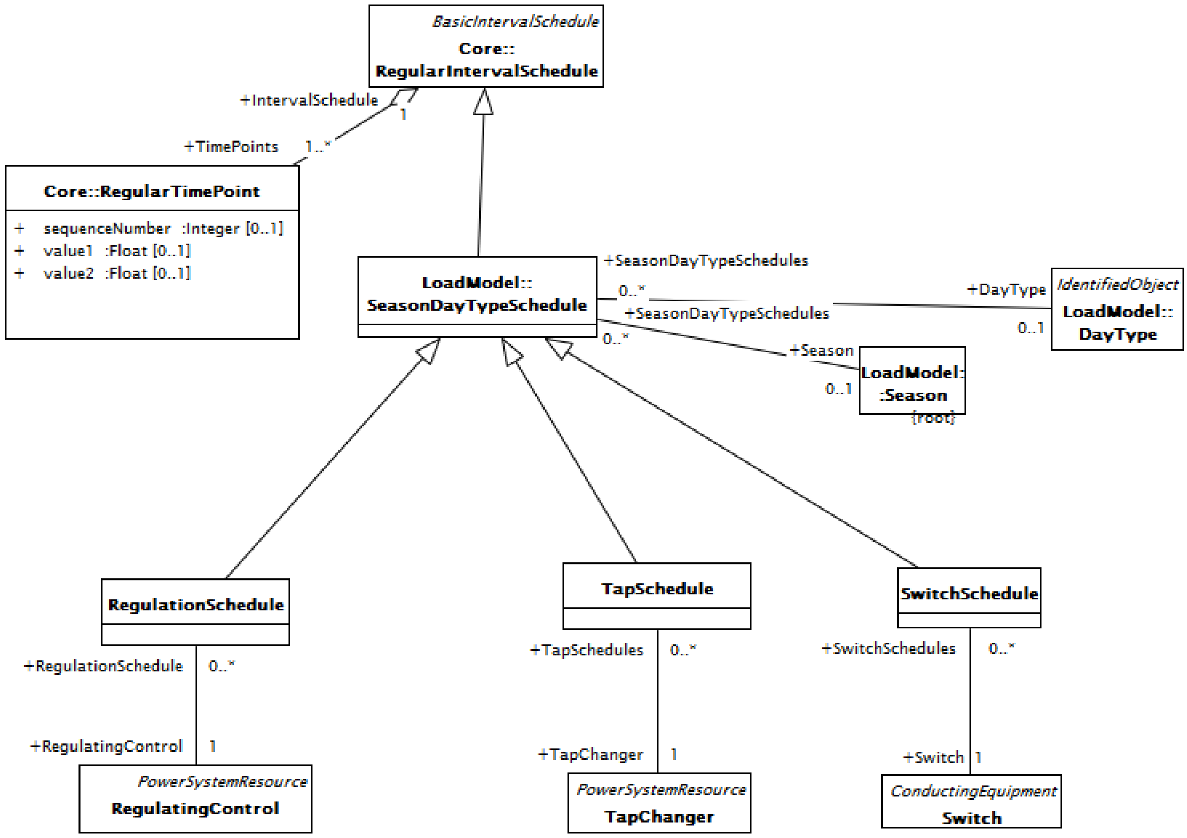

Figure 48 – Class diagram Wires::Schedules

Shows schedules of wires model.

Figure 49 shows class diagram SwitchingEquipment.

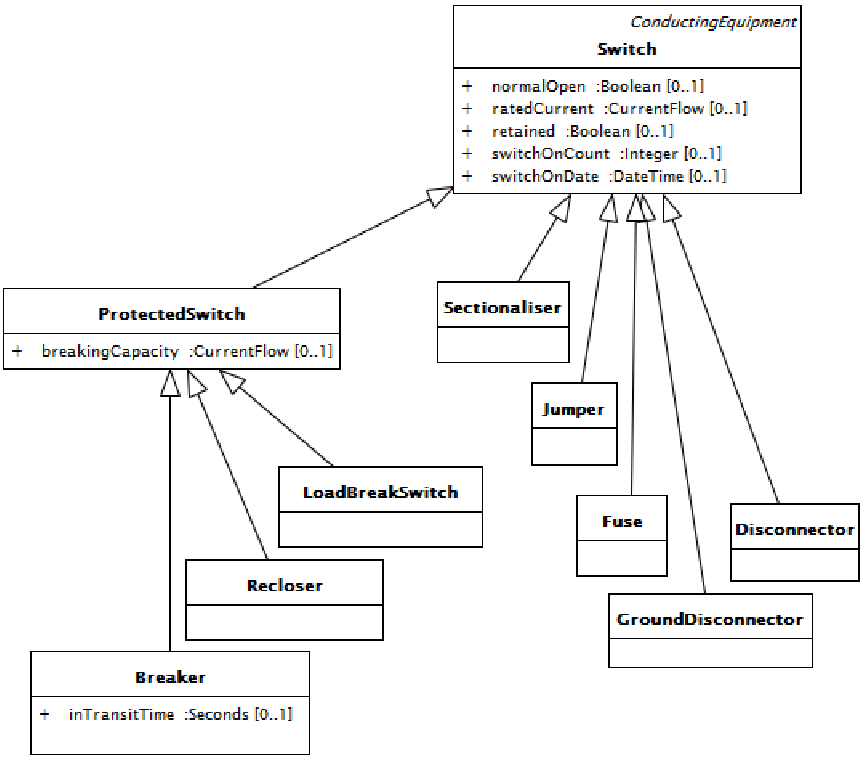

Figure 49 – Class diagram Wires::SwitchingEquipment

Shows switching equipment inheritance structure.

Figure 50 shows class diagram WiresPhaseModel.

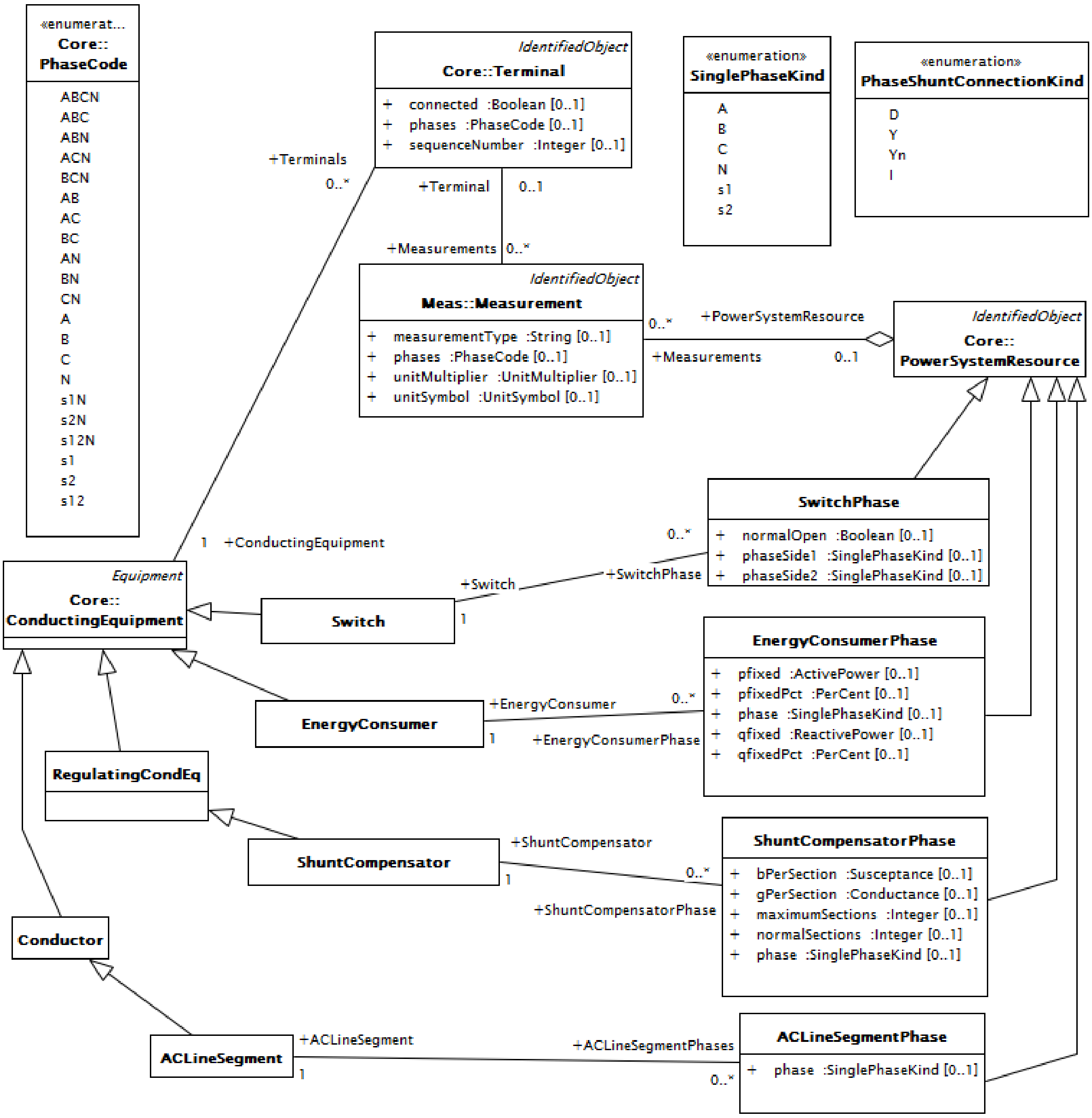

Figure 50 – Class diagram Wires::WiresPhaseModel

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

Figure 51 shows class diagram Datatypes.

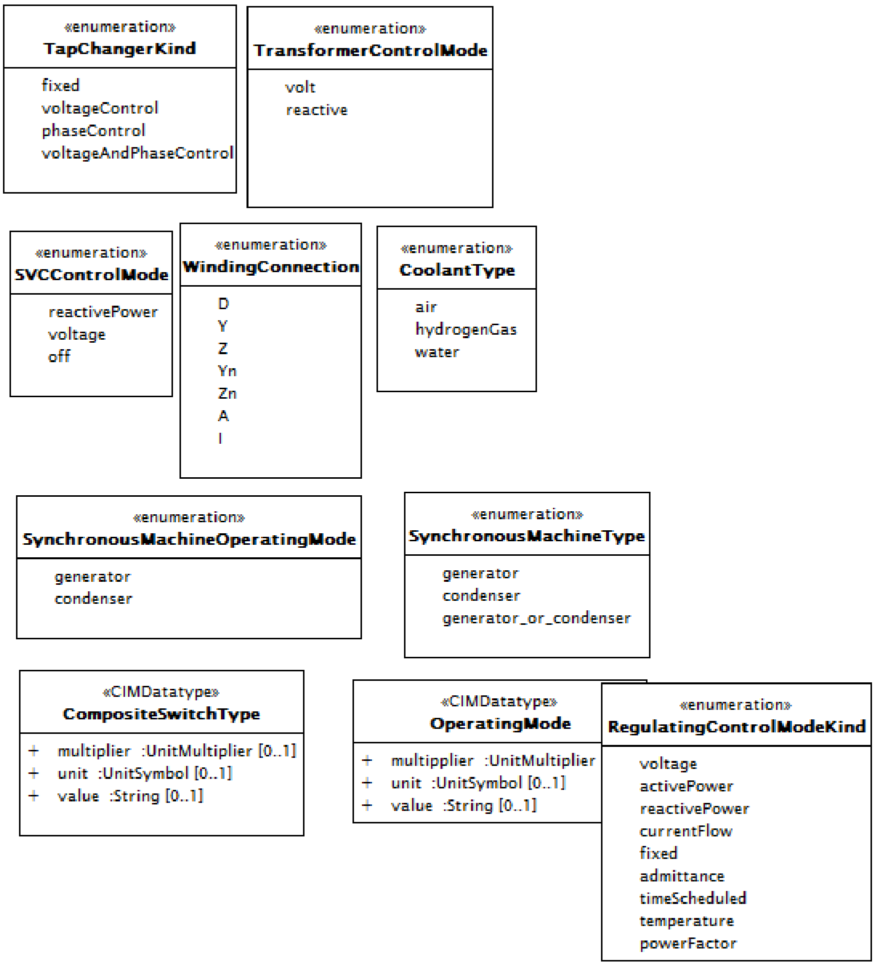

Figure 51 – Class diagram Wires::Datatypes

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

Figure 52 shows class diagram InheritanceHierarchy.

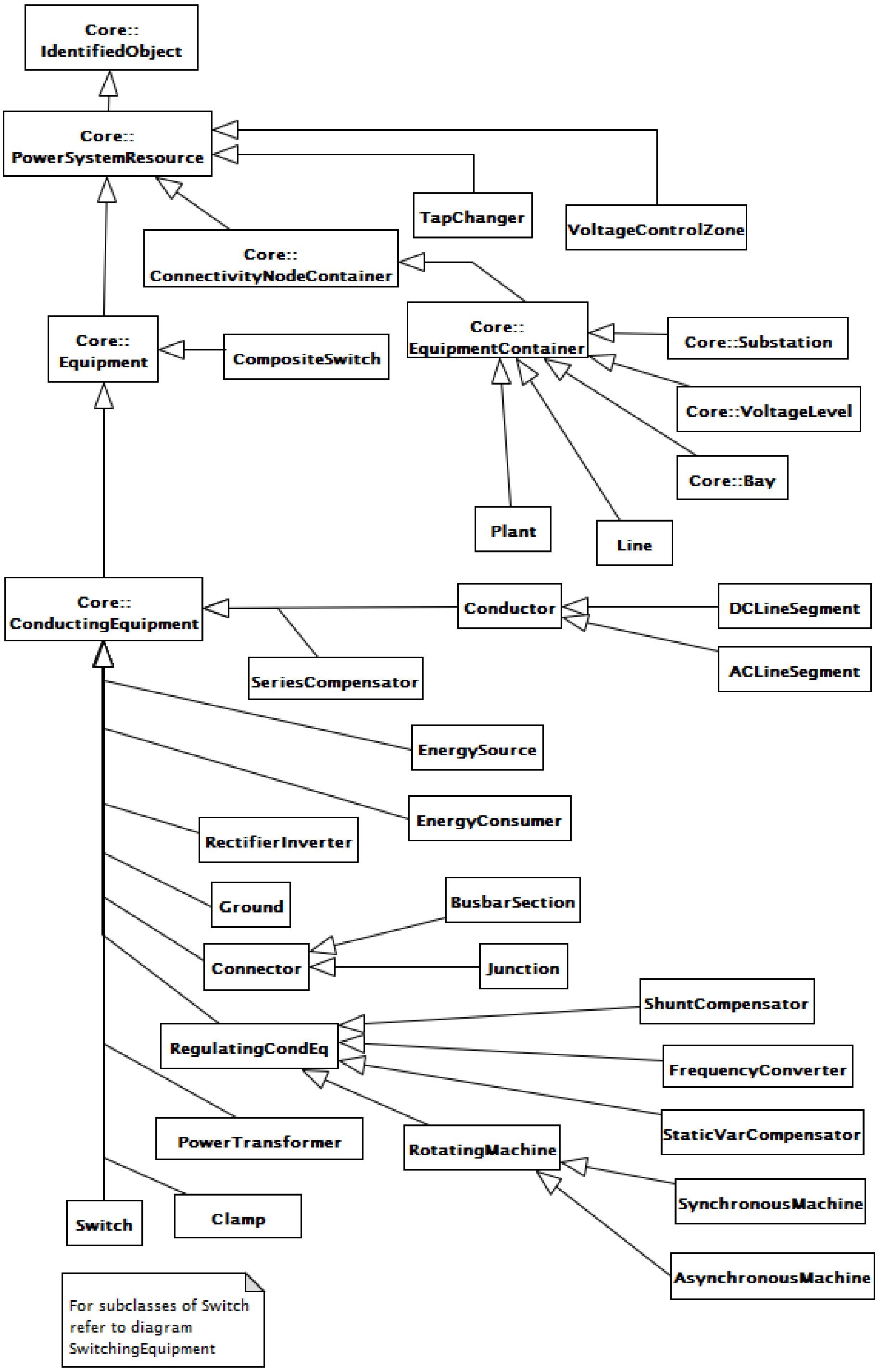

Figure 52 – Class diagram Wires::InheritanceHierarchy

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

Figure 53 shows class diagram LineModel.

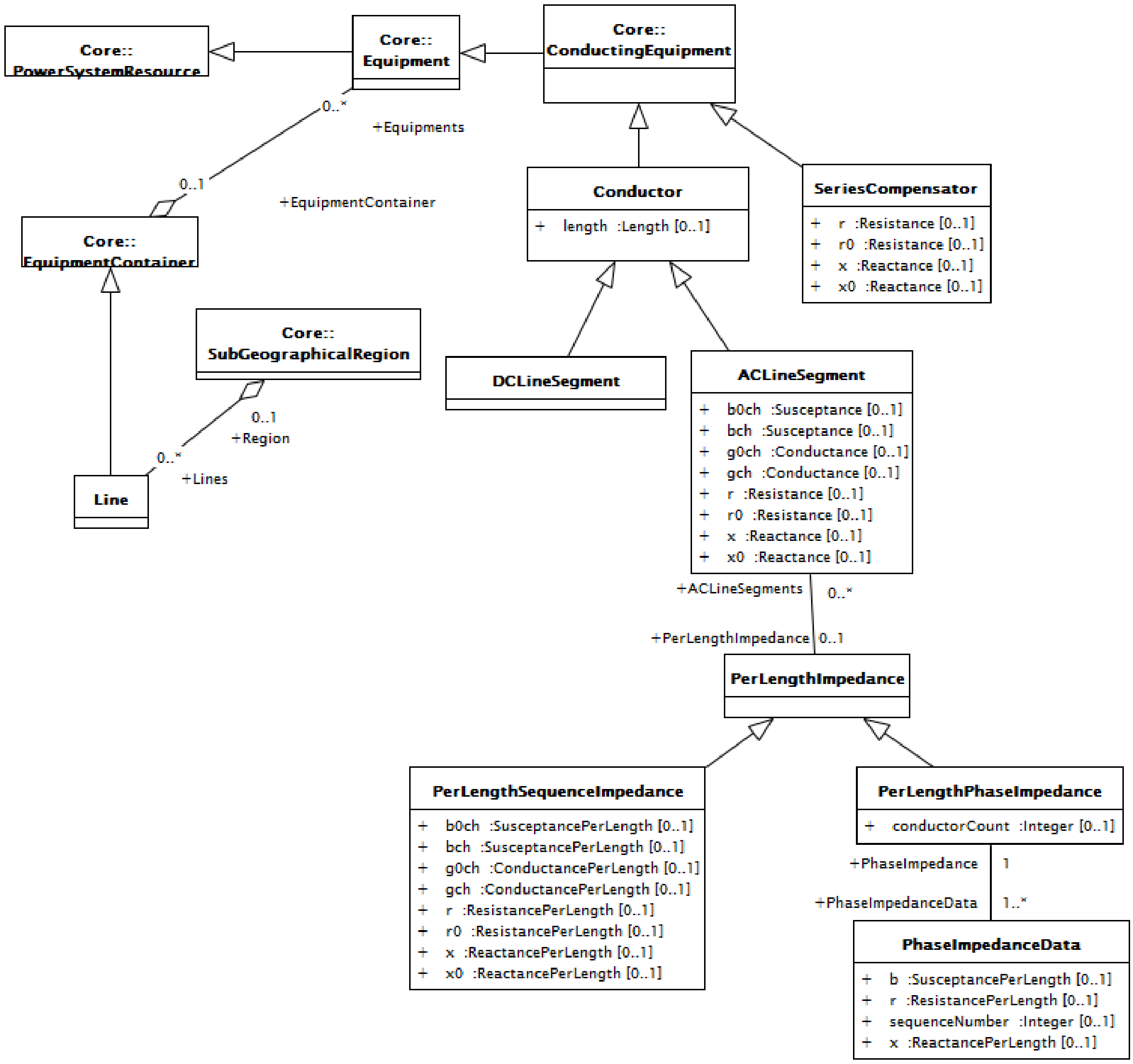

Figure 53 – Class diagram Wires::LineModel

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

Figure 54 shows class diagram NamingHierarchyPart1.

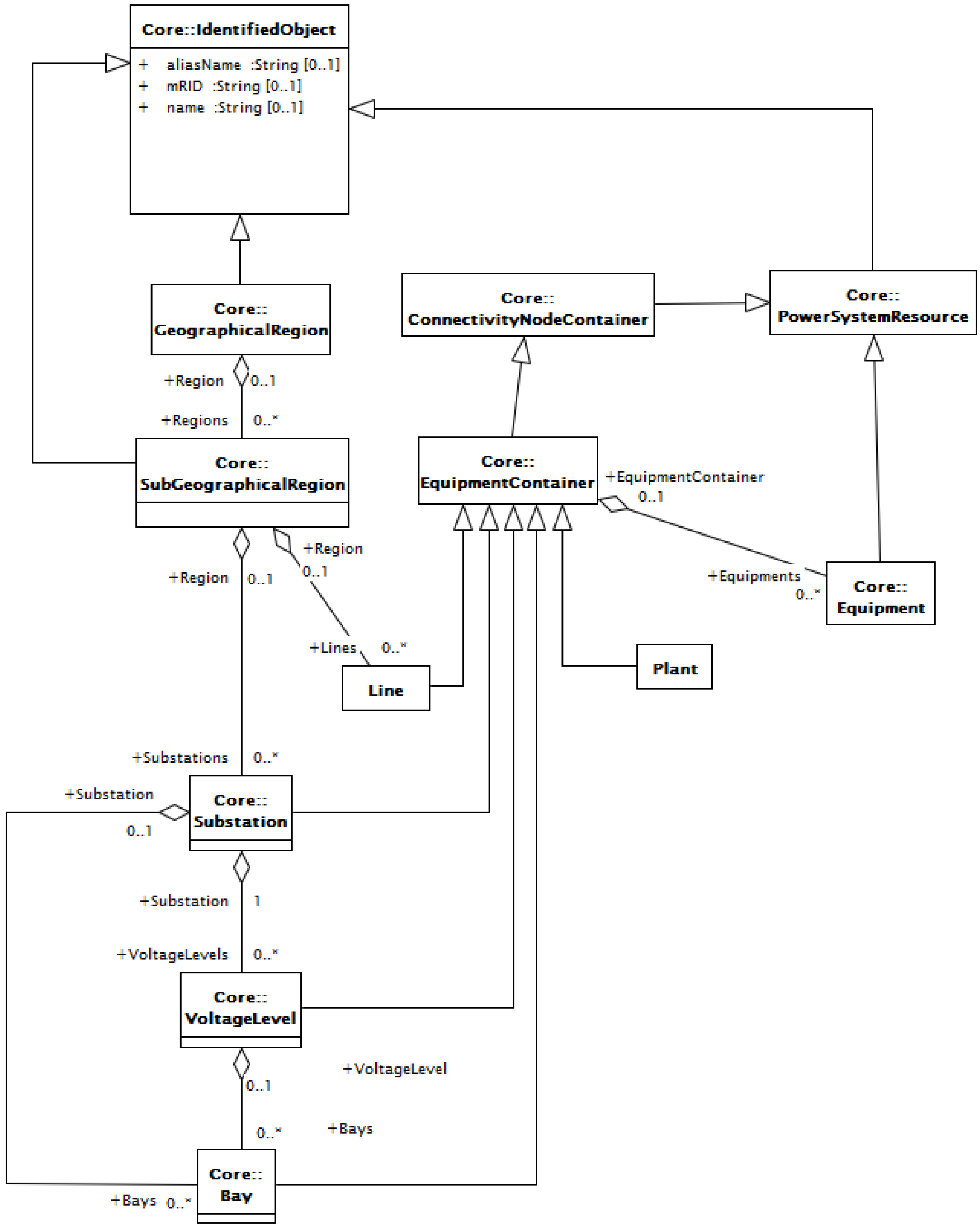

Figure 54 – Class diagram Wires::NamingHierarchyPart1

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

Figure 55 shows class diagram NamingHierarchyPart2.

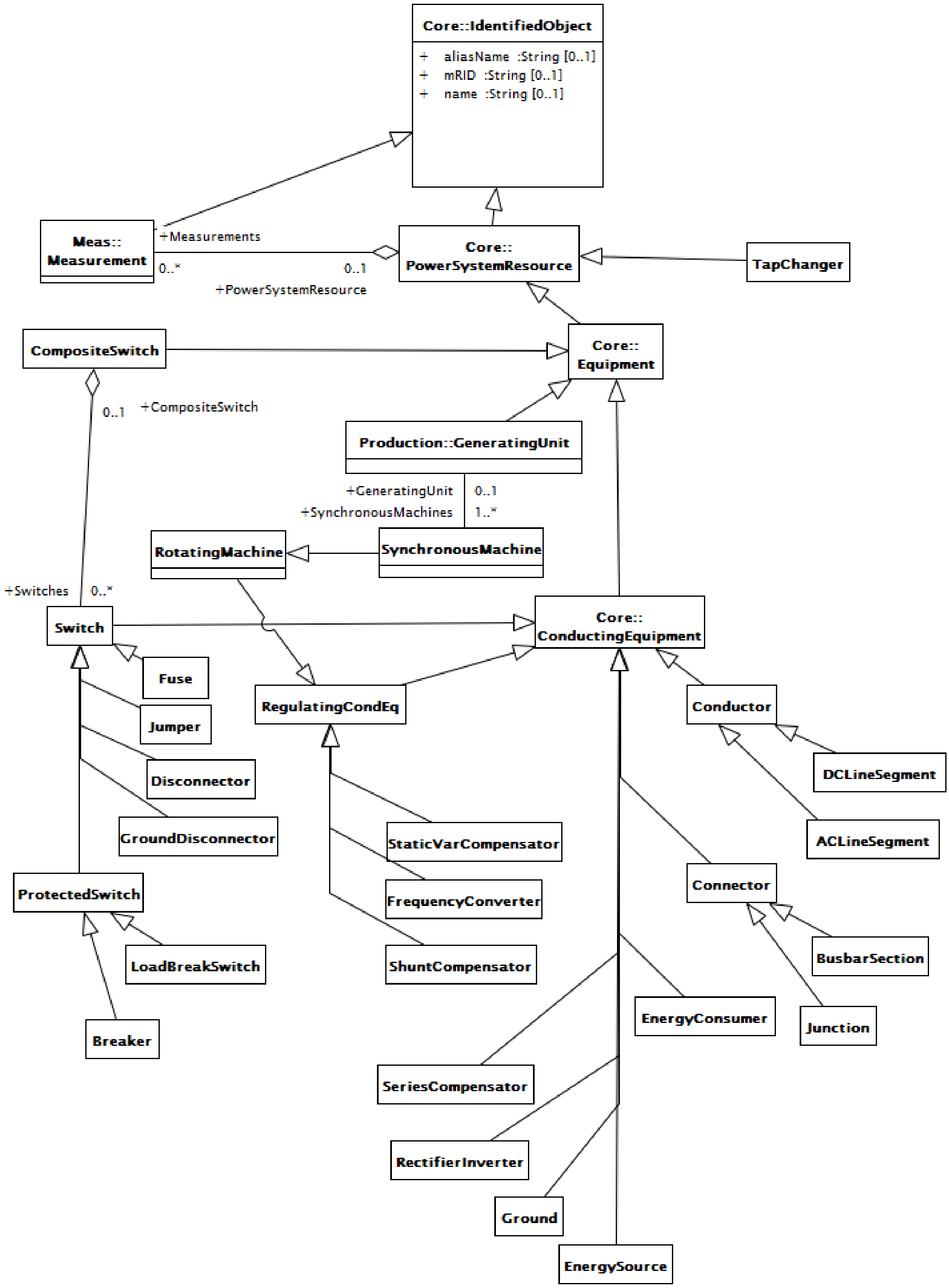

Figure 55 – Class diagram Wires::NamingHierarchyPart2

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

Figure 56 shows class diagram RegulatingEquipment.

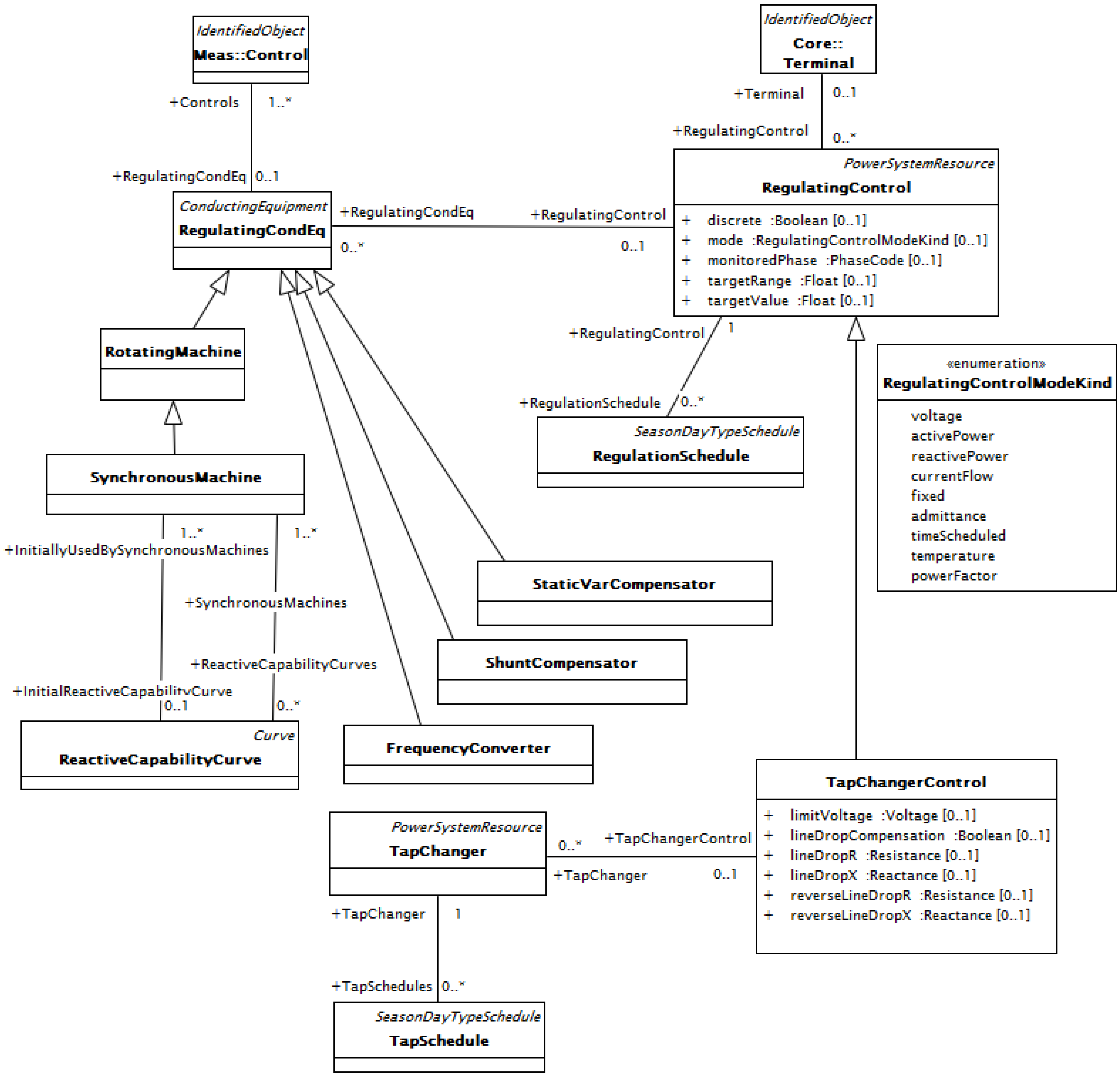

Figure 56 – Class diagram Wires::RegulatingEquipment

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

Figure 57 shows class diagram TapChanger.

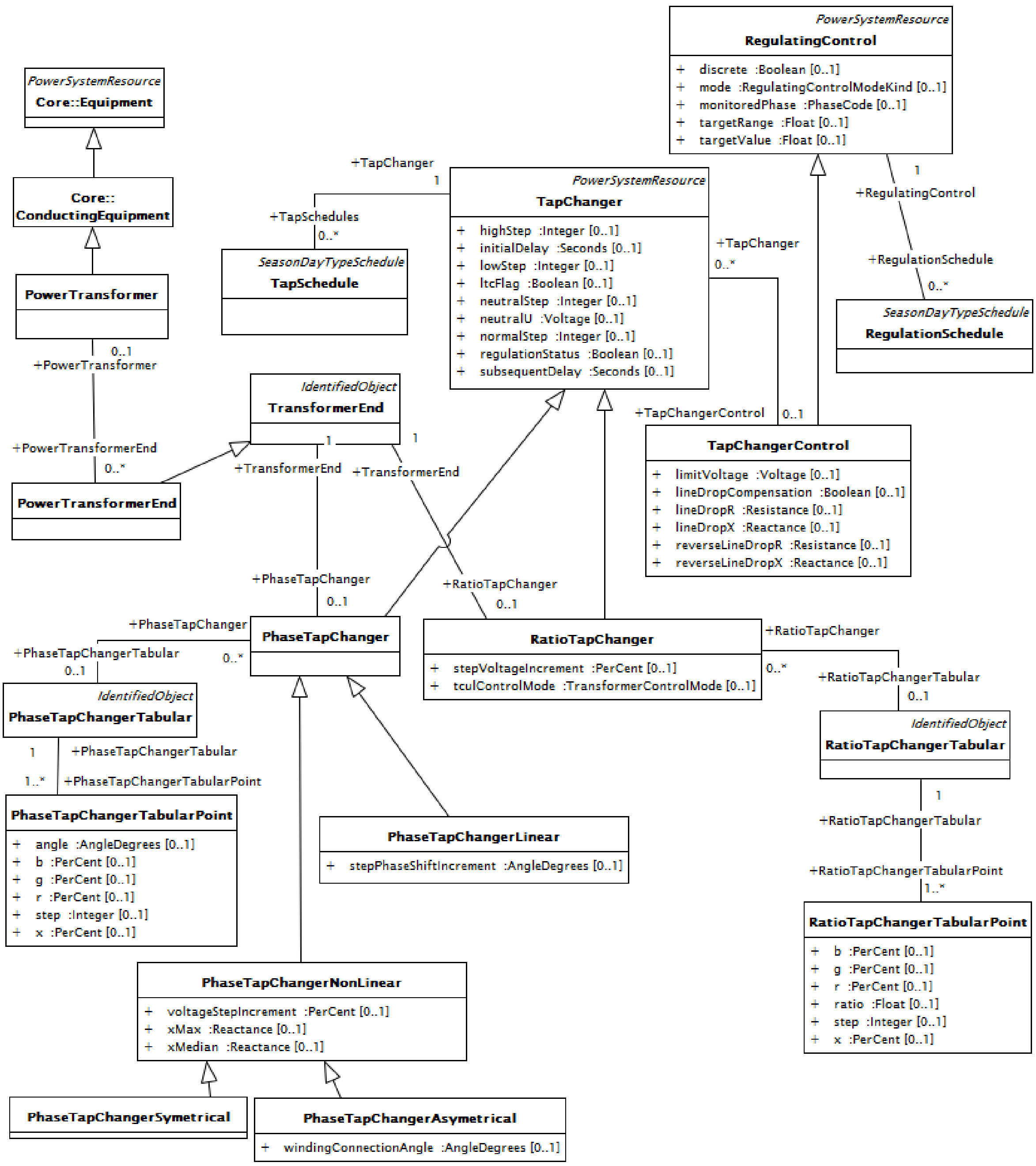

Figure 57 – Class diagram Wires::TapChanger

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

Figure 58 shows class diagram VoltageControl.

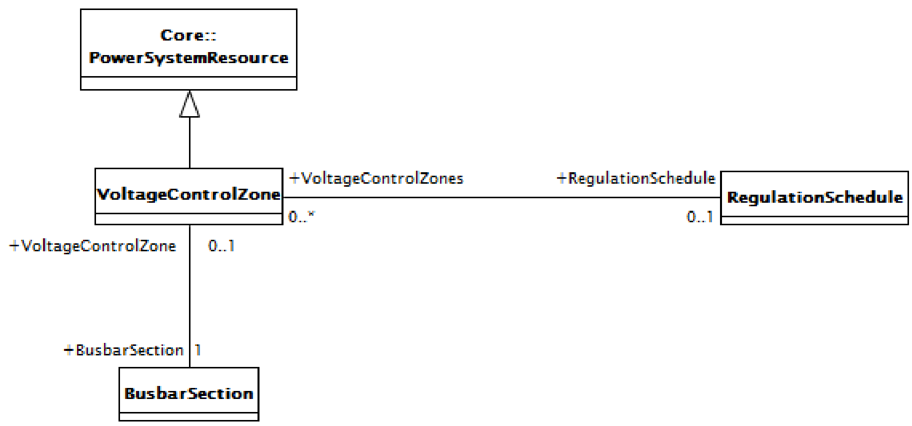

Figure 58 – Class diagram Wires::VoltageControl

This diagram shows all classes related to area voltage control.

Figure 59 shows class diagram Transformer.

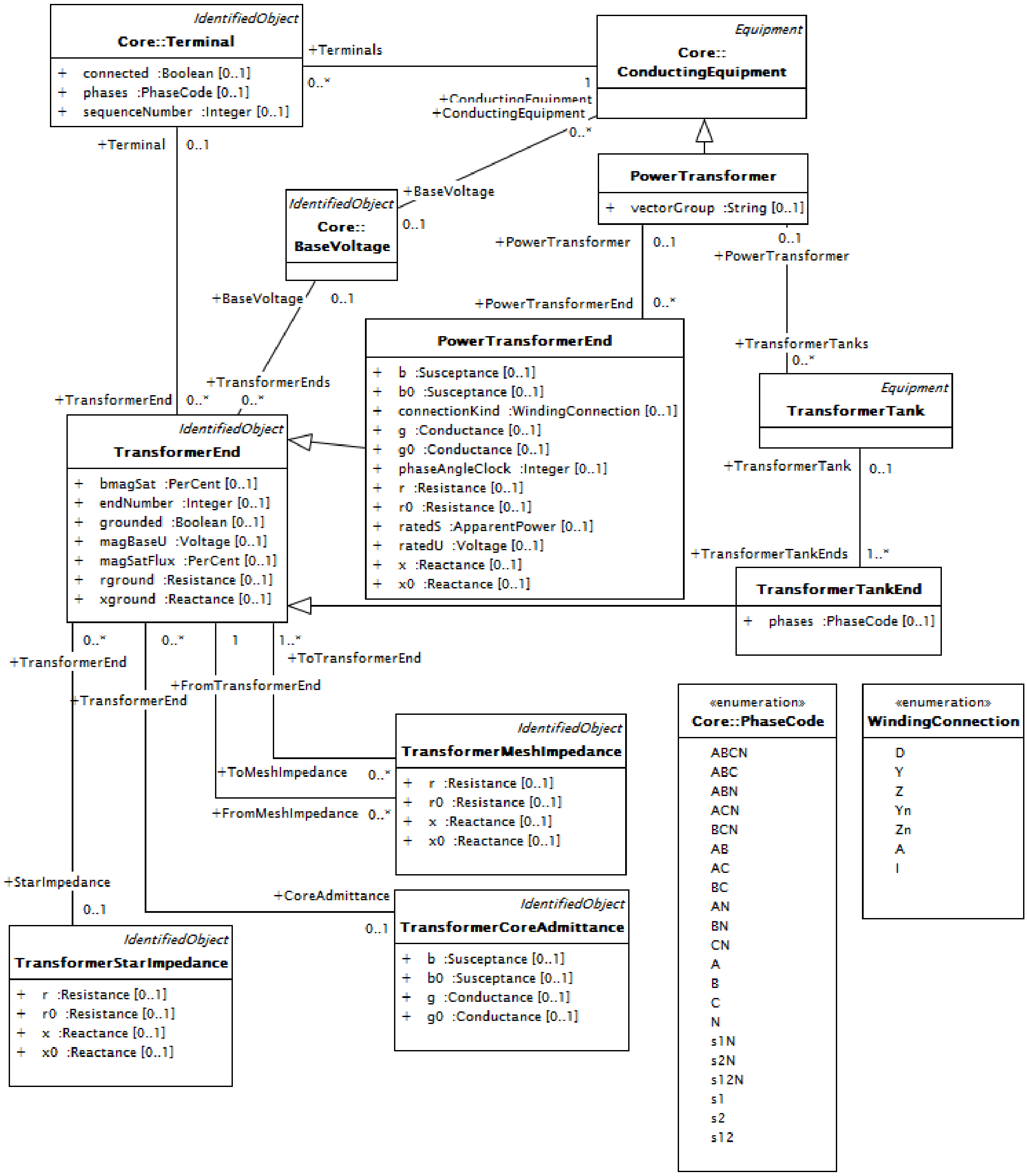

Figure 59 – Class diagram Wires::Transformer

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

6.8.2 ACLineSegment

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

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

Table 166 shows all attributes of ACLineSegment.

Table 166 – Attributes of Wires::ACLineSegment

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

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

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

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

6.8.3 ACLineSegmentPhase

Represents a single wire of an alternating current line segment.

Table 168 shows all attributes of ACLineSegmentPhase.

Table 168 – Attributes of Wires::ACLineSegmentPhase

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

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

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

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

6.8.4 AsynchronousMachine

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

Table 170 shows all attributes of AsynchronousMachine.

Table 170 – Attributes of Wires::AsynchronousMachine

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

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

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

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

6.8.5 Breaker

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

Table 172 shows all attributes of Breaker.

Table 172 – Attributes of Wires::Breaker

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

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

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

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

6.8.6 BusbarSection

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

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

Table 174 shows all attributes of BusbarSection.

Table 174 – Attributes of Wires::BusbarSection

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

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

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

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

6.8.7 Clamp

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

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

Table 176 shows all attributes of Clamp.

Table 176 – Attributes of Wires::Clamp

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

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

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

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

6.8.8 CompositeSwitch

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

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

Table 178 shows all attributes of CompositeSwitch.

Table 178 – Attributes of Wires::CompositeSwitch

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

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

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

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

6.8.9 CompositeSwitchType datatype

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

Table 180 shows all attributes of CompositeSwitchType.

Table 180 – Attributes of Wires::CompositeSwitchType

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

6.8.10 Conductor

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

Table 181 shows all attributes of Conductor.

Table 181 – Attributes of Wires::Conductor

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

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

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

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

6.8.11 Connector

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

Table 183 shows all attributes of Connector.

Table 183 – Attributes of Wires::Connector

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

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

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

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

6.8.12 CoolantType enumeration

Method of cooling a machine.

Table 185 shows all literals of CoolantType.

Table 185 – Literals of Wires::CoolantType

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

6.8.13 Cut

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

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

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

Table 186 shows all attributes of Cut.

Table 186 – Attributes of Wires::Cut

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

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

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

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

6.8.14 DCLineSegment

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

Table 188 shows all attributes of DCLineSegment.

Table 188 – Attributes of Wires::DCLineSegment

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

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

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

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

6.8.15 Disconnector

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

Table 190 shows all attributes of Disconnector.

Table 190 – Attributes of Wires::Disconnector

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

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

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

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

6.8.16 EnergyConsumer

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

Table 192 shows all attributes of EnergyConsumer.

Table 192 – Attributes of Wires::EnergyConsumer

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

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

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

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

6.8.17 EnergyConsumerPhase

A single phase of an energy consumer.

Table 194 shows all attributes of EnergyConsumerPhase.

Table 194 – Attributes of Wires::EnergyConsumerPhase

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

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

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

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

6.8.18 EnergySource

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

Table 196 shows all attributes of EnergySource.

Table 196 – Attributes of Wires::EnergySource

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

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

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

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

6.8.19 FrequencyConverter

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

Table 198 shows all attributes of FrequencyConverter.

Table 198 – Attributes of Wires::FrequencyConverter

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

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

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

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

6.8.20 Fuse

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

Table 200 shows all attributes of Fuse.

Table 200 – Attributes of Wires::Fuse

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

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

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

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

6.8.21 Ground

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

Table 202 shows all attributes of Ground.

Table 202 – Attributes of Wires::Ground

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

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

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

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

6.8.22 GroundDisconnector

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

Table 204 shows all attributes of GroundDisconnector.

Table 204 – Attributes of Wires::GroundDisconnector

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

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

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

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

6.8.23 Jumper

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

Table 206 shows all attributes of Jumper.

Table 206 – Attributes of Wires::Jumper

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

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

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

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

6.8.24 Junction

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

Table 208 shows all attributes of Junction.

Table 208 – Attributes of Wires::Junction

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

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

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

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

6.8.25 Line

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

Table 210 shows all attributes of Line.

Table 210 – Attributes of Wires::Line

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

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

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

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

6.8.26 LoadBreakSwitch

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

Table 212 shows all attributes of LoadBreakSwitch.

Table 212 – Attributes of Wires::LoadBreakSwitch

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

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

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

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

6.8.27 MutualCoupling

This class represents the zero sequence line mutual coupling.

Table 214 shows all attributes of MutualCoupling.

Table 214 – Attributes of Wires::MutualCoupling

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

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

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

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

6.8.28 OperatingMode datatype

Textual name for an operating mode.

Table 216 shows all attributes of OperatingMode.

Table 216 – Attributes of Wires::OperatingMode

name	type	description
multiplier	UnitMultiplier	
unit	UnitSymbol	
value	String	

6.8.29 PerLengthImpedance

Common type for per-length impedance electrical catalogues.

Table 217 shows all attributes of PerLengthImpedance.

Table 217 – Attributes of Wires::PerLengthImpedance

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

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

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

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

6.8.30 PerLengthPhaseImpedance

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

Table 219 shows all attributes of PerLengthPhaseImpedance.

Table 219 – Attributes of Wires::PerLengthPhaseImpedance

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

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

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

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

6.8.31 PerLengthSequenceImpedance

Sequence impedance and admittance parameters per unit length, for transposed lines of 1, 2, or 3 phases. For 1-phase lines, define x=x0=xself. For 2-phase lines, define x=xs-xm and x0=xs+xm.

Table 221 shows all attributes of PerLengthSequenceImpedance.

Table 221 – Attributes of Wires::PerLengthSequenceImpedance

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

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

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

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

6.8.32 PhaseImpedanceData root class

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

Table 223 shows all attributes of PhaseImpedanceData.

Table 223 – Attributes of Wires::PhaseImpedanceData

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

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

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

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

6.8.33 PhaseShuntConnectionKind enumeration

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

Table 225 shows all literals of PhaseShuntConnectionKind.

Table 225 – Literals of Wires::PhaseShuntConnectionKind

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

6.8.34 PhaseTapChanger

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

Table 226 shows all attributes of PhaseTapChanger.

Table 226 – Attributes of Wires::PhaseTapChanger

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

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

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

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

6.8.35 PhaseTapChangerAsymetrical

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

Table 228 shows all attributes of PhaseTapChangerAsymetrical.

Table 228 – Attributes of Wires::PhaseTapChangerAsymetrical

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

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

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

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

6.8.36 PhaseTapChangerLinear

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

Table 230 shows all attributes of PhaseTapChangerLinear.

Table 230 – Attributes of Wires::PhaseTapChangerLinear

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

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

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

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

6.8.37 PhaseTapChangerNonLinear

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

Table 232 shows all attributes of PhaseTapChangerNonLinear.

Table 232 – Attributes of Wires::PhaseTapChangerNonLinear

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

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

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

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

6.8.38 PhaseTapChangerSymetrical

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

Table 234 shows all attributes of PhaseTapChangerSymetrical.

Table 234 – Attributes of Wires::PhaseTapChangerSymetrical

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

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

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

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

6.8.39 PhaseTapChangerTabular

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

Table 236 shows all attributes of PhaseTapChangerTabular.

Table 236 – Attributes of Wires::PhaseTapChangerTabular

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

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

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

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

6.8.40 PhaseTapChangerTabularPoint root class

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

Table 238 shows all attributes of PhaseTapChangerTabularPoint.

Table 238 – Attributes of Wires::PhaseTapChangerTabularPoint

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

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

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

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

6.8.41 Plant

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

Table 240 shows all attributes of Plant.

Table 240 – Attributes of Wires::Plant

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

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

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

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

6.8.42 PowerTransformer

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

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

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

Table 242 shows all attributes of PowerTransformer.

Table 242 – Attributes of Wires::PowerTransformer

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

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

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

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

6.8.43 PowerTransformerEnd

A PowerTransformerEnd is associated with each Terminal of a PowerTransformer.

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

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

Table 244 shows all attributes of PowerTransformerEnd.

Table 244 – Attributes of Wires::PowerTransformerEnd

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

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

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

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

6.8.44 ProtectedSwitch

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

Table 246 shows all attributes of ProtectedSwitch.

Table 246 – Attributes of Wires::ProtectedSwitch

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

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

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

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

6.8.45 RatioTapChanger

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

Table 248 shows all attributes of RatioTapChanger.

Table 248 – Attributes of Wires::RatioTapChanger

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

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

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

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

6.8.46 RatioTapChangerTabular

Describes a curve for how the voltage magnitude and impedance vary with the tap step.

Table 250 shows all attributes of RatioTapChangerTabular.

Table 250 – Attributes of Wires::RatioTapChangerTabular

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

Table 251 shows all association ends of RatioTapChangerTabular with other classes.

Table 251 – Association ends of Wires::RatioTapChangerTabular with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] RatioTapChanger	RatioTapChanger	The ratio tap changer of this tap ratio table.
[1..1]	[1..*] RatioTapChangerTabularPoint	RatioTapChangerTabular Point	Points of this table.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.47 RatioTapChangerTabularPoint root class

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

Table 252 shows all attributes of RatioTapChangerTabularPoint.

Table 252 – Attributes of Wires::RatioTapChangerTabularPoint

name	type	description
b	PerCent	The magnetizing branch susceptance deviation in percent of nominal value. The actual susceptance is calculated as follows: calculated magnetizing susceptance = b(nominal) * (1 + b(from this class)/100). The b(nominal) is defined as the static magnetizing susceptance on the associated power transformer end or ends. This model assumes the star impedance (pi model) form.
g	PerCent	The magnetizing branch conductance deviation in percent of nominal value. The actual conductance is calculated as follows: calculated magnetizing conductance = g(nominal) * (1 + g(from this class)/100). The g(nominal) is defined as the static magnetizing conductance on the associated power transformer end or ends. This model assumes the star impedance (pi model) form.
r	PerCent	The resistance deviation in percent of nominal value. The actual reactance is calculated as follows: calculated resistance = r(nominal) * (1 + r(from this class)/100). The r(nominal) is defined as the static resistance on the associated power transformer end or ends. This model assumes the star impedance (pi model) form.
ratio	Float	The voltage ratio in per unit. Hence this is a value close to one.
step	Integer	The tap step.
x	PerCent	The series reactance deviation in percent of nominal value. The actual reactance is calculated as follows: calculated reactance = x(nominal) * (1 + x(from this class)/100). The x(nominal) is defined as the static series reactance on the associated power transformer end or ends. This model assumes the star impedance (pi model) form.

Table 253 shows all association ends of RatioTapChangerTabularPoint with other classes.

Table 253 – Association ends of Wires::RatioTapChangerTabularPoint with other classes

[mult from]	[mult to] name	type	description
[1..*]	[1..1] RatioTapChangerTabular	RatioTapChangerTabular	Table of this point.

6.8.48 ReactiveCapabilityCurve

Reactive power rating envelope versus the synchronous machine's active power, in both the generating and motoring modes. For each active power value there is a corresponding high and low reactive power limit value. Typically there will be a separate curve for each coolant condition, such as hydrogen pressure. The Y1 axis values represent reactive minimum and the Y2 axis values represent reactive maximum.

Table 254 shows all attributes of ReactiveCapabilityCurve.

Table 254 – Attributes of Wires::ReactiveCapabilityCurve

name	type	description
coolantTemperature	Temperature	The machine's coolant temperature (e.g., ambient air or stator circulating water).
hydrogenPressure	Pressure	The hydrogen coolant pressure
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 255 shows all association ends of ReactiveCapabilityCurve with other classes.

Table 255 – Association ends of Wires::ReactiveCapabilityCurve with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..*] InitiallyUsedBySynchronousMachines	SynchronousMachine	Synchronous machines using this curve as default.
[0..*]	[1..*] SynchronousMachines	SynchronousMachine	Synchronous machines using this curve.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.49 Recloser

Pole-mounted fault interrupter with built-in phase and ground relays, current transformer (CT), and supplemental controls.

Table 256 shows all attributes of Recloser.

Table 256 – Attributes of Wires::Recloser

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

Table 257 shows all association ends of Recloser with other classes.

Table 257 – Association ends of Wires::Recloser with other classes

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

[mult from]	[mult to] name	type	description
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.50 RectifierInverter

Bi-directional AC-DC conversion equipment that can be used to control DC current, DC voltage, DC power flow, or firing angle.

Table 258 shows all attributes of RectifierInverter.

Table 258 – Attributes of Wires::RectifierInverter

name	type	description
bridges	Integer	Number of bridges
commutatingReactance	Reactance	Commutating reactance at AC bus frequency.
commutatingResistance	Resistance	Commutating resistance.
compoundResistance	Resistance	Compounding resistance.
frequency	Frequency	Frequency on the AC side.
maxP	ActivePower	The maximum active power on the DC side at which the fconverter should operate.
maxU	Voltage	The maximum voltage on the DC side at which the converter should operate.
minCompoundVoltage	Voltage	Minimum compounded DC voltage.
minP	ActivePower	The minimum active power on the DC side at which the converter should operate.
minU	Voltage	The minimum voltage on the DC side at which the converter should operate.
operatingMode	OperatingMode	Operating mode for the converter.
ratedU	Voltage	Rectifier/inverter primary base voltage.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 259 shows all association ends of RectifierInverter with other classes.

Table 259 – Association ends of Wires::RectifierInverter with other classes

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

6.8.51 RegulatingCondEq

A type of conducting equipment that can regulate a quanity (i.e. voltage or flow) at a specific point in the network.

Table 260 shows all attributes of RegulatingCondEq.

Table 260 – Attributes of Wires::RegulatingCondEq

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

Table 261 shows all association ends of RegulatingCondEq with other classes.

Table 261 – Association ends of Wires::RegulatingCondEq with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] RegulatingControl	RegulatingControl	The regulating control scheme in which this equipment participates.
[0..1]	[1..*] Controls	Control	The controller outputs used to actually govern a regulating device, e.g. the magnetization of a synchronous machine or capacitor bank breaker actuator.
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.52 RegulatingControl

Specifies a set of equipment that works together to control a power system quantity such as voltage or flow.

Table 262 shows all attributes of RegulatingControl.

Table 262 – Attributes of Wires::RegulatingControl

name	type	description
mode	RegulatingControlModeKind	The regulating control mode presently available. This specifications allows for determining the kind of regulation without need for obtaining the units from a schedule.
targetRange	Float	This is the case input target range. This performs the same function as the value2 attribute on the regulation schedule in the case that schedules are not used. The units of those appropriate for the mode.
targetValue	Float	The target value specified for case input. This value can be used for the target value wihout the use of schedules. The value has the units appropriate to the mode attribute.
discrete	Boolean	The regulation is performed in a discrete mode.
monitoredPhase	PhaseCode	Phase voltage controlling this regulator, measured at regulator location.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 263 shows all association ends of RegulatingControl with other classes.

Table 263 – Association ends of Wires::RegulatingControl with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] Terminal	Terminal	The terminal associated with this regulating control. The terminal is associated instead of a node, since the terminal could connect into either a topological node (bus in bus-branch model) or a connectivity node (detailed switch model). Sometimes it is useful to model regulation at a terminal of a bus bar object since the bus bar can be present in both a bus-branch model or a model with switch detail.
[0..1]	[0..*] RegulatingCondEq	RegulatingCondEq	The equipment that participates in this regulating control scheme.
[1..1]	[0..*] RegulationSchedule	RegulationSchedule	Schedule for this regulating control.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.53 RegulatingControlModeKind enumeration

The kind of regulation model. For example regulating voltage, reactive power, active power, etc.

Table 264 shows all literals of RegulatingControlModeKind.

Table 264 – Literals of Wires::RegulatingControlModeKind

literal	description
voltage	Voltage is specified.
activePower	Active power is specified.
reactivePower	Reactive power is specified.
currentFlow	Current flow is specified.
fixed	The regulation mode is fixed, and thus not regulating.
admittance	Admittance is specified.
timeScheduled	Control switches on/off by time of day. The times may change on the weekend, or in different seasons.
temperature	Control switches on/off based on the local temperature (i.e., a thermostat).
powerFactor	Power factor is specified.

6.8.54 RegulationSchedule

A pre-established pattern over time for a controlled variable, e.g., busbar voltage.

Table 265 shows all attributes of RegulationSchedule.

Table 265 – Attributes of Wires::RegulationSchedule

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

Table 266 shows all association ends of RegulationSchedule with other classes.

Table 266 – Association ends of Wires::RegulationSchedule with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] RegulatingControl	RegulatingControl	Regulating controls that have this Schedule.
[0..1]	[0..*] VoltageControlZones	VoltageControlZone	A VoltageControlZone may have a voltage regulation schedule.
[0..*]	[0..1] DayType	DayType	inherited from: SeasonDayTypeSchedule
[0..*]	[0..1] Season	Season	inherited from: SeasonDayTypeSchedule
[1..1]	[1..*] TimePoints	RegularTimePoint	inherited from: RegularIntervalSchedule
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.55 Resistor

Resistor, typically used in filter configurations or as earthing resistor for transformers. Used for electrical model of distribution networks.

Table 267 shows all attributes of Resistor.

Table 267 – Attributes of Wires::Resistor

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

Table 268 shows all association ends of Resistor with other classes.

Table 268 – Association ends of Wires::Resistor with other classes

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

6.8.56 RotatingMachine

A rotating machine which may be used as a generator or motor.

Table 269 shows all attributes of RotatingMachine.

Table 269 – Attributes of Wires::RotatingMachine

name	type	description

name	type	description
damping	Float	Damping torque coefficient, a proportionality constant that, when multiplied by the angular velocity of the rotor poles with respect to the magnetic field (frequency), results in the damping torque. This value is often zero when the sources of damping torques (generator damper windings, load damping effects, etc.) are modeled in detail.
inertia	Seconds	Inertia constant of generator or motor and mechanical load. Must be greater than zero. This is the specification for the stored energy in the rotating mass when operating at rated speed. For a generator, this includes the generator plus all other elements (turbine, exciter) on the same shaft and has units of MW-sec. For a motor, it includes the motor plus its mechanical load. Conventional units are per unit on the generator MVA base, usually expressed as MW-second./MVA or just second. This value is used in the accelerating power reference frame for operator training simulator solutions.
ratedS	ApparentPower	Nameplate apparent power rating for the unit.
saturationFactor	Float	Saturation factor at rated terminal voltage. Should be greather than or equal to zero.
saturationFactor120	Float	Saturation factor at 120% of rated terminal voltage. Should be greater than or equal to the saturation factor at rated terminal voltage.
statorLeakageReactance	Reactance	Stator leakage reactance. Should be greater than or equal to 0.
statorResistance	Resistance	Stator (armature) resistance. Should be greather than or equal to zero. Equivalent resistance when used for GenEquiv model.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 270 shows all association ends of RotatingMachine with other classes.

Table 270 – Association ends of Wires::RotatingMachine with other classes

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

6.8.57 Sectionaliser

Automatic switch that will lock open to isolate a faulted section. It may, or may not, have load breaking capability. Its primary purpose is to provide fault sectionalising at locations where the fault current is either too high, or too low, for proper coordination of fuses.

Table 271 shows all attributes of Sectionaliser.

Table 271 – Attributes of Wires::Sectionaliser

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

Table 272 shows all association ends of Sectionaliser with other classes.

Table 272 – Association ends of Wires::Sectionaliser with other classes

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

6.8.58 SeriesCompensator

A Series Compensator is a series capacitor or reactor or an AC transmission line without charging susceptance. It is a two terminal device.

Table 273 shows all attributes of SeriesCompensator.

Table 273 – Attributes of Wires::SeriesCompensator

name	type	description
r	Resistance	Positive sequence resistance.
r0	Resistance	Zero sequence resistance.
x	Reactance	Positive sequence reactance.
x0	Reactance	Zero sequence reactance.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 274 shows all association ends of SeriesCompensator with other classes.

Table 274 – Association ends of Wires::SeriesCompensator with other classes

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

6.8.59 ShuntCompensator

A shunt capacitor or reactor or switchable bank of shunt capacitors or reactors. A section of a shunt compensator is an individual capacitor or reactor. A negative value for reactivePerSection indicates that the compensator is a reactor. ShuntCompensator is a single terminal device. Ground is implied.

Table 275 shows all attributes of ShuntCompensator.

Table 275 – Attributes of Wires::ShuntCompensator

name	type	description
aVRDelay	Seconds	Time delay required for the device to be connected or disconnected by automatic voltage regulation (AVR).
b0PerSection	Susceptance	Zero sequence shunt (charging) susceptance per section
bPerSection	Susceptance	Positive sequence shunt (charging) susceptance per section
g0PerSection	Conductance	Zero sequence shunt (charging) conductance per section
gPerSection	Conductance	Positive sequence shunt (charging) conductance per section
grounded	WindingConnection	Used for Yn and Zn connections. True if the neutral is solidly grounded.
maximumSections	Integer	The maximum number of sections that may be switched in.
nomU	Voltage	The voltage at which the nominal reactive power may be calculated. This should normally be within 10% of the voltage at which the capacitor is connected to the network.
normalSections	Integer	The normal number of sections switched in.
phaseConnection	PhaseShuntConnectionKind	The type of phase connection, such as wye or delta.
switchOnCount	Integer	The switch on count since the capacitor count was last reset or initialized.
switchOnDate	DateTime	The date and time when the capacitor bank was last switched on.
voltageSensitivity	VoltagePerReactivePower	Voltage sensitivity required for the device to regulate the bus voltage, in voltage/reactive power.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 276 shows all association ends of ShuntCompensator with other classes.

Table 276 – Association ends of Wires::ShuntCompensator with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] ShuntCompensatorPhase	ShuntCompensatorPhase	The individual phases models for the shunt compensator.
[1..1]	[0..1] SvShuntCompensatorSections	SvShuntCompensatorSections	The state for the number of shunt compensator sections in service.
[0..*]	[0..1] RegulatingControl	RegulatingControl	inherited from: RegulatingCondEq
[0..1]	[1..*] Controls	Control	inherited from: RegulatingCondEq
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.60 ShuntCompensatorPhase

Single phase of a multi-phase shunt compensator when its attributes might be different per phase.

Table 277 shows all attributes of ShuntCompensatorPhase.

Table 277 – Attributes of Wires::ShuntCompensatorPhase

name	type	description
bPerSection	Susceptance	Susceptance per section of the phase if shunt compensator is wye connected. Susceptance per section phase to phase if shunt compensator is delta connected.
gPerSection	Conductance	Conductance per section for this phase if shunt compensator is wye connected. Conductance per section phase to phase if shunt compensator is delta connected.
maximumSections	Integer	The maximum number of sections that may be switched in for this phase.
normalSections	Integer	For the capacitor phase, the normal number of sections switched in.
phase	SinglePhaseKind	Phase of this shunt compensator component. If the shunt compensator is wye connected, the connection is from the indicated phase to the central ground or neutral point. If the shunt compensator is delta connected, the phase indicates a shunt compensator connected from the indicated phase to the next logical non-neutral phase.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject

name	type	description
name	String	inherited from: IdentifiedObject

Table 278 shows all association ends of ShuntCompensatorPhase with other classes.

Table 278 – Association ends of Wires::ShuntCompensatorPhase with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] ShuntCompensator	ShuntCompensator	Shunt compensator of this shunt compensator phase.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.61 SinglePhaseKind enumeration

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

Table 279 shows all literals of SinglePhaseKind.

Table 279 – Literals of Wires::SinglePhaseKind

literal	description
A	Phase A.
B	Phase B.
C	Phase C.
N	Neutral.
s1	Secondary phase 1.
s2	Secondary phase 2.

6.8.62 StaticVarCompensator

A facility for providing variable and controllable shunt reactive power. The SVC typically consists of a stepdown transformer, filter, thyristor-controlled reactor, and thyristor-switched capacitor arms.

The SVC may operate in fixed MVar output mode or in voltage control mode. When in voltage control mode, the output of the SVC will be proportional to the deviation of voltage at the controlled bus from the voltage setpoint. The SVC characteristic slope defines the proportion. If the voltage at the controlled bus is equal to the voltage setpoint, the SVC MVar output is zero.

Table 280 shows all attributes of StaticVarCompensator.

Table 280 – Attributes of Wires::StaticVarCompensator

name	type	description
capacitiveRating	Reactance	Maximum available capacitive reactance.
inductiveRating	Reactance	Maximum available inductive reactance.
slope	VoltagePerReactivePower	The characteristics slope of an SVC defines how the reactive power output changes in proportion to the difference between the regulated bus voltage and the voltage setpoint.
sVCControlMode	SVCControlMode	SVC control mode.
voltageSetPoint	Voltage	The reactive power output of the SVC is proportional to the difference between the voltage at the regulated bus and the voltage setpoint. When the regulated bus voltage is equal to the voltage setpoint, the reactive power output is zero.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 281 shows all association ends of StaticVarCompensator with other classes.

Table 281 – Association ends of Wires::StaticVarCompensator with other classes

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

6.8.63 SVCControlMode enumeration

Static VAr Compensator control mode.

Table 282 shows all literals of SVCControlMode.

Table 282 – Literals of Wires::SVCControlMode

literal	description
reactivePower	
voltage	
off	

6.8.64 Switch

A generic device designed to close, or open, or both, one or more electric circuits.

Table 283 shows all attributes of Switch.

Table 283 – Attributes of Wires::Switch

name	type	description
normalOpen	Boolean	The attribute is used in cases when no Measurement for the status value is present. If the Switch has a status measurment the Discrete.normalValue is expected to match with the Switch.normalOpen.
ratedCurrent	CurrentFlow	The maximum continuous current carrying capacity in amps governed by the device material and construction.
switchOnCount	Integer	The switch on count since the switch was last reset or initialized.
switchOnDate	DateTime	The date and time when the switch was last switched on.
retained	Boolean	Branch is retained in a bus branch model.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 284 shows all association ends of Switch with other classes.

Table 284 – Association ends of Wires::Switch with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] SwitchSchedules	SwitchSchedule	A Switch can be associated with SwitchSchedules.
[1..1]	[0..*] SwitchPhase	SwitchPhase	The individual switch phases for the switch.
[0..*]	[0..1] CompositeSwitch	CompositeSwitch	Composite switch to which this Switch belongs.
[0..*]	[0..*] SwitchingOperations	SwitchingOperation	A switch may be operated by many schedules.
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.65 SwitchPhase

Single phase of a multi-phase switch when its attributes might be different per phase.

Table 285 shows all attributes of SwitchPhase.

Table 285 – Attributes of Wires::SwitchPhase

name	type	description
normalOpen	Boolean	Used in cases when no Measurement for the status value is present. If the SwitchPhase has a status measurement the Discrete.normalValue is expected to match with this value.
phaseSide1	SinglePhaseKind	Phase of this SwitchPhase on the side with terminal sequence number equal 1. Should be a phase contained in that terminal's phases attribute.
phaseSide2	SinglePhaseKind	Phase of this SwitchPhase on the side with terminal sequence number equal 2. Should be a phase contained in that terminal's Terminal.phases attribute.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 286 shows all association ends of SwitchPhase with other classes.

Table 286 – Association ends of Wires::SwitchPhase with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Switch	Switch	The switch of the switch phase.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.66 SwitchSchedule

A schedule of switch positions. If RegularTimePoint.value1 is 0, the switch is open. If 1, the switch is closed.

Table 287 shows all attributes of SwitchSchedule.

Table 287 – Attributes of Wires::SwitchSchedule

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

Table 288 shows all association ends of SwitchSchedule with other classes.

Table 288 – Association ends of Wires::SwitchSchedule with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Switch	Switch	A SwitchSchedule is associated with a Switch.
[0..*]	[0..1] DayType	DayType	inherited from: SeasonDayTypeSchedule
[0..*]	[0..1] Season	Season	inherited from: SeasonDayTypeSchedule
[1..1]	[1..*] TimePoints	RegularTimePoint	inherited from: RegularIntervalSchedule
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.67 SynchronousGeneratorType enumeration

Type of synchronous generator as used in dynamic simulation applications.

Table 289 shows all literals of SynchronousGeneratorType.

Table 289 – Literals of Wires::SynchronousGeneratorType

literal	description
roundRotor	Also known as genrou.
salientPole	Also known as gensal.
transient	Also known as gentra.
typeF	Also known as typeF.
typeJ	Also known as typeJ.

6.8.68 SynchronousMachine

An electromechanical device that operates with shaft rotating synchronously with the network. It is a single machine operating either as a generator or synchronous condenser or pump.

Table 290 shows all attributes of SynchronousMachine.

Table 290 – Attributes of Wires::SynchronousMachine

name	type	description
aVRToManualLag	Seconds	Time delay required when switching from Automatic Voltage Regulation (AVR) to Manual for a lagging MVAr violation.
aVRToManualLead	Seconds	Time delay required when switching from Automatic Voltage Regulation (AVR) to Manual for a leading MVAr violation.
baseQ	ReactivePower	Default base reactive power value. This value represents the initial reactive power that can be used by any application function.
condenserP	ActivePower	Active power consumed when in condenser mode operation.
coolantCondition	Float	Temperature or pressure of coolant medium
coolantType	CoolantType	Method of cooling the machine.
manualToAVR	Seconds	Time delay required when switching from Manual to Automatic Voltage Regulation. This value is used in the accelerating power reference frame for powerflow solutions
maxQ	ReactivePower	Maximum reactive power limit. This is the maximum (nameplate) limit for the unit.
maxU	Voltage	Maximum voltage limit for the unit.
minQ	ReactivePower	Minimum reactive power limit for the unit.
minU	Voltage	Minimum voltage limit for the unit.
operatingMode	SynchronousMachineOperatingMode	Current mode of operation.
qPercent	PerCent	Percent of the coordinated reactive control that comes from this machine.
r	Resistance	Positive sequence resistance of the synchronous machine.
r0	Resistance	Zero sequence resistance of the synchronous machine.
r2	Resistance	Negative sequence resistance.
referencePriority	Integer	Priority of unit for use as powerflow voltage phase angle reference bus selection. 0 = don t care (default) 1 = highest priority. 2 is less than 1 and so on.
synchronousGeneratorType	SynchronousGeneratorType	Type of Synchronous Generator used in Dynamic simulation applications.
tpdo	Seconds	Direct axis transient rotor time constant, also known as T'do
tppdo	Seconds	Direct axis sub-transient rotor time constant, also known as T''do
tppqo	Seconds	Quadrature axis subtransient rotor time constant, also known as T"qo
tpqo	Seconds	Quadrature axis transient rotor time constant, also known as T'qo
type	SynchronousMachineType	Modes that this synchronous machine can operate in.
x	Reactance	Positive sequence reactance of the synchronous machine.
x0	Reactance	Zero sequence reactance of the synchronous machine.
x2	Reactance	Negative sequence reactance.
xDirectSubtrans	Reactance	Direct-axis subtransient reactance, also known as X"d.

name	type	description
xDirectSync	Reactance	Direct-axis synchronous reactance. The quotient of a sustained value of that AC component of armature voltage that is produced by the total direct-axis flux due to direct-axis armature current and the value of the AC component of this current, the machine running at rated speed. (Xd)
xDirectTrans	Reactance	Direct-axis transient reactance, also known as X'd.
xQuadSubtrans	Reactance	Quadrature-axis subtransient reactance, also known as X"q.
xQuadSync	Reactance	Quadrature-axis synchronous reactance (Xq) , the ratio of the component of reactive armature voltage, due to the quadrature-axis component of armature current, to this component of current, under steady state conditions and at rated frequency.
xQuadTrans	Reactance	Quadrature-axis transient reactance, also known as X'q.
damping	Float	inherited from: RotatingMachine
inertia	Seconds	inherited from: RotatingMachine
ratedS	ApparentPower	inherited from: RotatingMachine
saturationFactor	Float	inherited from: RotatingMachine
saturationFactor120	Float	inherited from: RotatingMachine
statorLeakageReactance	Reactance	inherited from: RotatingMachine
statorResistance	Resistance	inherited from: RotatingMachine
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 291 shows all association ends of SynchronousMachine with other classes.

Table 291 – Association ends of Wires::SynchronousMachine with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..1] HydroPump	HydroPump	The synchronous machine drives the turbine which moves the water from a low elevation to a higher elevation. The direction of machine rotation for pumping may or may not be the same as for generating.
[1..*]	[0..1] InitialReactiveCapabilityCurve	ReactiveCapabilityCurve	The default reactive capability curve for use by a synchronous machine.
[1..*]	[0..*] ReactiveCapabilityCurves	ReactiveCapabilityCurve	All available reactive capability curves for this synchronous machine.
[1..*]	[0..1] GeneratingUnit	GeneratingUnit	A synchronous machine may operate as a generator and as such becomes a member of a generating unit.
[0..*]	[0..*] PrimeMovers	PrimeMover	Prime movers that drive this SynchronousMachine.
[0..*]	[0..1] RegulatingControl	RegulatingControl	inherited from: RegulatingCondEq
[0..1]	[1..*] Controls	Control	inherited from: RegulatingCondEq
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.69 SynchronousMachineOperatingMode enumeration

Synchronous machine operating mode.

Table 292 shows all literals of SynchronousMachineOperatingMode.

Table 292 – Literals of Wires::SynchronousMachineOperatingMode

literal	description
generator	
condenser	

6.8.70 SynchronousMachineType enumeration

Synchronous machine type.

Table 293 shows all literals of SynchronousMachineType.

Table 293 – Literals of Wires::SynchronousMachineType

literal	description
generator	
condenser	
generator_or_condenser	

6.8.71 TapChanger

Mechanism for changing transformer winding tap positions.

Table 294 shows all attributes of TapChanger.

Table 294 – Attributes of Wires::TapChanger

name	type	description
highStep	Integer	Highest possible tap step position, advance from neutral
initialDelay	Seconds	For an LTC, the delay for initial tap changer operation (first step change)
lowStep	Integer	Lowest possible tap step position, retard from neutral
ltcFlag	Boolean	Specifies whether or not a TapChanger has load tap changing capabilities.
neutralStep	Integer	The neutral tap step position for this winding.
neutralU	Voltage	Voltage at which the winding operates at the neutral tap setting.
normalStep	Integer	The tap step position used in "normal" network operation for this winding. For a "Fixed" tap changer indicates the current physical tap setting.
regulationStatus	Boolean	Specifies the default regulation status of the TapChanger. True is regulating. False is not regulating.
subsequentDelay	Seconds	For an LTC, the delay for subsequent tap changer operation (second and later step changes)
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 295 shows all association ends of TapChanger with other classes.

Table 295 – Association ends of Wires::TapChanger with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] TapChangerControl	TapChangerControl	The regulating control scheme in which this tap changer participates.
[1..1]	[0..1] SvTapStep	SvTapStep	The tap step state associated with the tap changer.
[1..1]	[0..*] TapSchedules	TapSchedule	A TapChanger can have TapSchedules.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.72 TapChangerControl

Discribes behaviour specific to tap changers, e.g. how the voltage at the end of a line varies with the load level and compensation of the voltage drop by tap adjustment.

Table 296 shows all attributes of TapChangerControl.

Table 296 – Attributes of Wires::TapChangerControl

name	type	description
limitVoltage	Voltage	Maximum allowed regulated voltage on the PT secondary, regardless of line drop compensation. Sometimes referred to as first-house protection.
lineDropCompensation	Boolean	If true, the line drop compensation is to be applied.
lineDropR	Resistance	Line drop compensator resistance setting for normal (forward) power flow.
lineDropX	Reactance	Line drop compensator reactance setting for normal (forward) power flow.
reverseLineDropR	Resistance	Line drop compensator resistance setting for reverse power flow.
reverseLineDropX	Reactance	Line drop compensator reactance setting for reverse power flow.
mode	RegulatingControlModeKind	inherited from: RegulatingControl
targetRange	Float	inherited from: RegulatingControl
targetValue	Float	inherited from: RegulatingControl
discrete	Boolean	inherited from: RegulatingControl
monitoredPhase	PhaseCode	inherited from: RegulatingControl
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 297 shows all association ends of TapChangerControl with other classes.

Table 297 – Association ends of Wires::TapChangerControl with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] TapChanger	TapChanger	The tap changer that participates in this regulating tap control scheme.
[0..*]	[0..1] Terminal	Terminal	inherited from: RegulatingControl
[0..1]	[0..*] RegulatingCondEq	RegulatingCondEq	inherited from: RegulatingControl
[1..1]	[0..*] RegulationSchedule	RegulationSchedule	inherited from: RegulatingControl
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.73 TapChangerKind enumeration

Transformer tap changer type. Indicates the capabilities of the tap changer independent of the operating mode.

Table 298 shows all literals of TapChangerKind.

Table 298 – Literals of Wires::TapChangerKind

literal	description
fixed	Not capable of control. This is also indicated by no association of TapChanger to a RegulatingControl.
voltageControl	Capable of voltage control.
phaseControl	Capable of phase control.
voltageAndPhaseControl	Capable of voltage and phase control.

6.8.74 TapSchedule

A pre-established pattern over time for a tap step.

Table 299 shows all attributes of TapSchedule.

Table 299 – Attributes of Wires::TapSchedule

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

Table 300 shows all association ends of TapSchedule with other classes.

Table 300 – Association ends of Wires::TapSchedule with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] TapChanger	TapChanger	A TapSchedule is associated with a TapChanger.
[0..*]	[0..1] DayType	DayType	inherited from: SeasonDayTypeSchedule
[0..*]	[0..1] Season	Season	inherited from: SeasonDayTypeSchedule
[1..1]	[1..*] TimePoints	RegularTimePoint	inherited from: RegularIntervalSchedule
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.75 TransformerControlMode enumeration

Control modes for a transformer.

Table 301 shows all literals of TransformerControlMode.

Table 301 – Literals of Wires::TransformerControlMode

literal	description
volt	Voltage control
reactive	Reactive power flow control

6.8.76 TransformerCoreAdmittance

The transformer core admittance. Used to specify the core admittance of a transformer in a manner that can be shared among power transformers.

Table 302 shows all attributes of TransformerCoreAdmittance.

Table 302 – Attributes of Wires::TransformerCoreAdmittance

name	type	description
b	Susceptance	Magnetizing branch susceptance (B mag). The value can be positive or negative.
b0	Susceptance	Zero sequence magnetizing branch susceptance.
g	Conductance	Magnetizing branch conductance (G mag).
g0	Conductance	Zero sequence magnetizing branch conductance.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 303 shows all association ends of TransformerCoreAdmittance with other classes.

Table 303 – Association ends of Wires::TransformerCoreAdmittance with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] TransformerEnd	TransformerEnd	All transformer ends having this core admittance.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.77 TransformerEnd

A conducting connection point of a power transformer. It corresponds to a physical transformer winding terminal. In earlier CIM versions, the TransformerWinding class served a similar purpose, but this class is more flexible because it associates to terminal but is not a specialization of ConductingEquipment.

Table 304 shows all attributes of TransformerEnd.

Table 304 – Attributes of Wires::TransformerEnd

name	type	description
bmagSat	PerCent	Core shunt magnetizing susceptance in the saturation region.
endNumber	Integer	Number for this transformer end, corresponding to the end's order in the power transformer vector group or phase angle clock number. Highest voltage winding should be 1. Each end within a power transformer should have a unique subsequent end number. Note the transformer end number need not match the terminal sequence number.
grounded	Boolean	(for Yn and Zn connections) True if the neutral is solidly grounded.
magBaseU	Voltage	The reference voltage at which the magnetizing saturation measurements were made
magSatFlux	PerCent	Core magnetizing saturation curve knee flux level.
rground	Resistance	(for Yn and Zn connections) Resistance part of neutral impedance where 'grounded' is true.
xground	Reactance	(for Yn and Zn connections) Reactive part of neutral impedance where 'grounded' is true.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 305 shows all association ends of TransformerEnd with other classes.

Table 305 – Association ends of Wires::TransformerEnd with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] BaseVoltage	BaseVoltage	Base voltage of the transformer end. This is essential for PU calculation.
[0..*]	[0..1] Terminal	Terminal	Terminal of the power transformer to which this transformer end belongs.
[1..1]	[0..1] PhaseTapChanger	PhaseTapChanger	Phase tap changer associated with this transformer end.
[1..1]	[0..1] RatioTapChanger	RatioTapChanger	Ratio tap changer associated with this transformer end.
[0..*]	[0..1] StarImpedance	TransformerStarImpedance	(accurate for 2- or 3-winding transformers only) Pi-model impedances of this transformer end. By convention, for a two winding transformer, the full values of the transformer should be entered on the high voltage end (endNumber=1).
[0..*]	[0..1] CoreAdmittance	TransformerCoreAdmittance	Core admittance of this transformer end, representing magnetising current and core losses. The full values of the transformer should be supplied for one transformer end only.
[1..1]	[0..*] FromMeshImpedance	TransformerMeshImpedance	All mesh impedances between this 'to' and other 'from' transformer ends.
[1..*]	[0..*] ToMeshImpedance	TransformerMeshImpedance	All mesh impedances between this 'from' and other 'to' transformer ends.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.78 TransformerMeshImpedance

Transformer mesh impedance (Delta-model) between transformer ends.

The typical case is that this class describes the impedance between two transformer ends pair-wise, i.e. the cardinalities at both tranformer end associations are 1. But in cases where two or more transformer ends are modeled the cardinalities are larger than 1.

Table 306 shows all attributes of TransformerMeshImpedance.

Table 306 – Attributes of Wires::TransformerMeshImpedance

name	type	description
r	Resistance	Resistance between the 'from' and the 'to' end, seen from the 'from' end.
r0	Resistance	Zero-sequence resistance between the 'from' and the 'to' end, seen from the 'from' end.
x	Reactance	Reactance between the 'from' and the 'to' end, seen from the 'from' end.
x0	Reactance	Zero-sequence reactance between the 'from' and the 'to' end, seen from the 'from' end.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 307 shows all association ends of TransformerMeshImpedance with other classes.

Table 307 – Association ends of Wires::TransformerMeshImpedance with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] FromTransformerEnd	TransformerEnd	From end this mesh impedance is connected to. It determines the voltage reference.
[0..*]	[1..*] ToTransformerEnd	TransformerEnd	All transformer ends this mesh impedance is connected to.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.79 TransformerStarImpedance

Transformer star impedance (Pi-model) that accurately reflects impedance for transformers with 2 or 3 windings. For transformers with 4 or more windings, you must use TransformerMeshImpedance class.

For transmission networks use PowerTransformerEnd impedances (r, r0, x, x0, b, b0, g and g0).

Table 308 shows all attributes of TransformerStarImpedance.

Table 308 – Attributes of Wires::TransformerStarImpedance

name	type	description
r	Resistance	Resistance of the transformer end.
r0	Resistance	Zero sequence series resistance of the transformer end.
x	Reactance	Positive sequence series reactance of the transformer end.
x0	Reactance	Zero sequence series reactance of the transformer end.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 309 shows all association ends of TransformerStarImpedance with other classes.

Table 309 – Association ends of Wires::TransformerStarImpedance with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] TransformerEnd	TransformerEnd	All transformer ends having this star impedance.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.80 TransformerTank

An assembly of two or more coupled windings that transform electrical power between voltage levels. These windings are bound on a common core and place in the same tank. Transformer tank can be used to model both single-phase and 3-phase transformers.

Table 310 shows all attributes of TransformerTank.

Table 310 – Attributes of Wires::TransformerTank

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

Table 311 shows all association ends of TransformerTank with other classes.

Table 311 – Association ends of Wires::TransformerTank with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] PowerTransformer	PowerTransformer	Bank this transformer belongs to.
[0..1]	[1..*] TransformerTankEnds	TransformerTankEnd	All windings of this transformer.
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.81 TransformerTankEnd

Transformer tank end represents an individual winding for unbalanced models or for transformer tanks connected into a bank (and bank is modelled with the PowerTransformer).

Table 312 shows all attributes of TransformerTankEnd.

Table 312 – Attributes of Wires::TransformerTankEnd

name	type	description
phases	PhaseCode	Describes the phases carried by a conducting equipment.
bmagSat	PerCent	inherited from: TransformerEnd
endNumber	Integer	inherited from: TransformerEnd
grounded	Boolean	inherited from: TransformerEnd
magBaseU	Voltage	inherited from: TransformerEnd
magSatFlux	PerCent	inherited from: TransformerEnd
rground	Resistance	inherited from: TransformerEnd
xground	Reactance	inherited from: TransformerEnd
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 313 shows all association ends of TransformerTankEnd with other classes.

Table 313 – Association ends of Wires::TransformerTankEnd with other classes

[mult from]	[mult to] name	type	description
[1..*]	[0..1] TransformerTank	TransformerTank	Transformer this winding belongs to.
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: TransformerEnd
[0..*]	[0..1] Terminal	Terminal	inherited from: TransformerEnd
[1..1]	[0..1] PhaseTapChanger	PhaseTapChanger	inherited from: TransformerEnd
[1..1]	[0..1] RatioTapChanger	RatioTapChanger	inherited from: TransformerEnd
[0..*]	[0..1] StarImpedance	TransformerStarImpedance	inherited from: TransformerEnd
[0..*]	[0..1] CoreAdmittance	TransformerCoreAdmittance	inherited from: TransformerEnd
[1..1]	[0..*] FromMeshImpedance	TransformerMeshImpedance	inherited from: TransformerEnd
[1..*]	[0..*] ToMeshImpedance	TransformerMeshImpedance	inherited from: TransformerEnd
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.82 VoltageControlZone

An area of the power system network which is defined for secondary voltage control purposes. A voltage control zone consists of a collection of substations with a designated bus bar section whose voltage will be controlled.

Table 314 shows all attributes of VoltageControlZone.

Table 314 – Attributes of Wires::VoltageControlZone

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

Table 315 shows all association ends of VoltageControlZone with other classes.

Table 315 – Association ends of Wires::VoltageControlZone with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] BusbarSection	BusbarSection	A VoltageControlZone is controlled by a designated BusbarSection.
[0..*]	[0..1] RegulationSchedule	RegulationSchedule	A VoltageControlZone may have a voltage regulation schedule.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.8.83 WindingConnection enumeration

Winding connection type.

Table 316 shows all literals of WindingConnection.

Table 316 – Literals of Wires::WindingConnection

literal	description
D	Delta
Y	Wye
Z	ZigZag
Yn	Wye, with neutral brought out for grounding.
Zn	ZigZag, with neutral brought out for grounding.
A	Autotransformer common winding
I	Independent winding, for single-phase connections

6.9 Package Generation

6.9.1 General

This package contains packages that have information for Unit Commitment and Economic Dispatch of Hydro and Thermal Generating Units, Load Forecasting, Automatic Generation Control, and Unit Modeling for Dynamic Training Simulator.

Figure 60 shows class diagram Main.

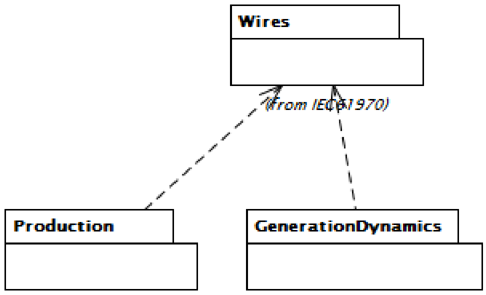

Figure 60 – Class diagram Generation::Main

This diagram shows the main generation packages and their dependency relationships to other packages.

6.9.2 Package GenerationDynamics

6.9.2.1 General

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

Figure 61 shows class diagram Main.

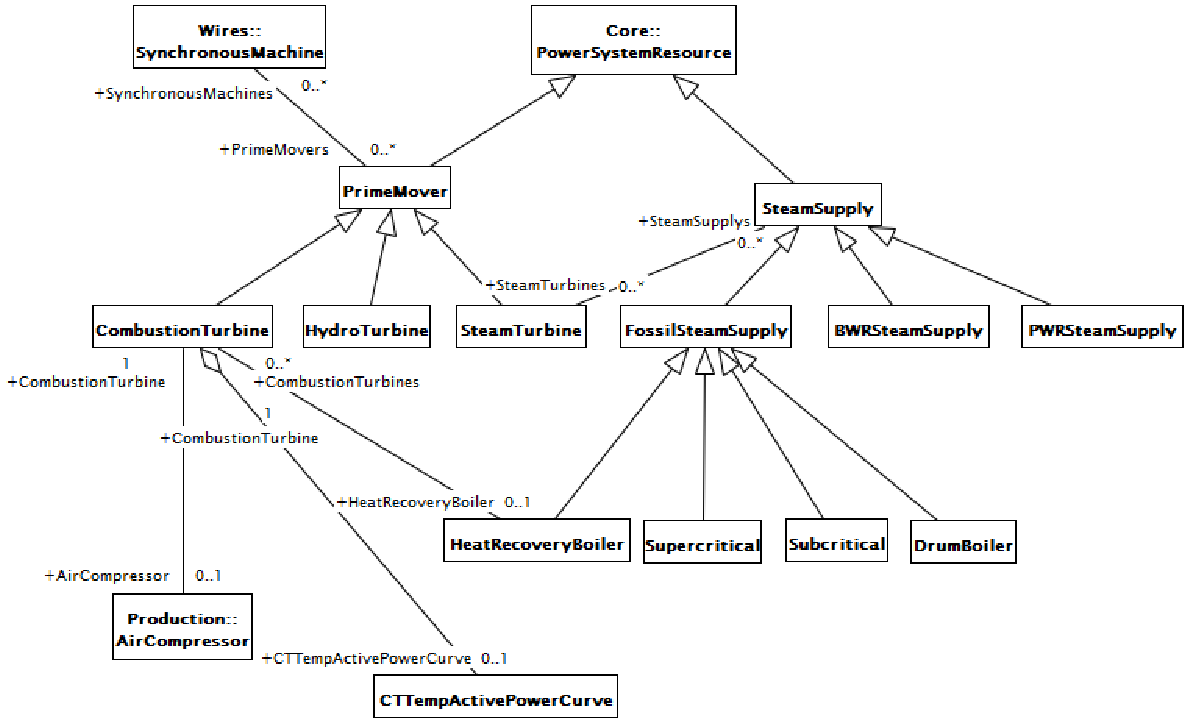

Figure 61 – Class diagram GenerationDynamics::Main

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

Figure 62 shows class diagram Datatypes.

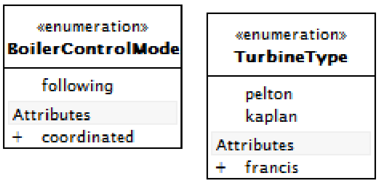

Figure 62 – Class diagram GenerationDynamics::Datatypes

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

6.9.2.2 BoilerControlMode enumeration

Boiler control mode.

Table 317 shows all literals of BoilerControlMode.

Table 317 – Literals of GenerationDynamics::BoilerControlMode

literal	description
following	Following.
coordinated	Coordinated.

6.9.2.3 BWRSteamSupply

Boiling water reactor used as a steam supply to a steam turbine.

Table 318 shows all attributes of BWRSteamSupply.

Table 318 – Attributes of GenerationDynamics::BWRSteamSupply

name	type	description
highPowerLimit	PU	High power limit.
inCoreThermalTC	Seconds	In-core thermal time constant.
integralGain	Float	Integral gain.
lowerLimit	PU	Initial lower limit.
lowPowerLimit	PU	Low power limit.
pressureLimit	PU	Pressure limit.
pressureSetpointGA	Float	Pressure setpoint gain adjuster.
pressureSetpointTC1	Seconds	Pressure setpoint time constant.
pressureSetpointTC2	Seconds	Pressure setpoint time constant.
proportionalGain	Float	Proportional gain.
rfAux1	PU	Coefficient for modeling the effect of off-nominal frequency and voltage on recirculation and core flow, which affects the BWR power output.
rfAux2	PU	Coefficient for modeling the effect of off-nominal frequency and voltage on recirculation and core flow, which affects the BWR power output.
rfAux3	PU	Coefficient for modeling the effect of off-nominal frequency and voltage on recirculation and core flow, which affects the BWR power output.
rfAux4	PU	Coefficient for modeling the effect of off-nominal frequency and voltage on recirculation and core flow, which affects the BWR power output.
rfAux5	PU	Coefficient for modeling the effect of off-nominal frequency and voltage on recirculation and core flow, which affects the BWR power output.

name	type	description
rfAux6	PU	Coefficient for modeling the effect of off-nominal frequency and voltage on recirculation and core flow, which affects the BWR power output.
rfAux7	PU	Coefficient for modeling the effect of off-nominal frequency and voltage on recirculation and core flow, which affects the BWR power output.
rfAux8	PU	Coefficient for modeling the effect of off-nominal frequency and voltage on recirculation and core flow, which affects the BWR power output.
rodPattern	PU	Rod pattern.
rodPatternConstant	Float	Constant associated with rod pattern.
upperLimit	PU	Initial upper limit.
steamSupplyRating	Float	inherited from: SteamSupply
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 319 shows all association ends of BWRSteamSupply with other classes.

Table 319 – Association ends of GenerationDynamics::BWRSteamSupply with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] SteamTurbines	SteamTurbine	inherited from: SteamSupply
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.2.4 CombustionTurbine

A prime mover that is typically fueled by gas or light oil.

Table 320 shows all attributes of CombustionTurbine.

Table 320 – Attributes of GenerationDynamics::CombustionTurbine

name	type	description
ambientTemp	Temperature	Default ambient temperature to be used in modeling applications.
auxPowerVersusFrequency	PU	Off-nominal frequency effect on turbine auxiliaries. Per unit reduction in auxiliary active power consumption versus per unit reduction in frequency (from rated frequency).

name	type	description
auxPowerVersusVoltage	PU	Off-nominal voltage effect on turbine auxiliaries. Per unit reduction in auxiliary active power consumption versus per unit reduction in auxiliary bus voltage (from a specified voltage level).
capabilityVersusFrequency	PU	Off-nominal frequency effect on turbine capability. Per unit reduction in unit active power capability versus per unit reduction in frequency (from rated frequency).
heatRecoveryFlag	Boolean	Flag that is set to true if the combustion turbine is associated with a heat recovery boiler.
powerVariationByTemp	PU	Per unit change in power per (versus) unit change in ambient temperature.
referenceTemp	Temperature	Reference temperature at which the output of the turbine was defined.
timeConstant	Seconds	The time constant for the turbine.
primeMoverRating	Float	inherited from: PrimeMover
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 321 shows all association ends of CombustionTurbine with other classes.

Table 321 – Association ends of GenerationDynamics::CombustionTurbine with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..1] AirCompressor	AirCompressor	A CAES air compressor is driven by combustion turbine.
[0..*]	[0..1] HeatRecoveryBoiler	HeatRecoveryBoiler	A combustion turbine may have a heat recovery boiler for making steam.
[1..1]	[0..1] CTTempActivePowerCurve	CTTempActivePowerCurve	A combustion turbine may have an active power versus ambient temperature relationship.
[0..*]	[0..*] SynchronousMachines	SynchronousMachine	inherited from: PrimeMover
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.2.5 CTTempActivePowerCurve

Relationship between the combustion turbine's power output rating in gross active power (X-axis) and the ambient air temperature (Y-axis).

Table 322 shows all attributes of CTTempActivePowerCurve.

Table 322 – Attributes of GenerationDynamics::CTTempActivePowerCurve

name	type	description
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 323 shows all association ends of CTTempActivePowerCurve with other classes.

Table 323 – Association ends of GenerationDynamics::CTTempActivePowerCurve with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] CombustionTurbine	CombustionTurbine	A combustion turbine may have an active power versus ambient temperature relationship.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.2.6 DrumBoiler

Drum boiler.

Table 324 shows all attributes of DrumBoiler.

Table 324 – Attributes of GenerationDynamics::DrumBoiler

name	type	description
drumBoilerRating	Float	Rating of drum boiler in steam units.
auxPowerVersusFrequency	PU	inherited from: FossilSteamSupply
auxPowerVersusVoltage	PU	inherited from: FossilSteamSupply
boilerControlMode	BoilerControlMode	inherited from: FossilSteamSupply
controlErrorBiasP	Float	inherited from: FossilSteamSupply
controlIC	Float	inherited from: FossilSteamSupply
controlPC	Float	inherited from: FossilSteamSupply
controlPEB	Float	inherited from: FossilSteamSupply
controlPED	PU	inherited from: FossilSteamSupply

name	type	description
controlTC	Float	inherited from: FossilSteamSupply
feedWaterIG	Float	inherited from: FossilSteamSupply
feedWaterPG	Float	inherited from: FossilSteamSupply
feedWaterTC	Seconds	inherited from: FossilSteamSupply
fuelDemandLimit	PU	inherited from: FossilSteamSupply
fuelSupplyDelay	Seconds	inherited from: FossilSteamSupply
fuelSupplyTC	Seconds	inherited from: FossilSteamSupply
maxErrorRateP	Float	inherited from: FossilSteamSupply
mechPowerSensorLag	Seconds	inherited from: FossilSteamSupply
minErrorRateP	Float	inherited from: FossilSteamSupply
pressureCtrlDG	Float	inherited from: FossilSteamSupply
pressureCtrlIG	Float	inherited from: FossilSteamSupply
pressureCtrlPG	Float	inherited from: FossilSteamSupply
pressureFeedback	Integer	inherited from: FossilSteamSupply
superHeater1Capacity	Float	inherited from: FossilSteamSupply
superHeater2Capacity	Float	inherited from: FossilSteamSupply
superHeaterPipePD	Float	inherited from: FossilSteamSupply
throttlePressureSP	PU	inherited from: FossilSteamSupply
steamSupplyRating	Float	inherited from: SteamSupply
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 325 shows all association ends of DrumBoiler with other classes.

Table 325 – Association ends of GenerationDynamics::DrumBoiler with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] SteamTurbines	SteamTurbine	inherited from: SteamSupply
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.2.7 FossilSteamSupply

Fossil fueled boiler (e.g., coal, oil, gas).

Table 326 shows all attributes of FossilSteamSupply.

Table 326 – Attributes of GenerationDynamics::FossilSteamSupply

name	type	description
auxPowerVersusFrequency	PU	Off nominal frequency effect on auxiliary real power. Per unit active power variation versus per unit frequency variation.
auxPowerVersusVoltage	PU	Off nominal voltage effect on auxiliary real power. Per unit active power variation versus per unit voltage variation.
boilerControlMode	BoilerControlMode	The control mode of the boiler.
controlErrorBiasP	Float	Active power error bias ratio.
controlIC	Float	Integral constant.
controlPC	Float	Proportional constant.
controlPEB	Float	Pressure error bias ratio.
controlPED	PU	Pressure error deadband.
controlTC	Float	Time constant.
feedWaterIG	Float	Feedwater integral gain ratio.
feedWaterPG	Float	Feedwater proportional gain ratio.
feedWaterTC	Seconds	Feedwater time constant ratio.
fuelDemandLimit	PU	Fuel demand limit.
fuelSupplyDelay	Seconds	Fuel delay.
fuelSupplyTC	Seconds	Fuel supply time constant.
maxErrorRateP	Float	Active power maximum error rate limit.
mechPowerSensorLag	Seconds	Mechanical power sensor lag.
minErrorRateP	Float	Active power minimum error rate limit.
pressureCtrlDG	Float	Pressure control derivative gain ratio.
pressureCtrlIG	Float	Pressure control integral gain ratio.
pressureCtrlPG	Float	Pressure control proportional gain ratio.
pressureFeedback	Integer	Pressure feedback indicator.
superHeater1Capacity	Float	Drum/primary superheater capacity.
superHeater2Capacity	Float	Secondary superheater capacity.
superHeaterPipePD	Float	Superheater pipe pressure drop constant.
throttlePressureSP	PU	Throttle pressure setpoint.
steamSupplyRating	Float	inherited from: SteamSupply
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 327 shows all association ends of FossilSteamSupply with other classes.

Table 327 – Association ends of GenerationDynamics::FossilSteamSupply with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] SteamTurbines	SteamTurbine	inherited from: SteamSupply
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.2.8 HeatRecoveryBoiler

The heat recovery system associated with combustion turbines in order to produce steam for combined cycle plants.

Table 328 shows all attributes of HeatRecoveryBoiler.

Table 328 – Attributes of GenerationDynamics::HeatRecoveryBoiler

name	type	description
steamSupplyRating2	Float	The steam supply rating in kilopounds per hour, if dual pressure boiler.
auxPowerVersusFrequency	PU	inherited from: FossilSteamSupply
auxPowerVersusVoltage	PU	inherited from: FossilSteamSupply
boilerControlMode	BoilerControlMode	inherited from: FossilSteamSupply
controlErrorBiasP	Float	inherited from: FossilSteamSupply
controlIC	Float	inherited from: FossilSteamSupply
controlPC	Float	inherited from: FossilSteamSupply
controlPEB	Float	inherited from: FossilSteamSupply
controlPED	PU	inherited from: FossilSteamSupply
controlTC	Float	inherited from: FossilSteamSupply
feedWaterIG	Float	inherited from: FossilSteamSupply
feedWaterPG	Float	inherited from: FossilSteamSupply
feedWaterTC	Seconds	inherited from: FossilSteamSupply
fuelDemandLimit	PU	inherited from: FossilSteamSupply
fuelSupplyDelay	Seconds	inherited from: FossilSteamSupply
fuelSupplyTC	Seconds	inherited from: FossilSteamSupply
maxErrorRateP	Float	inherited from: FossilSteamSupply
mechPowerSensorLag	Seconds	inherited from: FossilSteamSupply
minErrorRateP	Float	inherited from: FossilSteamSupply
pressureCtrlDG	Float	inherited from: FossilSteamSupply
pressureCtrlIG	Float	inherited from: FossilSteamSupply
pressureCtrlPG	Float	inherited from: FossilSteamSupply

name	type	description
pressureFeedback	Integer	inherited from: FossilSteamSupply
superHeater1Capacity	Float	inherited from: FossilSteamSupply
superHeater2Capacity	Float	inherited from: FossilSteamSupply
superHeaterPipePD	Float	inherited from: FossilSteamSupply
throttlePressureSP	PU	inherited from: FossilSteamSupply
steamSupplyRating	Float	inherited from: SteamSupply
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 329 shows all association ends of HeatRecoveryBoiler with other classes.

Table 329 – Association ends of GenerationDynamics::HeatRecoveryBoiler with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] CombustionTurbines	CombustionTurbine	A combustion turbine may have a heat recovery boiler for making steam.
[0..*]	[0..*] SteamTurbines	SteamTurbine	inherited from: SteamSupply
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.2.9 HydroTurbine

A water driven prime mover. Typical turbine types are: Francis, Kaplan, and Pelton.

Table 330 shows all attributes of HydroTurbine.

Table 330 – Attributes of GenerationDynamics::HydroTurbine

name	type	description
gateRateLimit	Float	Gate rate limit.
gateUpperLimit	PU	Gate upper limit.
maxHeadMaxP	ActivePower	Maximum efficiency active power at maximum head conditions.
minHeadMaxP	ActivePower	Maximum efficiency active power at minimum head conditions.
speedRating	RotationSpeed	Rated speed in number of revolutions.
speedRegulation	PU	Speed regulation.
transientDroopTime	Seconds	Transient droop time constant.
transientRegulation	PU	Transient regulation.
turbineRating	ActivePower	Rated turbine active power.
turbineType	TurbineType	Type of turbine.
waterStartingTime	Seconds	Water starting time.
primeMoverRating	Float	inherited from: PrimeMover
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 331 shows all association ends of HydroTurbine with other classes.

Table 331 – Association ends of GenerationDynamics::HydroTurbine with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] SynchronousMachines	SynchronousMachine	inherited from: PrimeMover
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.2.10 PrimeMover

The machine used to develop mechanical energy used to drive a generator.

Table 332 shows all attributes of PrimeMover.

Table 332 – Attributes of GenerationDynamics::PrimeMover

name	type	description
primeMoverRating	Float	Rating of prime mover.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 333 shows all association ends of PrimeMover with other classes.

Table 333 – Association ends of GenerationDynamics::PrimeMover with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] SynchronousMachines	SynchronousMachine	Synchronous machines this Prime mover drives.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.2.11 PWRSteamSupply

Pressurized water reactor used as a steam supply to a steam turbine.

Table 334 shows all attributes of PWRSteamSupply.

Table 334 – Attributes of GenerationDynamics::PWRSteamSupply

name	type	description
coldLegFBLagTC	PU	Cold leg feedback lag time constant.
coldLegFBLeadTC1	PU	Cold leg feedback lead time constant.
coldLegFBLeadTC2	PU	Cold leg feedback lead time constant.
coldLegFG1	PU	Cold leg feedback gain 1.
coldLegFG2	PU	Cold leg feedback gain 2.
coldLegLagTC	PU	Cold leg lag time constant.
coreHTLagTC1	PU	Core heat transfer lag time constant.
coreHTLagTC2	PU	Core heat transfer lag time constant.
coreNeutronicsEffTC	PU	Core neutronics effective time constant.
coreNeutronicsHT	PU	Core neutronics and heat transfer.
feedbackFactor	PU	Feedback factor.
hotLegLagTC	PU	Hot leg lag time constant.
hotLegSteamGain	PU	Hot leg steam gain.
hotLegToColdLegGain	PU	Hot leg to cold leg gain.

name	type	description
pressureCG	PU	Pressure control gain.
steamFlowFG	PU	Steam flow feedback gain.
steamPressureDropLagTC	PU	Steam pressure drop lag time constant.
steamPressureFG	PU	Steam pressure feedback gain.
throttlePressureFactor	PU	Throttle pressure factor.
throttlePressureSP	PU	Throttle pressure setpoint.
steamSupplyRating	Float	inherited from: SteamSupply
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 335 shows all association ends of PWRSteamSupply with other classes.

Table 335 – Association ends of GenerationDynamics::PWRSteamSupply with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] SteamTurbines	SteamTurbine	inherited from: SteamSupply
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.2.12 SteamSupply

Steam supply for steam turbine.

Table 336 shows all attributes of SteamSupply.

Table 336 – Attributes of GenerationDynamics::SteamSupply

name	type	description
steamSupplyRating	Float	Rating of steam supply.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 337 shows all association ends of SteamSupply with other classes.

Table 337 – Association ends of GenerationDynamics::SteamSupply with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] SteamTurbines	SteamTurbine	Steam turbines may have steam supplied by a steam supply.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.2.13 SteamTurbine

Steam turbine.

Table 338 shows all attributes of SteamTurbine.

Table 338 – Attributes of GenerationDynamics::SteamTurbine

name	type	description
crossoverTC	Seconds	Crossover time constant.
reheater1TC	Seconds	First reheater time constant.
reheater2TC	Seconds	Second reheater time constant.
shaft1PowerHP	Float	Fraction of power from shaft 1 high pressure turbine output.
shaft1PowerIP	Float	Fraction of power from shaft 1 intermediate pressure turbine output.
shaft1PowerLP1	Float	Fraction of power from shaft 1 first low pressure turbine output.
shaft1PowerLP2	Float	Fraction of power from shaft 1 second low pressure turbine output.
shaft2PowerHP	Float	Fraction of power from shaft 2 high pressure turbine output.
shaft2PowerIP	Float	Fraction of power from shaft 2 intermediate pressure turbine output.
shaft2PowerLP1	Float	Fraction of power from shaft 2 first low pressure turbine output.
shaft2PowerLP2	Float	Fraction of power from shaft 2 second low pressure turbine output.
steamChestTC	Seconds	Steam chest time constant.
primeMoverRating	Float	inherited from: PrimeMover
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 339 shows all association ends of SteamTurbine with other classes.

Table 339 – Association ends of GenerationDynamics::SteamTurbine with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] SteamSupplys	SteamSupply	Steam turbines may have steam supplied by a steam supply.
[0..*]	[0..*] SynchronousMachines	SynchronousMachine	inherited from: PrimeMover
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.2.14 Subcritical

Once-through subcritical boiler.

Table 340 shows all attributes of Subcritical.

Table 340 – Attributes of GenerationDynamics::Subcritical

name	type	description
auxPowerVersusFrequency	PU	inherited from: FossilSteamSupply
auxPowerVersusVoltage	PU	inherited from: FossilSteamSupply
boilerControlMode	BoilerControlMode	inherited from: FossilSteamSupply
controlErrorBiasP	Float	inherited from: FossilSteamSupply
controlIC	Float	inherited from: FossilSteamSupply
controlPC	Float	inherited from: FossilSteamSupply
controlPEB	Float	inherited from: FossilSteamSupply
controlPED	PU	inherited from: FossilSteamSupply
controlTC	Float	inherited from: FossilSteamSupply
feedWaterIG	Float	inherited from: FossilSteamSupply
feedWaterPG	Float	inherited from: FossilSteamSupply
feedWaterTC	Seconds	inherited from: FossilSteamSupply
fuelDemandLimit	PU	inherited from: FossilSteamSupply
fuelSupplyDelay	Seconds	inherited from: FossilSteamSupply
fuelSupplyTC	Seconds	inherited from: FossilSteamSupply
maxErrorRateP	Float	inherited from: FossilSteamSupply
mechPowerSensorLag	Seconds	inherited from: FossilSteamSupply
minErrorRateP	Float	inherited from: FossilSteamSupply
pressureCtrlDG	Float	inherited from: FossilSteamSupply
pressureCtrlIG	Float	inherited from: FossilSteamSupply
pressureCtrlPG	Float	inherited from: FossilSteamSupply
pressureFeedback	Integer	inherited from: FossilSteamSupply
superHeater1Capacity	Float	inherited from: FossilSteamSupply
superHeater2Capacity	Float	inherited from: FossilSteamSupply
superHeaterPipePD	Float	inherited from: FossilSteamSupply
throttlePressureSP	PU	inherited from: FossilSteamSupply
steamSupplyRating	Float	inherited from: SteamSupply
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 341 shows all association ends of Subcritical with other classes.

Table 341 – Association ends of GenerationDynamics::Subcritical with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] SteamTurbines	SteamTurbine	inherited from: SteamSupply
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.2.15 Supercritical

Once-through supercritical boiler.

Table 342 shows all attributes of Supercritical.

Table 342 – Attributes of GenerationDynamics::Supercritical

name	type	description
auxPowerVersusFrequency	PU	inherited from: FossilSteamSupply
auxPowerVersusVoltage	PU	inherited from: FossilSteamSupply
boilerControlMode	BoilerControlMode	inherited from: FossilSteamSupply
controlErrorBiasP	Float	inherited from: FossilSteamSupply
controlIC	Float	inherited from: FossilSteamSupply
controlPC	Float	inherited from: FossilSteamSupply
controlPEB	Float	inherited from: FossilSteamSupply
controlPED	PU	inherited from: FossilSteamSupply
controlTC	Float	inherited from: FossilSteamSupply
feedWaterIG	Float	inherited from: FossilSteamSupply
feedWaterPG	Float	inherited from: FossilSteamSupply
feedWaterTC	Seconds	inherited from: FossilSteamSupply
fuelDemandLimit	PU	inherited from: FossilSteamSupply
fuelSupplyDelay	Seconds	inherited from: FossilSteamSupply
fuelSupplyTC	Seconds	inherited from: FossilSteamSupply
maxErrorRateP	Float	inherited from: FossilSteamSupply
mechPowerSensorLag	Seconds	inherited from: FossilSteamSupply
minErrorRateP	Float	inherited from: FossilSteamSupply
pressureCtrlDG	Float	inherited from: FossilSteamSupply
pressureCtrlIG	Float	inherited from: FossilSteamSupply
pressureCtrlPG	Float	inherited from: FossilSteamSupply
pressureFeedback	Integer	inherited from: FossilSteamSupply

name	type	description
superHeater1Capacity	Float	inherited from: FossilSteamSupply
superHeater2Capacity	Float	inherited from: FossilSteamSupply
superHeaterPipePD	Float	inherited from: FossilSteamSupply
throttlePressureSP	PU	inherited from: FossilSteamSupply
steamSupplyRating	Float	inherited from: SteamSupply
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 343 shows all association ends of Supercritical with other classes.

Table 343 – Association ends of GenerationDynamics::Supercritical with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] SteamTurbines	SteamTurbine	inherited from: SteamSupply
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.2.16 TurbineType enumeration

Type of turbine.

Table 344 shows all literals of TurbineType.

Table 344 – Literals of GenerationDynamics::TurbineType

literal	description
francis	Francis.
pelton	Pelton.
kaplan	Kaplan.

6.9.3 Package Production

6.9.3.1 General

The production package is responsible for classes which describe various kinds of generators. These classes also provide production costing information which is used to economically allocate demand among committed units and calculate reserve quantities.

Figure 63 shows class diagram Nuclear.

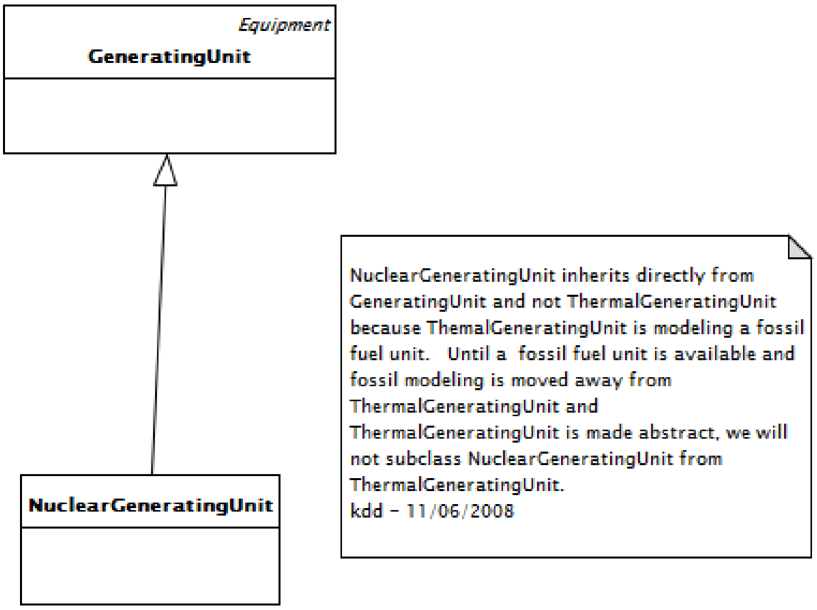

Figure 63 – Class diagram Production::Nuclear

This diagram shows nuclear generation inheritance.

Figure 64 shows class diagram Main.

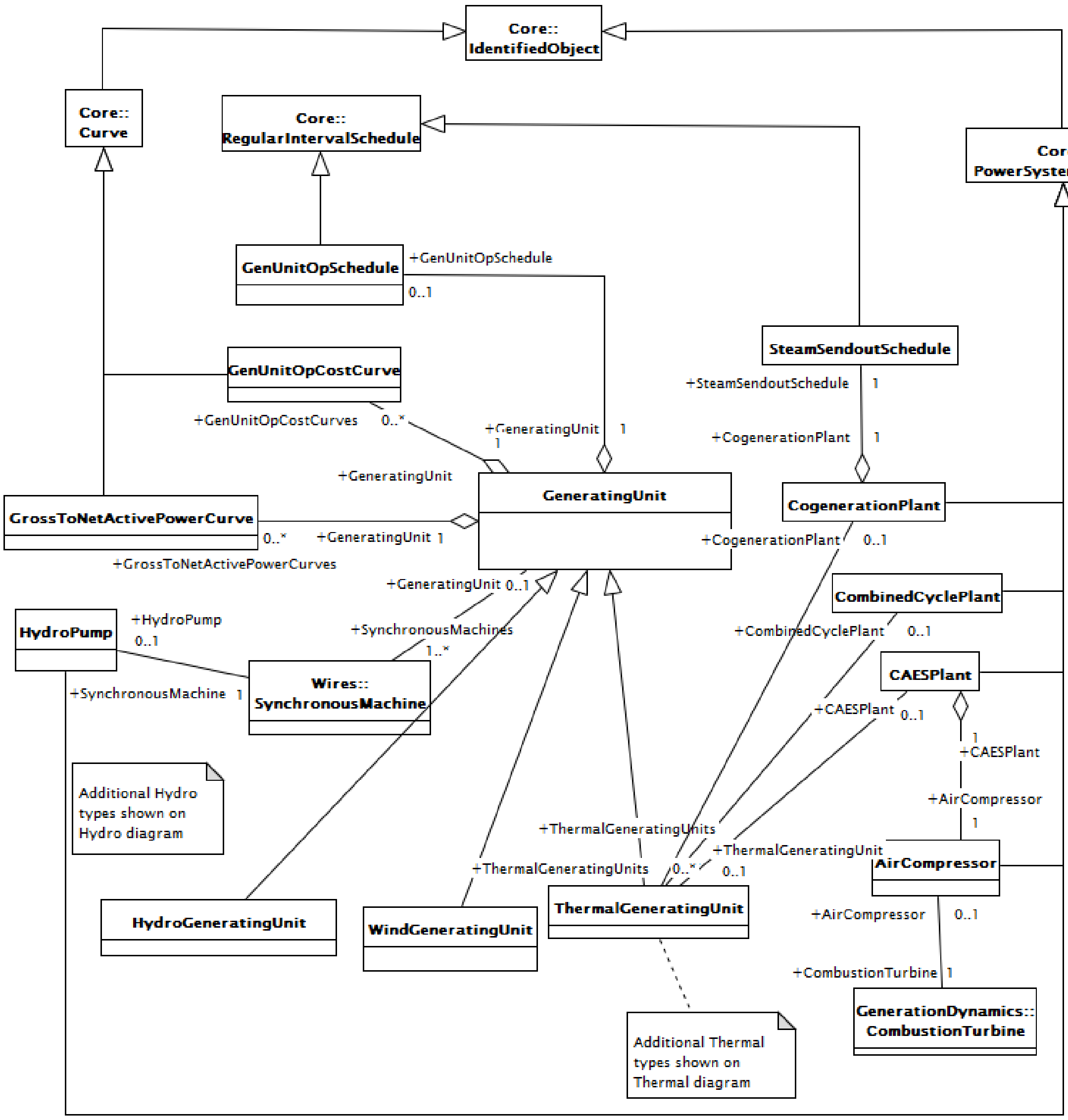

Figure 64 – Class diagram Production::Main

This diagram shows all classes included in the Production package that are needed by both Hydro and Thernal generation. It also shows key external classes that have associations with Production classes.

Figure 65 shows class diagram Datatypes.

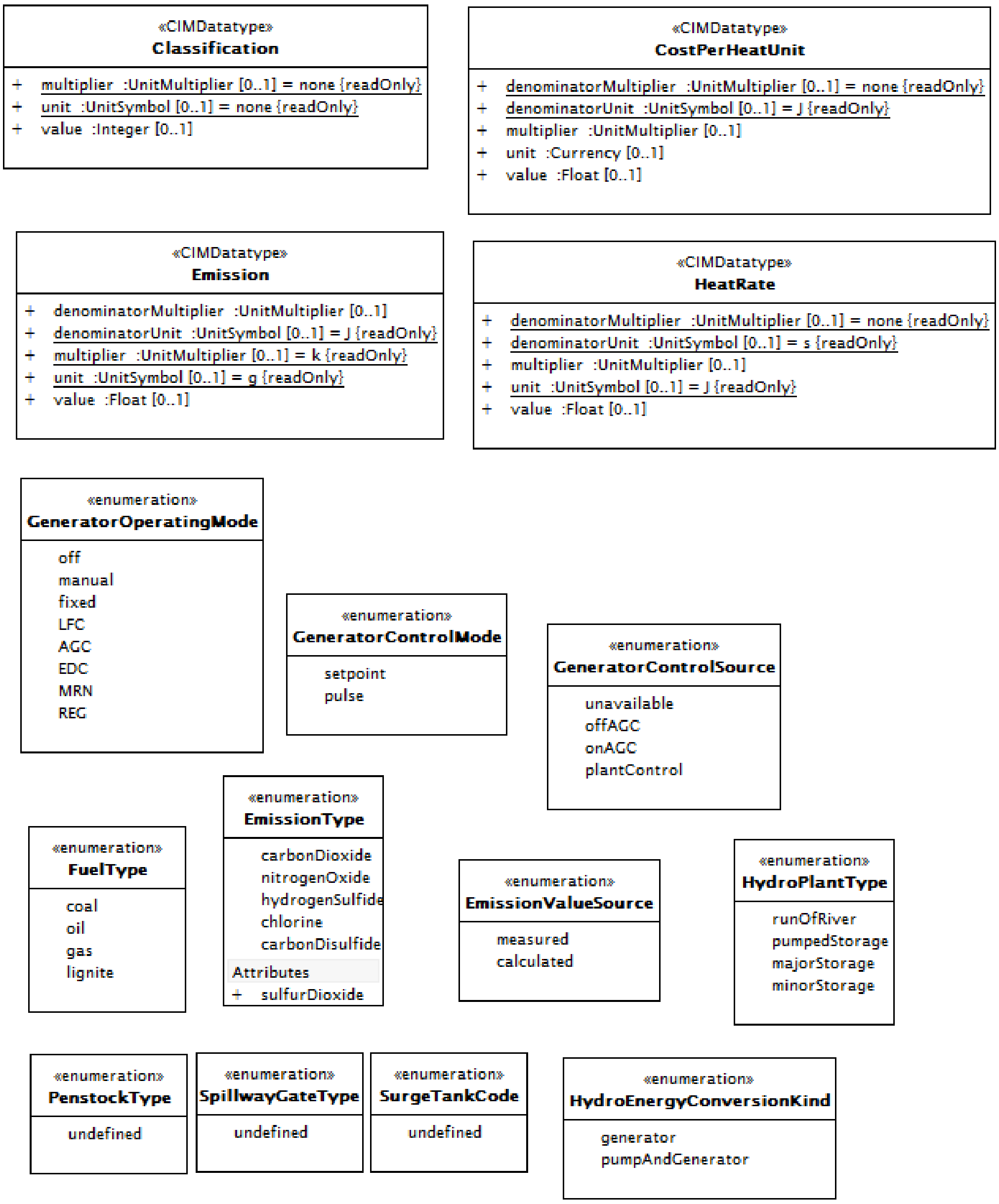

Figure 65 – Class diagram Production::Datatypes

This diagram show the data types specific to the Production package.

Figure 66 shows class diagram Hydro.

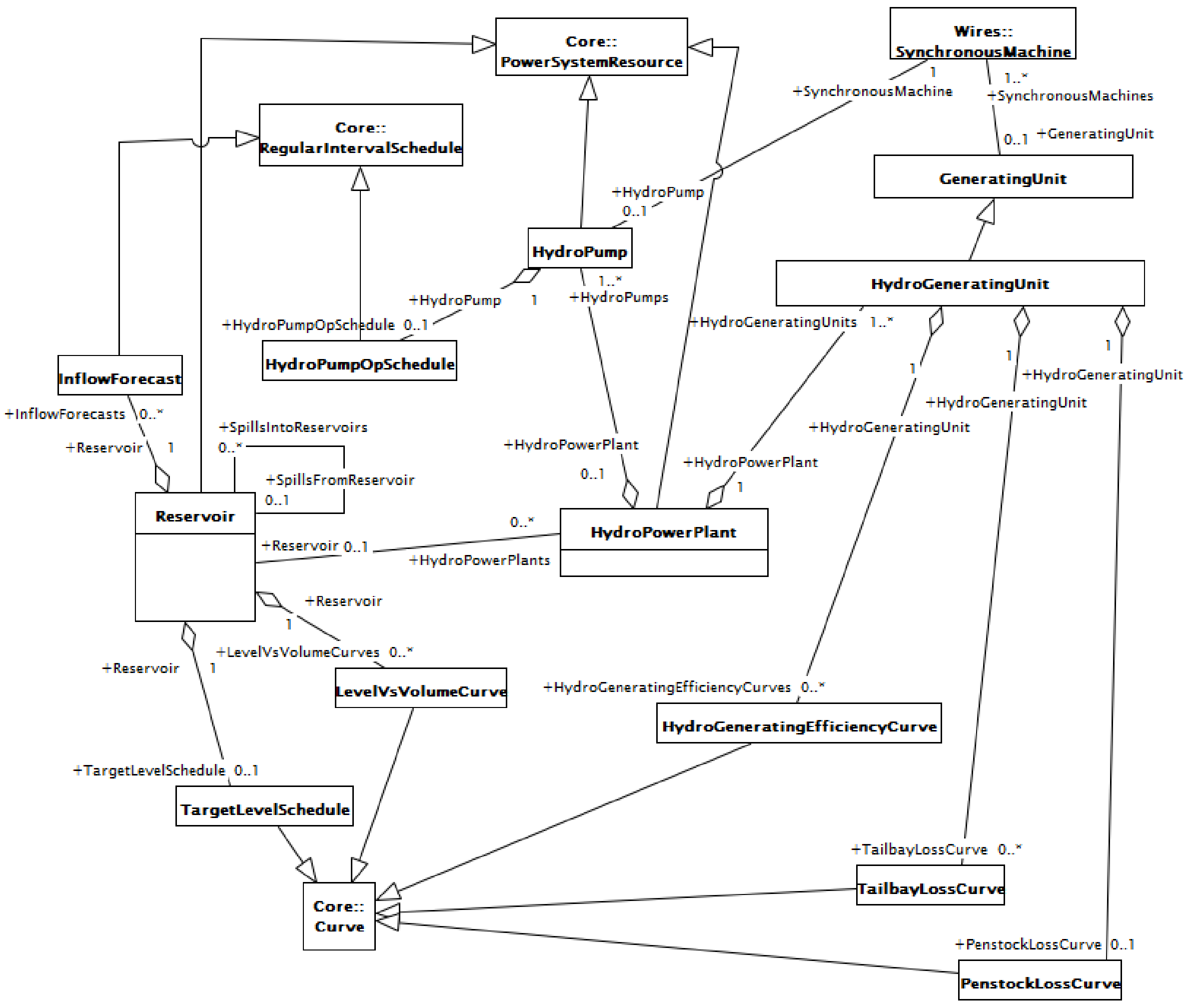

Figure 66 – Class diagram Production::Hydro

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

Figure 67 shows class diagram Thermal.

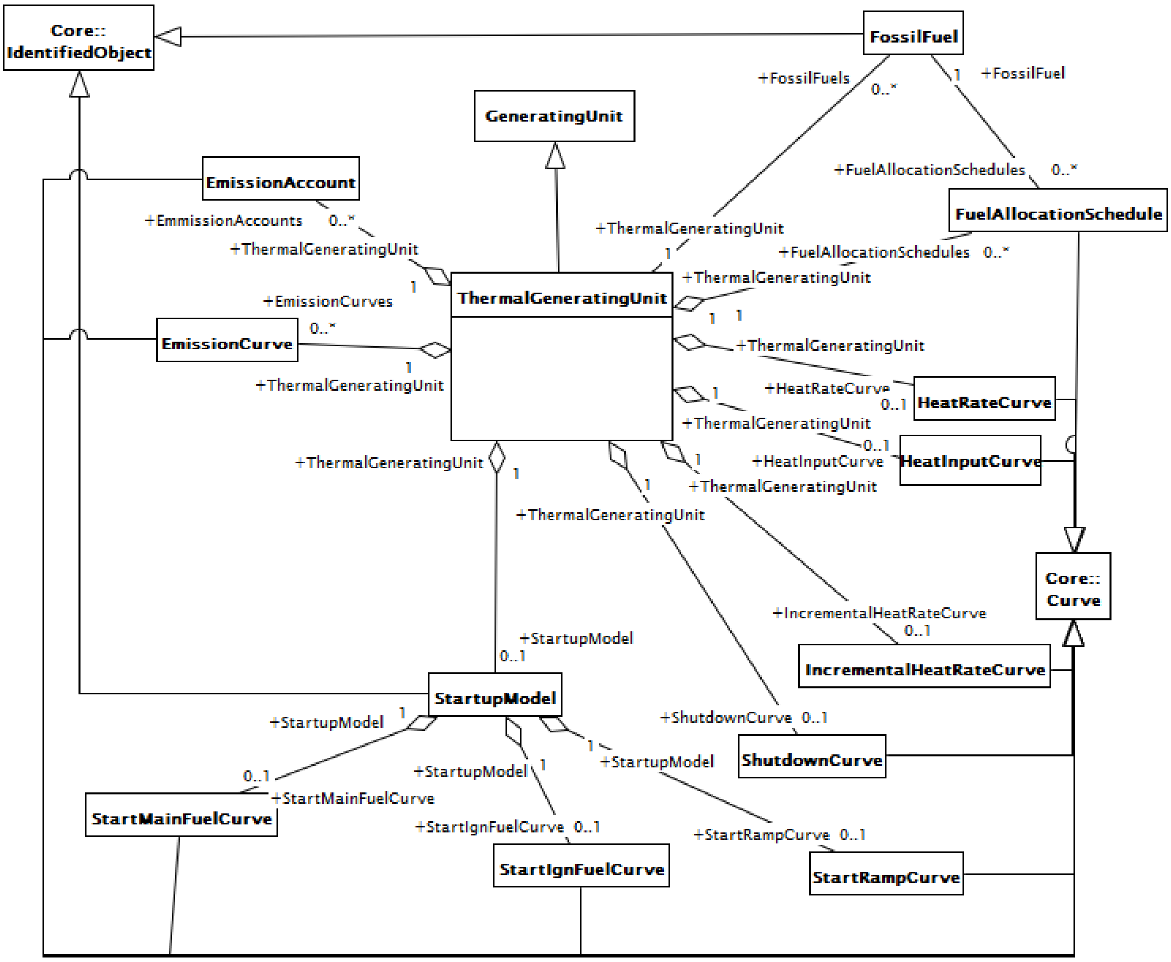

Figure 67 – Class diagram Production::Thermal

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

6.9.3.2 AirCompressor

Combustion turbine air compressor which is an integral part of a compressed air energy storage (CAES) plant.

Table 345 shows all attributes of AirCompressor.

Table 345 – Attributes of Production::AirCompressor

name	type	description
airCompressorRating	Float	Rating of the CAES air compressor.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 346 shows all association ends of AirCompressor with other classes.

Table 346 – Association ends of Production::AirCompressor with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] CombustionTurbine	CombustionTurbine	A CAES air compressor is driven by combustion turbine.
[1..1]	[1..1] CAESPlant	CAESPlant	An air compressor may be a member of a compressed air energy storage plant.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.3 CAESPlant

Compressed air energy storage plant.

Table 347 shows all attributes of CAESPlant.

Table 347 – Attributes of Production::CAESPlant

name	type	description
energyStorageCapacity	RealEnergy	The rated energy storage capacity.
ratedCapacityP	ActivePower	The CAES plant's gross rated generating capacity.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 348 shows all association ends of CAESPlant with other classes.

Table 348 – Association ends of Production::CAESPlant with other classes

[mult from]	[mult to] name	type	description
[1..1]	[1..1] AirCompressor	AirCompressor	An air compressor may be a member of a compressed air energy storage plant.
[0..1]	[0..1] ThermalGeneratingUnit	ThermalGeneratingUnit	A thermal generating unit may be a member of a compressed air energy storage plant.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.4 Classification datatype

Classification of level. Specify as 1..n, with 1 being the most detailed, highest priority, etc as described on the attribue using this data type.

Table 349 shows all attributes of Classification.

Table 349 – Attributes of Production::Classification

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

6.9.3.5 CogenerationPlant

A set of thermal generating units for the production of electrical energy and process steam (usually from the output of the steam turbines). The steam sendout is typically used for industrial purposes or for municipal heating and cooling.

Table 350 shows all attributes of CogenerationPlant.

Table 350 – Attributes of Production::CogenerationPlant

name	type	description
cogenHPSendoutRating	Float	The high pressure steam sendout.
cogenHPSteamRating	Float	The high pressure steam rating.
cogenLPSendoutRating	Float	The low pressure steam sendout.
cogenLPSteamRating	Float	The low pressure steam rating.
ratedP	ActivePower	The rated output active power of the cogeneration plant.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 351 shows all association ends of CogenerationPlant with other classes.

Table 351 – Association ends of Production::CogenerationPlant with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] ThermalGeneratingUnits	ThermalGeneratingUnit	A thermal generating unit may be a member of a cogeneration plant.
[1..1]	[1..1] SteamSendoutSchedule	SteamSendoutSchedule	A cogeneration plant has a steam sendout schedule.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.6 CombinedCyclePlant

A set of combustion turbines and steam turbines where the exhaust heat from the combustion turbines is recovered to make steam for the steam turbines, resulting in greater overall plant efficiency.

Table 352 shows all attributes of CombinedCyclePlant.

Table 352 – Attributes of Production::CombinedCyclePlant

name	type	description
combCyclePlantRating	ActivePower	The combined cycle plant's active power output rating.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 353 shows all association ends of CombinedCyclePlant with other classes.

Table 353 – Association ends of Production::CombinedCyclePlant with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] ThermalGeneratingUnits	ThermalGeneratingUnit	A thermal generating unit may be a member of a combined cycle plant.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.7 CostPerHeatUnit datatype

Cost, in units of currency, per quantity of heat generated.

Table 354 shows all attributes of CostPerHeatUnit.

Table 354 – Attributes of Production::CostPerHeatUnit

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

6.9.3.8 Emission datatype

Quantity of emission per fuel heat content.

Table 355 shows all attributes of Emission.

Table 355 – Attributes of Production::Emission

name	type	description
value	Float	
unit=g (const)	UnitSymbol	
multiplier=k (const)	UnitMultiplier	
denominatorUnit=J (const)	UnitSymbol	
denominatorMultiplier	UnitMultiplier	

6.9.3.9 EmissionAccount

Accounts for tracking emissions usage and credits for thermal generating units. A unit may have zero or more emission accounts, and will typically have one for tracking usage and one for tracking credits.

Table 356 shows all attributes of EmissionAccount.

Table 356 – Attributes of Production::EmissionAccount

name	type	description
emissionType	EmissionType	The type of emission, for example sulfur dioxide (SO2). The y1AxisUnits of the curve contains the unit of measure (e.g. kg) and the emissionType is the type of emission (e.g. sulfer dioxide).
emissionValueSource	EmissionValueSource	The source of the emission value.
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 357 shows all association ends of EmissionAccount with other classes.

Table 357 – Association ends of Production::EmissionAccount with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] ThermalGeneratingUnit	ThermalGeneratingUnit	A thermal generating unit may have one or more emission allowance accounts.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.10 EmissionCurve

Relationship between the unit's emission rate in units of mass per hour (Y-axis) and output active power (X-axis) for a given type of emission. This curve applies when only one type of fuel is being burned.

Table 358 shows all attributes of EmissionCurve.

Table 358 – Attributes of Production::EmissionCurve

name	type	description
emissionContent	Emission	The emission content per quantity of fuel burned.
emissionType	EmissionTyp e	The type of emission, which also gives the production rate measurement unit. The y1AxisUnits of the curve contains the unit of measure (e.g. kg) and the emissionType is the type of emission (e.g. sulfer dioxide).
isNetGrossP	Boolean	Flag is set to true when output is expressed in net active power.
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 359 shows all association ends of EmissionCurve with other classes.

Table 359 – Association ends of Production::EmissionCurve with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] ThermalGeneratingUnit	ThermalGeneratingUnit	A thermal generating unit may have one or more emission curves.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.11 EmissionType enumeration

The type of emission.

Table 360 shows all literals of EmissionType.

Table 360 – Literals of Production::EmissionType

literal	description
sulfurDioxide	Sulfer dioxide.
carbonDioxide	Carbon dioxide.
nitrogenOxide	Nitrogen oxide.
hydrogenSulfide	Hydrogen sulfide.
chlorine	Clorine.
carbonDisulfide	Carbon disulfide.

6.9.3.12 EmissionValueSource enumeration

The source of the emission value.

Table 361 shows all literals of EmissionValueSource.

Table 361 – Literals of Production::EmissionValueSource

literal	description
measured	Measured.
calculated	Calculated.

6.9.3.13 FossilFuel

The fossil fuel consumed by the non-nuclear thermal generating unit. For example, coal, oil, gas, etc. This a the specific fuels that that the generating unit can consume.

Table 362 shows all attributes of FossilFuel.

Table 362 – Attributes of Production::FossilFuel

name	type	description
fossilFuelType	FuelType	The type of fossil fuel, such as coal, oil, or gas.
fuelCost	CostPerHeatUnit	The cost in terms of heat value for the given type of fuel.
fuelDispatchCost	CostPerHeatUnit	The cost of fuel used for economic dispatching which includes: fuel cost, transportation cost, and incremental maintenance cost.
fuelEffFactor	PU	The efficiency factor for the fuel (per unit) in terms of the effective energy absorbed.
fuelHandlingCost	CostPerHeatUnit	Handling and processing cost associated with this fuel.
fuelHeatContent	Float	The amount of heat per weight (or volume) of the given type of fuel.
fuelMixture	PerCent	Relative amount of the given type of fuel, when multiple fuels are being consumed.
fuelSulfur	PU	The fuel's fraction of pollution credit per unit of heat content.
highBreakpointP	ActivePower	The active power output level of the unit at which the given type of fuel is switched on. This fuel (e.g., oil) is sometimes used to supplement the base fuel (e.g., coal) at high active power output levels.
lowBreakpointP	ActivePower	The active power output level of the unit at which the given type of fuel is switched off. This fuel (e.g., oil) is sometimes used to stabilize the base fuel (e.g., coal) at low active power output levels.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 363 shows all association ends of FossilFuel with other classes.

Table 363 – Association ends of Production::FossilFuel with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] FuelAllocationSchedules	FuelAllocationSchedule	A fuel allocation schedule must have a fossil fuel.
[0..*]	[1..1] ThermalGeneratingUnit	ThermalGeneratingUnit	A thermal generating unit may have one or more fossil fuels.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.14 FuelAllocationSchedule

The amount of fuel of a given type which is allocated for consumption over a specified period of time.

Table 364 shows all attributes of FuelAllocationSchedule.

Table 364 – Attributes of Production::FuelAllocationSchedule

name	type	description
fuelAllocationEndDate	DateTime	The end time and date of the fuel allocation schedule.
fuelAllocationStartDate	DateTime	The start time and date of the fuel allocation schedule.
fuelType	FuelType	The type of fuel, which also indicates the corresponding measurement unit.
maxFuelAllocation	Float	The maximum amount fuel that is allocated for consumption for the scheduled time period.
minFuelAllocation	Float	The minimum amount fuel that is allocated for consumption for the scheduled time period, e.g., based on a "take-or-pay" contract.
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 365 shows all association ends of FuelAllocationSchedule with other classes.

Table 365 – Association ends of Production::FuelAllocationSchedule with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] ThermalGeneratingUnit	ThermalGeneratingUnit	A thermal generating unit may have one or more fuel allocation schedules.
[0..*]	[1..1] FossilFuel	FossilFuel	A fuel allocation schedule must have a fossil fuel.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.15 FuelType enumeration

Type of fuel.

Table 366 shows all literals of FuelType.

Table 366 – Literals of Production::FuelType

literal	description
coal	Generic coal, not including lignite type.
oil	Oil.
gas	Gas.
lignite	The fuel is lignite coal. Note that this is a special type of coal, so the other enum of coal is reserved for hard coal types or if the exact type of coal is not known.

6.9.3.16 GeneratingUnit

A single or set of synchronous machines for converting mechanical power into alternating-current power. For example, individual machines within a set may be defined for scheduling purposes while a single control signal is derived for the set. In this case there would be a GeneratingUnit for each member of the set and an additional GeneratingUnit corresponding to the set.

Table 367 shows all attributes of GeneratingUnit.

Table 367 – Attributes of Production::GeneratingUnit

name	type	description
allocSpinResP	ActivePower	The planned unused capacity (spinning reserve) which can be used to support emergency load.
autoCntrlMarginP	ActivePower	The planned unused capacity which can be used to support automatic control overruns.
baseP	ActivePower	For dispatchable units, this value represents the economic active power basepoint, for units that are not dispatchable, this value represents the fixed generation value. The value must be between the operating low and high limits.
controlDeadband	ActivePower	Unit control error deadband. When a unit's desired active power change is less than this deadband, then no control pulses will be sent to the unit.
controlPulseHigh	Seconds	Pulse high limit which is the largest control pulse that the unit can respond to.

name	type	description
controlPulseLow	Seconds	Pulse low limit which is the smallest control pulse that the unit can respond to.
controlResponseRate	ActivePowerChangeRate	Unit response rate which specifies the active power change for a control pulse of one second in the most responsive loading level of the unit.
dispReserveFlag	Boolean	Recommended to be deleted from standard.
efficiency	PU	The efficiency of the unit in converting mechanical energy, from the prime mover, into electrical energy.
energyMinP	HeatRate	Recommended to be deleted from standard.
fastStartFlag	Boolean	Indicates the generating unit has a fast start capability and implies the generating unit can potentially contribute to spinning reserve even while not spinning.
fuelPriority	Integer	Recommended to be deleted from standard.
genControlMode	GeneratorControlMode	The unit control mode.
genControlSource	GeneratorControlSource	The source of controls for a generating unit.
genOperatingMode	GeneratorOperatingMode	Operating mode for secondary control.
governorMPL	PU	Governor motor position limit.
governorSCD	PerCent	Governor Speed Changer Droop. This is the change in generator power output divided by the change in frequency normalized by the nominal power of the generator and the nominal frequency and expressed in percent and negated. A positive value of speed change droop provides additional generator output upon a drop in frequency.
highControlLimit	ActivePower	High limit for secondary (AGC) control.
initialP	ActivePower	Default initial active power which is used to store a powerflow result for the initial active power for this unit in this network configuration.
longPF	Float	Generating unit long term economic participation factor.
lowControlLimit	ActivePower	Low limit for secondary (AGC) control.
lowerRampRate	ActivePowerChangeRate	The normal maximum rate the generating unit active power output can be lowered by control actions.
maxEconomicP	ActivePower	Maximum high economic active power limit, that should not exceed the maximum operating active power limit.
maximumAllowableSpinningReserve	ActivePower	Maximum allowable spinning reserve. Spinning reserve will never be considered greater than this value regardless of the current operating point.
maxOperatingP	ActivePower	This is the maximum operating active power limit the dispatcher can enter for this unit.
minEconomicP	ActivePower	Low economic active power limit that must be greater than or equal to the minimum operating active power limit.
minimumOffTime	Seconds	Minimum time interval between unit shutdown and startup.
minOperatingP	ActivePower	This is the minimum operating active power limit the dispatcher can enter for this unit.
modelDetail	Classification	Detail level of the generator model data.
nominalP	ActivePower	The nominal power of the generating unit. Used to give precise meaning to percentage based attributes such as the govenor speed change droop (govenorSCD attribute).

name	type	description
normalPF	Float	Generating unit economic participation factor.
penaltyFactor	Float	Defined as: 1 / (1 – Incremental Transmission Loss); with the Incremental Transmission Loss expressed as a plus or minus value. The typical range of penalty factors is (0.9 to 1.1).
raiseRampRate	ActivePowerChangeRate	The normal maximum rate the generating unit active power output can be raised by control actions.
ratedGrossMaxP	ActivePower	The unit's gross rated maximum capacity (book value).
ratedGrossMinP	ActivePower	The gross rated minimum generation level which the unit can safely operate at while delivering power to the transmission grid.
ratedNetMaxP	ActivePower	The net rated maximum capacity determined by subtracting the auxiliary power used to operate the internal plant machinery from the rated gross maximum capacity.
shortPF	Float	Generating unit short term economic participation factor.
spinReserveRamp	ActivePowerChangeRate	Reccommended to be deleted from standard.
startupCost	Money	The initial startup cost incurred for each start of the GeneratingUnit.
startupTime	Seconds	Time it takes to get the unit on-line, from the time that the prime mover mechanical power is applied.
stepChange	ActivePower	Recommended to be deleted from standard.
tieLinePF	Float	Generating unit economic participation factor.
variableCost	Money	The variable cost component of production per unit of ActivePower.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 368 shows all association ends of GeneratingUnit with other classes.

Table 368 – Association ends of Production::GeneratingUnit with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..*] SynchronousMachines	SynchronousMachine	A synchronous machine may operate as a generator and as such becomes a member of a generating unit.
[1..1]	[0..1] GenUnitOpSchedule	GenUnitOpSchedule	A generating unit may have an operating schedule, indicating the planned operation of the unit.
[1..1]	[0..*] GrossToNetActivePowerCurves	GrossToNetActivePowerCurve	A generating unit may have a gross active power to net active power curve, describing the losses and auxiliary power requirements of the unit.
[1..1]	[0..*] GenUnitOpCostCurves	GenUnitOpCostCurve	A generating unit may have one or more cost curves, depending upon fuel mixture and fuel cost.
[1..1]	[0..*] ControlAreaGeneratingUnit	ControlAreaGeneratingUnit	ControlArea specifications for this generating unit.
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.17 GeneratorControlMode enumeration

Unit control modes.

Table 369 shows all literals of GeneratorControlMode.

Table 369 – Literals of Production::GeneratorControlMode

literal	description
setpoint	Setpoint control mode.
pulse	Pulse control mode.

6.9.3.18 GeneratorControlSource enumeration

The source of controls for a generating unit.

Table 370 shows all literals of GeneratorControlSource.

Table 370 – Literals of Production::GeneratorControlSource

literal	description
unavailable	Not available.
offAGC	Off of automatic generation control (AGC).
onAGC	On automatic generation control (AGC).
plantControl	Plant is controlling.

6.9.3.19 GeneratorOperatingMode enumeration

Operating mode for secondary generator control.

Table 371 shows all literals of GeneratorOperatingMode.

Table 371 – Literals of Production::GeneratorOperatingMode

literal	description
off	Off.
manual	Manual.
fixed	Fixed.
LFC	Load frequency control.
AGC	Automatic generation control (AGC).
EDC	Economic dispatch control.
MRN	MRN.
REG	REG.

6.9.3.20 GenUnitOpCostCurve

Relationship between unit operating cost (Y-axis) and unit output active power (X-axis). The operating cost curve for thermal units is derived from heat input and fuel costs. The operating cost curve for hydro units is derived from water flow rates and equivalent water costs.

Table 372 shows all attributes of GenUnitOpCostCurve.

Table 372 – Attributes of Production::GenUnitOpCostCurve

name	type	description
isNetGrossP	Boolean	Flag is set to true when output is expressed in net active power.
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 373 shows all association ends of GenUnitOpCostCurve with other classes.

Table 373 – Association ends of Production::GenUnitOpCostCurve with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] GeneratingUnit	GeneratingUnit	A generating unit may have one or more cost curves, depending upon fuel mixture and fuel cost.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.21 GenUnitOpSchedule

The generating unit's Operator-approved current operating schedule (or plan), typically produced with the aid of unit commitment type analyses. The X-axis represents absolute time. The Y1-axis represents the status (0=off-line and unavailable: 1=available: 2=must run: 3=must run at fixed power value: etc.). The Y2-axis represents the must run fixed power value where required.

Table 374 shows all attributes of GenUnitOpSchedule.

Table 374 – Attributes of Production::GenUnitOpSchedule

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

Table 375 shows all association ends of GenUnitOpSchedule with other classes.

Table 375 – Association ends of Production::GenUnitOpSchedule with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] GeneratingUnit	GeneratingUnit	A generating unit may have an operating schedule, indicating the planned operation of the unit.
[1..1]	[1..*] TimePoints	RegularTimePoint	inherited from: RegularIntervalSchedule
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.22 GrossToNetActivePowerCurve

Relationship between the generating unit's gross active power output on the X-axis (measured at the terminals of the machine(s)) and the generating unit's net active power output on the Y-axis (based on utility-defined measurements at the power station). Station service loads, when modeled, should be treated as non-conforming bus loads. There may be more than one curve, depending on the auxiliary equipment that is in service.

Table 376 shows all attributes of GrossToNetActivePowerCurve.

Table 376 – Attributes of Production::GrossToNetActivePowerCurve

name	type	description
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 377 shows all association ends of GrossToNetActivePowerCurve with other classes.

Table 377 – Association ends of Production::GrossToNetActivePowerCurve with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] GeneratingUnit	GeneratingUnit	A generating unit may have a gross active power to net active power curve, describing the losses and auxiliary power requirements of the unit.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.23 HeatInputCurve

Relationship between unit heat input in energy per time for main fuel (Y1-axis) and supplemental fuel (Y2-axis) versus unit output in active power (X-axis). The quantity of main fuel used to sustain generation at this output level is prorated for throttling between definition points. The quantity of supplemental fuel used at this output level is fixed and not prorated.

Table 378 shows all attributes of HeatInputCurve.

Table 378 – Attributes of Production::HeatInputCurve

name	type	description
auxPowerMult	PU	Power output – auxiliary power multiplier adjustment factor.
auxPowerOffset	ActivePower	Power output – auxiliary power offset adjustment factor.
heatInputEff	PU	Heat input – efficiency multiplier adjustment factor.
heatInputOffset	HeatRate	Heat input – offset adjustment factor.
isNetGrossP	Boolean	Flag is set to true when output is expressed in net active power.
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 379 shows all association ends of HeatInputCurve with other classes.

Table 379 – Association ends of Production::HeatInputCurve with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] ThermalGeneratingUnit	ThermalGeneratingUnit	A thermal generating unit may have a heat input curve.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.24 HeatRate datatype

Heat generated, in energy pertime unit of elapsed time.

Table 380 shows all attributes of HeatRate.

Table 380 – Attributes of Production::HeatRate

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

6.9.3.25 HeatRateCurve

Relationship between unit heat rate per active power (Y-axis) and unit output (X-axis). The heat input is from all fuels.

Table 381 shows all attributes of HeatRateCurve.

Table 381 – Attributes of Production::HeatRateCurve

name	type	description
isNetGrossP	Boolean	Flag is set to true when output is expressed in net active power.
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 382 shows all association ends of HeatRateCurve with other classes.

Table 382 – Association ends of Production::HeatRateCurve with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] ThermalGeneratingUnit	ThermalGeneratingUnit	A thermal generating unit may have a heat rate curve.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.26 HydroEnergyConversionKind enumeration

Specifies the capability of the hydro generating unit to convert energy as a generator or pump.

Table 383 shows all literals of HydroEnergyConversionKind.

Table 383 – Literals of Production::HydroEnergyConversionKind

literal	description
generator	Able to generate power, but not able to pump water for energy storage.
pumpAndGenerator	Able to both generate power and pump water for energy storage.

6.9.3.27 HydroGeneratingEfficiencyCurve

Relationship between unit efficiency in percent and unit output active power for a given net head in metres. The relationship between efficiency, discharge, head, and power output is expressed as follows: E =KP/HQ

Where: (E=percentage) (P=active power) (H=height) (Q=volume/time unit) (K=constant)

For example, a curve instance for a given net head could relate efficiency (Y-axis) versus active power output (X-axis) or versus discharge on the X-axis.

Table 384 shows all attributes of HydroGeneratingEfficiencyCurve.

Table 384 – Attributes of Production::HydroGeneratingEfficiencyCurve

name	type	description
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 385 shows all association ends of HydroGeneratingEfficiencyCurve with other classes.

Table 385 – Association ends of Production::HydroGeneratingEfficiencyCurve with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] HydroGeneratingUnit	HydroGeneratingUnit	A hydro generating unit has an efficiency curve.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.28 HydroGeneratingUnit

A generating unit whose prime mover is a hydraulic turbine (e.g., Francis, Pelton, Kaplan).

Table 386 shows all attributes of HydroGeneratingUnit.

Table 386 – Attributes of Production::HydroGeneratingUnit

name	type	description
energyConversionCapability	HydroEnergyConversionKind	Energy conversion capability for generating.
hydroUnitWaterCost	CostPerVolume	The equivalent cost of water that drives the hydro turbine.
allocSpinResP	ActivePower	inherited from: GeneratingUnit
autoCntrlMarginP	ActivePower	inherited from: GeneratingUnit
baseP	ActivePower	inherited from: GeneratingUnit
controlDeadband	ActivePower	inherited from: GeneratingUnit
controlPulseHigh	Seconds	inherited from: GeneratingUnit
controlPulseLow	Seconds	inherited from: GeneratingUnit
controlResponseRate	ActivePowerChangeRate	inherited from: GeneratingUnit
dispReserveFlag	Boolean	inherited from: GeneratingUnit
efficiency	PU	inherited from: GeneratingUnit
energyMinP	HeatRate	inherited from: GeneratingUnit
fastStartFlag	Boolean	inherited from: GeneratingUnit
fuelPriority	Integer	inherited from: GeneratingUnit
genControlMode	GeneratorControlMode	inherited from: GeneratingUnit
genControlSource	GeneratorControlSource	inherited from: GeneratingUnit
genOperatingMode	GeneratorOperatingMode	inherited from: GeneratingUnit
governorMPL	PU	inherited from: GeneratingUnit
governorSCD	PerCent	inherited from: GeneratingUnit
highControlLimit	ActivePower	inherited from: GeneratingUnit
initialP	ActivePower	inherited from: GeneratingUnit
longPF	Float	inherited from: GeneratingUnit
lowControlLimit	ActivePower	inherited from: GeneratingUnit
lowerRampRate	ActivePowerChangeRate	inherited from: GeneratingUnit
maxEconomicP	ActivePower	inherited from: GeneratingUnit
maximumAllowableSpinningReserve	ActivePower	inherited from: GeneratingUnit
maxOperatingP	ActivePower	inherited from: GeneratingUnit
minEconomicP	ActivePower	inherited from: GeneratingUnit
minimumOffTime	Seconds	inherited from: GeneratingUnit
minOperatingP	ActivePower	inherited from: GeneratingUnit
modelDetail	Classification	inherited from: GeneratingUnit
nominalP	ActivePower	inherited from: GeneratingUnit
normalPF	Float	inherited from: GeneratingUnit
penaltyFactor	Float	inherited from: GeneratingUnit
raiseRampRate	ActivePowerChangeRate	inherited from: GeneratingUnit
ratedGrossMaxP	ActivePower	inherited from: GeneratingUnit
ratedGrossMinP	ActivePower	inherited from: GeneratingUnit

name	type	description
ratedNetMaxP	ActivePower	inherited from: GeneratingUnit
shortPF	Float	inherited from: GeneratingUnit
spinReserveRamp	ActivePowerChangeRate	inherited from: GeneratingUnit
startupCost	Money	inherited from: GeneratingUnit
startupTime	Seconds	inherited from: GeneratingUnit
stepChange	ActivePower	inherited from: GeneratingUnit
tieLinePF	Float	inherited from: GeneratingUnit
variableCost	Money	inherited from: GeneratingUnit
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 387 shows all association ends of HydroGeneratingUnit with other classes.

Table 387 – Association ends of Production::HydroGeneratingUnit with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] HydroGeneratingEfficiencyCurves	HydroGeneratingEfficiencyCurve	A hydro generating unit has an efficiency curve.
[1..1]	[0..*] TailbayLossCurve	TailbayLossCurve	A hydro generating unit has a tailbay loss curve.
[1..*]	[1..1] HydroPowerPlant	HydroPowerPlant	The hydro generating unit belongs to a hydro power plant.
[1..1]	[0..1] PenstockLossCurve	PenstockLossCurve	A hydro generating unit has a penstock loss curve.
[0..1]	[1..*] SynchronousMachines	SynchronousMachine	inherited from: GeneratingUnit
[1..1]	[0..1] GenUnitOpSchedule	GenUnitOpSchedule	inherited from: GeneratingUnit
[1..1]	[0..*] GrossToNetActivePowerCurves	GrossToNetActivePowerCurve	inherited from: GeneratingUnit
[1..1]	[0..*] GenUnitOpCostCurves	GenUnitOpCostCurve	inherited from: GeneratingUnit
[1..1]	[0..*] ControlAreaGeneratingUnit	ControlAreaGeneratingUnit	inherited from: GeneratingUnit
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.29 HydroPlantType enumeration

The type of hydro power plant.

Table 388 shows all literals of HydroPlantType.

Table 388 – Literals of Production::HydroPlantType

literal	description
runOfRiver	Run of river.
pumpedStorage	Pumped storage.
majorStorage	Major storage.
minorStorage	Minor storage.

6.9.3.30 HydroPowerPlant

A hydro power station which can generate or pump. When generating, the generator turbines receive there water from an upper reservoir. When pumping, the pumps receive their water from a lower reservoir.

Table 389 shows all attributes of HydroPowerPlant.

Table 389 – Attributes of Production::HydroPowerPlant

name	type	description
dischargeTravelDelay	Seconds	Water travel delay from tailbay to next downstream hydro power station.
genRatedP	ActivePower	The hydro plant's generating rating active power for rated head conditions.
hydroPlantType	HydroPlantType	The type of hydro power plant.
penstockType	PenstockType	Type and configuration of hydro plant penstock(s).
plantDischargeCapacity	VolumeFlowRate	Total plant discharge capacity.
plantRatedHead	Length	The plant's rated gross head.
pumpRatedP	ActivePower	The hydro plant's pumping rating active power for rated head conditions.
surgeTankCode	SurgeTankCode	A code describing the type (or absence) of surge tank that is associated with the hydro power plant.
surgeTankCrestLevel	WaterLevel	The level at which the surge tank spills.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 390 shows all association ends of HydroPowerPlant with other classes.

Table 390 – Association ends of Production::HydroPowerPlant with other classes

[mult from]	[mult to] name	type	description
[1..1]	[1..*] HydroGeneratingUnits	HydroGeneratingUnit	The hydro generating unit belongs to a hydro power plant.
[0..1]	[1..*] HydroPumps	HydroPump	The hydro pump may be a member of a pumped storage plant or a pump for distributing water.
[0..*]	[0..1] Reservoir	Reservoir	Generators discharge water to or pumps are supplied water from a downstream reservoir.
[0..*]	[1..1] GenSourcePumpDischargeReservoir	Reservoir	Generators are supplied water from or pumps discharge water to an upstream reservoir.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.31 HydroPump

A synchronous motor-driven pump, typically associated with a pumped storage plant.

Table 391 shows all attributes of HydroPump.

Table 391 – Attributes of Production::HydroPump

name	type	description
pumpDischAtMaxHead	VolumeFlowRate	The pumping discharge under maximum head conditions, usually at full gate.
pumpDischAtMinHead	VolumeFlowRate	The pumping discharge under minimum head conditions, usually at full gate.
pumpPowerAtMaxHead	ActivePower	The pumping power under maximum head conditions, usually at full gate.
pumpPowerAtMinHead	ActivePower	The pumping power under minimum head conditions, usually at full gate.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 392 shows all association ends of HydroPump with other classes.

Table 392 – Association ends of Production::HydroPump with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] SynchronousMachine	SynchronousMachine	The synchronous machine drives the turbine which moves the water from a low elevation to a higher elevation. The direction of machine rotation for pumping may or may not be the same as for generating.
[1..*]	[0..1] HydroPowerPlant	HydroPowerPlant	The hydro pump may be a member of a pumped storage plant or a pump for distributing water.
[1..1]	[0..1] HydroPumpOpSchedule	HydroPumpOpSchedule	The hydro pump has a pumping schedule over time, indicating when pumping is to occur.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.32 HydroPumpOpSchedule

The hydro pump's Operator-approved current operating schedule (or plan), typically produced with the aid of unit commitment type analyses.The unit's operating schedule status is typically given as: (0=unavailable) (1=available to startup or shutdown) (2=must pump).

Table 393 shows all attributes of HydroPumpOpSchedule.

Table 393 – Attributes of Production::HydroPumpOpSchedule

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

Table 394 shows all association ends of HydroPumpOpSchedule with other classes.

Table 394 – Association ends of Production::HydroPumpOpSchedule with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] HydroPump	HydroPump	The hydro pump has a pumping schedule over time, indicating when pumping is to occur.
[1..1]	[1..*] TimePoints	RegularTimePoint	inherited from: RegularIntervalSchedule
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.33 IncrementalHeatRateCurve

Relationship between unit incremental heat rate in (delta energy/time) per (delta active power) and unit output in active power. The IHR curve represents the slope of the HeatInputCurve. Note that the "incremental heat rate" and the "heat rate" have the same engineering units.

Table 395 shows all attributes of IncrementalHeatRateCurve.

Table 395 – Attributes of Production::IncrementalHeatRateCurve

name	type	description
isNetGrossP	Boolean	Flag is set to true when output is expressed in net active power.
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 396 shows all association ends of IncrementalHeatRateCurve with other classes.

Table 396 – Association ends of Production::IncrementalHeatRateCurve with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] ThermalGeneratingUnit	ThermalGeneratingUnit	A thermal generating unit may have an incremental heat rate curve.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.34 InflowForecast

Natural water inflow to a reservoir, usually forecasted from predicted rain and snowmelt. Typically in one hour increments for up to 10 days. The forecast is given in average cubic metres per second over the time increment.

Table 397 shows all attributes of InflowForecast.

Table 397 – Attributes of Production::InflowForecast

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

Table 398 shows all association ends of InflowForecast with other classes.

Table 398 – Association ends of Production::InflowForecast with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Reservoir	Reservoir	A reservoir may have a "natural" inflow forecast.
[1..1]	[1..*] TimePoints	RegularTimePoint	inherited from: RegularIntervalSchedule
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.35 LevelVsVolumeCurve

Relationship between reservoir volume and reservoir level. The volume is at the Y-axis and the reservoir level at the X-axis.

Table 399 shows all attributes of LevelVsVolumeCurve.

Table 399 – Attributes of Production::LevelVsVolumeCurve

name	type	description
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 400 shows all association ends of LevelVsVolumeCurve with other classes.

Table 400 – Association ends of Production::LevelVsVolumeCurve with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Reservoir	Reservoir	A reservoir may have a level versus volume relationship.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.36 NuclearGeneratingUnit

A nuclear generating unit.

Table 401 shows all attributes of NuclearGeneratingUnit.

Table 401 – Attributes of Production::NuclearGeneratingUnit

name	type	description
allocSpinResP	ActivePower	inherited from: GeneratingUnit
autoCntrlMarginP	ActivePower	inherited from: GeneratingUnit
baseP	ActivePower	inherited from: GeneratingUnit
controlDeadband	ActivePower	inherited from: GeneratingUnit
controlPulseHigh	Seconds	inherited from: GeneratingUnit
controlPulseLow	Seconds	inherited from: GeneratingUnit
controlResponseRate	ActivePowerChangeRate	inherited from: GeneratingUnit
dispReserveFlag	Boolean	inherited from: GeneratingUnit
efficiency	PU	inherited from: GeneratingUnit
energyMinP	HeatRate	inherited from: GeneratingUnit
fastStartFlag	Boolean	inherited from: GeneratingUnit
fuelPriority	Integer	inherited from: GeneratingUnit

name	type	description
genControlMode	GeneratorControlMode	inherited from: GeneratingUnit
genControlSource	GeneratorControlSource	inherited from: GeneratingUnit
genOperatingMode	GeneratorOperatingMode	inherited from: GeneratingUnit
governorMPL	PU	inherited from: GeneratingUnit
governorSCD	PerCent	inherited from: GeneratingUnit
highControlLimit	ActivePower	inherited from: GeneratingUnit
initialP	ActivePower	inherited from: GeneratingUnit
longPF	Float	inherited from: GeneratingUnit
lowControlLimit	ActivePower	inherited from: GeneratingUnit
lowerRampRate	ActivePowerChangeRate	inherited from: GeneratingUnit
maxEconomicP	ActivePower	inherited from: GeneratingUnit
maximumAllowableSpinningReserve	ActivePower	inherited from: GeneratingUnit
maxOperatingP	ActivePower	inherited from: GeneratingUnit
minEconomicP	ActivePower	inherited from: GeneratingUnit
minimumOffTime	Seconds	inherited from: GeneratingUnit
minOperatingP	ActivePower	inherited from: GeneratingUnit
modelDetail	Classification	inherited from: GeneratingUnit
nominalP	ActivePower	inherited from: GeneratingUnit
normalPF	Float	inherited from: GeneratingUnit
penaltyFactor	Float	inherited from: GeneratingUnit
raiseRampRate	ActivePowerChangeRate	inherited from: GeneratingUnit
ratedGrossMaxP	ActivePower	inherited from: GeneratingUnit
ratedGrossMinP	ActivePower	inherited from: GeneratingUnit
ratedNetMaxP	ActivePower	inherited from: GeneratingUnit
shortPF	Float	inherited from: GeneratingUnit
spinReserveRamp	ActivePowerChangeRate	inherited from: GeneratingUnit
startupCost	Money	inherited from: GeneratingUnit
startupTime	Seconds	inherited from: GeneratingUnit
stepChange	ActivePower	inherited from: GeneratingUnit
tieLinePF	Float	inherited from: GeneratingUnit
variableCost	Money	inherited from: GeneratingUnit
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 402 shows all association ends of NuclearGeneratingUnit with other classes.

Table 402 – Association ends of Production::NuclearGeneratingUnit with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..*] SynchronousMachines	SynchronousMachine	inherited from: GeneratingUnit
[1..1]	[0..1] GenUnitOpSchedule	GenUnitOpSchedule	inherited from: GeneratingUnit
[1..1]	[0..*] GrossToNetActivePowerCurves	GrossToNetActivePowerCurve	inherited from: GeneratingUnit
[1..1]	[0..*] GenUnitOpCostCurves	GenUnitOpCostCurve	inherited from: GeneratingUnit
[1..1]	[0..*] ControlAreaGeneratingUnit	ControlAreaGeneratingUnit	inherited from: GeneratingUnit
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.37 PenstockLossCurve

Relationship between penstock head loss (in metres) and total discharge through the penstock (in cubic metres per second). One or more turbines may be connected to the same penstock.

Table 403 shows all attributes of PenstockLossCurve.

Table 403 – Attributes of Production::PenstockLossCurve

name	type	description
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 404 shows all association ends of PenstockLossCurve with other classes.

Table 404 – Association ends of Production::PenstockLossCurve with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] HydroGeneratingUnit	HydroGeneratingUnit	A hydro generating unit has a penstock loss curve.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.38 PenstockType enumeration

Type of hydro plant penstock.

Table 405 shows all literals of PenstockType.

Table 405 – Literals of Production::PenstockType

literal	description
undefined	The penstock type is undefined.

6.9.3.39 Reservoir

A water storage facility within a hydro system, including: ponds, lakes, lagoons, and rivers. The storage is usually behind some type of dam.

Table 406 shows all attributes of Reservoir.

Table 406 – Attributes of Production::Reservoir

name	type	description
activeStorageCapacity	Volume	Storage volume between the full supply level and the normal minimum operating level.
energyStorageRating	Float	The reservoir's energy storage rating in energy for given head conditions.
fullSupplyLevel	WaterLevel	Full supply level, above which water will spill. This can be the spillway crest level or the top of closed gates.
grossCapacity	Volume	Total capacity of reservoir.
normalMinOperateLevel	WaterLevel	Normal minimum operating level below which the penstocks will draw air.
riverOutletWorks	String	River outlet works for riparian right releases or other purposes.
spillTravelDelay	Seconds	The spillway water travel delay to the next downstream reservoir.
spillwayCapacity	Float	The flow capacity of the spillway in cubic metres per second.
spillwayCrestLength	Length	The length of the spillway crest.
spillwayCrestLevel	WaterLevel	Spillway crest level above which water will spill.
spillWayGateType	SpillwayGateType	Type of spillway gate, including parameters.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 407 shows all association ends of Reservoir with other classes.

Table 407 – Association ends of Production::Reservoir with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] SpillsIntoReservoirs	Reservoir	A reservoir may spill into a downstream reservoir.
[0..1]	[0..*] HydroPowerPlants	HydroPowerPlant	Generators discharge water to or pumps are supplied water from a downstream reservoir.
[1..1]	[0..*] UpstreamFromHydroPowerPlants	HydroPowerPlant	Generators are supplied water from or pumps discharge water to an upstream reservoir.
[1..1]	[0..*] InflowForecasts	InflowForecast	A reservoir may have a "natural" inflow forecast.
[1..1]	[0..*] LevelVsVolumeCurves	LevelVsVolumeCurve	A reservoir may have a level versus volume relationship.
[1..1]	[0..1] TargetLevelSchedule	TargetLevelSchedule	A reservoir may have a water level target schedule.
[0..*]	[0..1] SpillsFromReservoir	Reservoir	A reservoir may spill into a downstream reservoir.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.40 ShutdownCurve

Relationship between the rate in gross active power/minute (Y-axis) at which a unit should be shutdown and its present gross MW output (X-axis).

Table 408 shows all attributes of ShutdownCurve.

Table 408 – Attributes of Production::ShutdownCurve

name	type	description
shutdownCost	Money	Fixed shutdown cost.
shutdownDate	DateTime	The date and time of the most recent generating unit shutdown.
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 409 shows all association ends of ShutdownCurve with other classes.

Table 409 – Association ends of Production::ShutdownCurve with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] ThermalGeneratingUnit	ThermalGeneratingUnit	A thermal generating unit may have a shutdown curve.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.41 SpillwayGateType enumeration

Type of spillway gate.

Table 410 shows all literals of SpillwayGateType.

Table 410 – Literals of Production::SpillwayGateType

literal	description
undefined	The type is spillway gate type is undefined.

6.9.3.42 StartIgnFuelCurve

The quantity of ignition fuel (Y-axis) used to restart and repay the auxiliary power consumed versus the number of hours (X-axis) the unit was off line.

Table 411 shows all attributes of StartIgnFuelCurve.

Table 411 – Attributes of Production::StartIgnFuelCurve

name	type	description
ignitionFuelType	FuelType	Type of ignition fuel.
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 412 shows all association ends of StartIgnFuelCurve with other classes.

Table 412 – Association ends of Production::StartIgnFuelCurve with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] StartupModel	StartupModel	The unit's startup model may have a startup ignition fuel curve.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.43 StartMainFuelCurve

The quantity of main fuel (Y-axis) used to restart and repay the auxiliary power consumed versus the number of hours (X-axis) the unit was off line.

Table 413 shows all attributes of StartMainFuelCurve.

Table 413 – Attributes of Production::StartMainFuelCurve

name	type	description
mainFuelType	FuelType	Type of main fuel.
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 414 shows all association ends of StartMainFuelCurve with other classes.

Table 414 – Association ends of Production::StartMainFuelCurve with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] StartupModel	StartupModel	The unit's startup model may have a startup main fuel curve.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.44 StartRampCurve

Rate in gross active power/minute (Y-axis) at which a unit can be loaded versus the number of hours (X-axis) the unit was off line.

Table 415 shows all attributes of StartRampCurve.

Table 415 – Attributes of Production::StartRampCurve

name	type	description
hotStandbyRamp	ActivePowerChangeRate	The startup ramp rate in gross for a unit that is on hot standby.
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 416 shows all association ends of StartRampCurve with other classes.

Table 416 – Association ends of Production::StartRampCurve with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] StartupModel	StartupModel	The unit's startup model may have a startup ramp curve.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.45 StartupModel

Unit start up characteristics depending on how long the unit has been off line.

Table 417 shows all attributes of StartupModel.

Table 417 – Attributes of Production::StartupModel

name	type	description
fixedMaintCost	CostRate	Fixed maintenance cost.
hotStandbyHeat	HeatRate	The amount of heat input per time unit required for hot standby operation.
incrementalMaintCost	CostPerEnergyUnit	Incremental maintenance cost.
minimumDownTime	Hours	The minimum number of hours the unit must be down before restart.
minimumRunTime	Hours	The minimum number of hours the unit must be operating before being allowed to shut down.
riskFactorCost	Money	The opportunity cost associated with the return in monetary unit. This represents the restart's "share" of the unit depreciation and risk of an event which would damage the unit.
startupCost	Money	Total miscellaneous start up costs.

name	type	description
startupDate	DateTime	The date and time of the most recent generating unit startup.
startupPriority	Integer	Startup priority within control area where lower numbers indicate higher priorities. More than one unit in an area may be assigned the same priority.
stbyAuxP	ActivePower	The unit's auxiliary active power consumption to maintain standby mode.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 418 shows all association ends of StartupModel with other classes.

Table 418 – Association ends of Production::StartupModel with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] ThermalGeneratingUnit	ThermalGeneratingUnit	A thermal generating unit may have a startup model.
[1..1]	[0..1] StartIgnFuelCurve	StartIgnFuelCurve	The unit's startup model may have a startup ignition fuel curve.
[1..1]	[0..1] StartMainFuelCurve	StartMainFuelCurve	The unit's startup model may have a startup main fuel curve.
[1..1]	[0..1] StartRampCurve	StartRampCurve	The unit's startup model may have a startup ramp curve.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.46 SteamSendoutSchedule

The cogeneration plant's steam sendout schedule in volume per time unit.

Table 419 shows all attributes of SteamSendoutSchedule.

Table 419 – Attributes of Production::SteamSendoutSchedule

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

Table 420 shows all association ends of SteamSendoutSchedule with other classes.

Table 420 – Association ends of Production::SteamSendoutSchedule with other classes

[mult from]	[mult to] name	type	description
[1..1]	[1..1] CogenerationPlant	CogenerationPlant	A cogeneration plant has a steam sendout schedule.
[1..1]	[1..*] TimePoints	RegularTimePoint	inherited from: RegularIntervalSchedule
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.47 SurgeTankCode enumeration

Type (or absence) of surge tank that is associated with the hydro power plant.

Table 421 shows all literals of SurgeTankCode.

Table 421 – Literals of Production::SurgeTankCode

literal	description
undefined	The surge tank code is undefined.

6.9.3.48 TailbayLossCurve

Relationship between tailbay head loss hight (Y-axis) and the total discharge into the power station's tailbay volume per time unit (X-axis) . There could be more than one curve depending on the level of the tailbay reservoir or river level.

Table 422 shows all attributes of TailbayLossCurve.

Table 422 – Attributes of Production::TailbayLossCurve

name	type	description
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 423 shows all association ends of TailbayLossCurve with other classes.

Table 423 – Association ends of Production::TailbayLossCurve with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] HydroGeneratingUnit	HydroGeneratingUnit	A hydro generating unit has a tailbay loss curve.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.49 TargetLevelSchedule

Reservoir water level targets from advanced studies or "rule curves". Typically in one hour increments for up to 10 days.

Table 424 shows all attributes of TargetLevelSchedule.

Table 424 – Attributes of Production::TargetLevelSchedule

name	type	description
highLevelLimit	WaterLevel	High target level limit, above which the reservoir operation will be penalized.
lowLevelLimit	WaterLevel	Low target level limit, below which the reservoir operation will be penalized.
curveStyle	CurveStyle	inherited from: Curve
xMultiplier	UnitMultiplier	inherited from: Curve
xUnit	UnitSymbol	inherited from: Curve
y1Multiplier	UnitMultiplier	inherited from: Curve
y1Unit	UnitSymbol	inherited from: Curve
y2Multiplier	UnitMultiplier	inherited from: Curve
y2Unit	UnitSymbol	inherited from: Curve
y3Multiplier	UnitMultiplier	inherited from: Curve
y3Unit	UnitSymbol	inherited from: Curve
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 425 shows all association ends of TargetLevelSchedule with other classes.

Table 425 – Association ends of Production::TargetLevelSchedule with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] Reservoir	Reservoir	A reservoir may have a water level target schedule.
[1..1]	[0..*] CurveDatas	CurveData	inherited from: Curve
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.50 ThermalGeneratingUnit

A generating unit whose prime mover could be a steam turbine, combustion turbine, or diesel engine.

Table 426 shows all attributes of ThermalGeneratingUnit.

Table 426 – Attributes of Production::ThermalGeneratingUnit

name	type	description
oMCost	CostPerHeatUnit	Operating and maintenance cost for the thermal unit.
allocSpinResP	ActivePower	inherited from: GeneratingUnit
autoCntrlMarginP	ActivePower	inherited from: GeneratingUnit
baseP	ActivePower	inherited from: GeneratingUnit
controlDeadband	ActivePower	inherited from: GeneratingUnit
controlPulseHigh	Seconds	inherited from: GeneratingUnit
controlPulseLow	Seconds	inherited from: GeneratingUnit
controlResponseRate	ActivePowerChangeRate	inherited from: GeneratingUnit
dispReserveFlag	Boolean	inherited from: GeneratingUnit
efficiency	PU	inherited from: GeneratingUnit
energyMinP	HeatRate	inherited from: GeneratingUnit
fastStartFlag	Boolean	inherited from: GeneratingUnit
fuelPriority	Integer	inherited from: GeneratingUnit
genControlMode	GeneratorControlMode	inherited from: GeneratingUnit
genControlSource	GeneratorControlSource	inherited from: GeneratingUnit
genOperatingMode	GeneratorOperatingMode	inherited from: GeneratingUnit
governorMPL	PU	inherited from: GeneratingUnit
governorSCD	PerCent	inherited from: GeneratingUnit
highControlLimit	ActivePower	inherited from: GeneratingUnit
initialP	ActivePower	inherited from: GeneratingUnit
longPF	Float	inherited from: GeneratingUnit
lowControlLimit	ActivePower	inherited from: GeneratingUnit
lowerRampRate	ActivePowerChangeRate	inherited from: GeneratingUnit
maxEconomicP	ActivePower	inherited from: GeneratingUnit
maximumAllowableSpinningReserve	ActivePower	inherited from: GeneratingUnit
maxOperatingP	ActivePower	inherited from: GeneratingUnit
minEconomicP	ActivePower	inherited from: GeneratingUnit
minimumOffTime	Seconds	inherited from: GeneratingUnit
minOperatingP	ActivePower	inherited from: GeneratingUnit
modelDetail	Classification	inherited from: GeneratingUnit
nominalP	ActivePower	inherited from: GeneratingUnit
normalPF	Float	inherited from: GeneratingUnit
penaltyFactor	Float	inherited from: GeneratingUnit
raiseRampRate	ActivePowerChangeRate	inherited from: GeneratingUnit
ratedGrossMaxP	ActivePower	inherited from: GeneratingUnit
ratedGrossMinP	ActivePower	inherited from: GeneratingUnit

name	type	description
ratedNetMaxP	ActivePower	inherited from: GeneratingUnit
shortPF	Float	inherited from: GeneratingUnit
spinReserveRamp	ActivePowerChangeRate	inherited from: GeneratingUnit
startupCost	Money	inherited from: GeneratingUnit
startupTime	Seconds	inherited from: GeneratingUnit
stepChange	ActivePower	inherited from: GeneratingUnit
tieLinePF	Float	inherited from: GeneratingUnit
variableCost	Money	inherited from: GeneratingUnit
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 427 shows all association ends of ThermalGeneratingUnit with other classes.

Table 427 – Association ends of Production::ThermalGeneratingUnit with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..1] CAESPlant	CAESPlant	A thermal generating unit may be a member of a compressed air energy storage plant.
[0..*]	[0..1] CogenerationPlant	CogenerationPlant	A thermal generating unit may be a member of a cogeneration plant.
[0..*]	[0..1] CombinedCyclePlant	CombinedCyclePlant	A thermal generating unit may be a member of a combined cycle plant.
[1..1]	[0..*] EmmissionAccounts	EmissionAccount	A thermal generating unit may have one or more emission allowance accounts.
[1..1]	[0..*] EmissionCurves	EmissionCurve	A thermal generating unit may have one or more emission curves.
[1..1]	[0..*] FossilFuels	FossilFuel	A thermal generating unit may have one or more fossil fuels.
[1..1]	[0..*] FuelAllocationSchedules	FuelAllocationSchedule	A thermal generating unit may have one or more fuel allocation schedules.
[1..1]	[0..1] HeatInputCurve	HeatInputCurve	A thermal generating unit may have a heat input curve.
[1..1]	[0..1] HeatRateCurve	HeatRateCurve	A thermal generating unit may have a heat rate curve.
[1..1]	[0..1] IncrementalHeatRateCurve	IncrementalHeatRateCurve	A thermal generating unit may have an incremental heat rate curve.
[1..1]	[0..1] ShutdownCurve	ShutdownCurve	A thermal generating unit may have a shutdown curve.
[1..1]	[0..1] StartupModel	StartupModel	A thermal generating unit may have a startup model.
[0..1]	[1..*] SynchronousMachines	SynchronousMachine	inherited from: GeneratingUnit
[1..1]	[0..1] GenUnitOpSchedule	GenUnitOpSchedule	inherited from: GeneratingUnit
[1..1]	[0..*] GrossToNetActivePowerCurves	GrossToNetActivePowerCurve	inherited from: GeneratingUnit

[mult from]	[mult to] name	type	description
[1..1]	[0..*] GenUnitOpCostCurves	GenUnitOpCostCurve	inherited from: GeneratingUnit
[1..1]	[0..*] ControlAreaGeneratingUnit	ControlAreaGeneratingUnit	inherited from: GeneratingUnit
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.9.3.51 WindGeneratingUnit

A wind driven generating unit.

Table 428 shows all attributes of WindGeneratingUnit.

Table 428 – Attributes of Production::WindGeneratingUnit

name	type	description
allocSpinResP	ActivePower	inherited from: GeneratingUnit
autoCntrlMarginP	ActivePower	inherited from: GeneratingUnit
baseP	ActivePower	inherited from: GeneratingUnit
controlDeadband	ActivePower	inherited from: GeneratingUnit
controlPulseHigh	Seconds	inherited from: GeneratingUnit
controlPulseLow	Seconds	inherited from: GeneratingUnit
controlResponseRate	ActivePowerChangeRate	inherited from: GeneratingUnit
dispReserveFlag	Boolean	inherited from: GeneratingUnit
efficiency	PU	inherited from: GeneratingUnit
energyMinP	HeatRate	inherited from: GeneratingUnit
fastStartFlag	Boolean	inherited from: GeneratingUnit
fuelPriority	Integer	inherited from: GeneratingUnit
genControlMode	GeneratorControlMode	inherited from: GeneratingUnit
genControlSource	GeneratorControlSource	inherited from: GeneratingUnit
genOperatingMode	GeneratorOperatingMode	inherited from: GeneratingUnit
governorMPL	PU	inherited from: GeneratingUnit
governorSCD	PerCent	inherited from: GeneratingUnit
highControlLimit	ActivePower	inherited from: GeneratingUnit
initialP	ActivePower	inherited from: GeneratingUnit
longPF	Float	inherited from: GeneratingUnit
lowControlLimit	ActivePower	inherited from: GeneratingUnit

name	type	description
lowerRampRate	ActivePowerChangeRate	inherited from: GeneratingUnit
maxEconomicP	ActivePower	inherited from: GeneratingUnit
maximumAllowableSpinningReserve	ActivePower	inherited from: GeneratingUnit
maxOperatingP	ActivePower	inherited from: GeneratingUnit
minEconomicP	ActivePower	inherited from: GeneratingUnit
minimumOffTime	Seconds	inherited from: GeneratingUnit
minOperatingP	ActivePower	inherited from: GeneratingUnit
modelDetail	Classification	inherited from: GeneratingUnit
nominalP	ActivePower	inherited from: GeneratingUnit
normalPF	Float	inherited from: GeneratingUnit
penaltyFactor	Float	inherited from: GeneratingUnit
raiseRampRate	ActivePowerChangeRate	inherited from: GeneratingUnit
ratedGrossMaxP	ActivePower	inherited from: GeneratingUnit
ratedGrossMinP	ActivePower	inherited from: GeneratingUnit
ratedNetMaxP	ActivePower	inherited from: GeneratingUnit
shortPF	Float	inherited from: GeneratingUnit
spinReserveRamp	ActivePowerChangeRate	inherited from: GeneratingUnit
startupCost	Money	inherited from: GeneratingUnit
startupTime	Seconds	inherited from: GeneratingUnit
stepChange	ActivePower	inherited from: GeneratingUnit
tieLinePF	Float	inherited from: GeneratingUnit
variableCost	Money	inherited from: GeneratingUnit
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 429 shows all association ends of WindGeneratingUnit with other classes.

Table 429 – Association ends of Production::WindGeneratingUnit with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..*] SynchronousMachines	SynchronousMachine	inherited from: GeneratingUnit
[1..1]	[0..1] GenUnitOpSchedule	GenUnitOpSchedule	inherited from: GeneratingUnit
[1..1]	[0..*] GrossToNetActivePowerCurves	GrossToNetActivePowerCurve	inherited from: GeneratingUnit
[1..1]	[0..*] GenUnitOpCostCurves	GenUnitOpCostCurve	inherited from: GeneratingUnit
[1..1]	[0..*] ControlAreaGeneratingUnit	ControlAreaGeneratingUnit	inherited from: GeneratingUnit
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject

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

6.10 Package LoadModel

6.10.1 General

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

This information is used by Load Forecasting and Load Management.

Figure 68 shows class diagram Main.

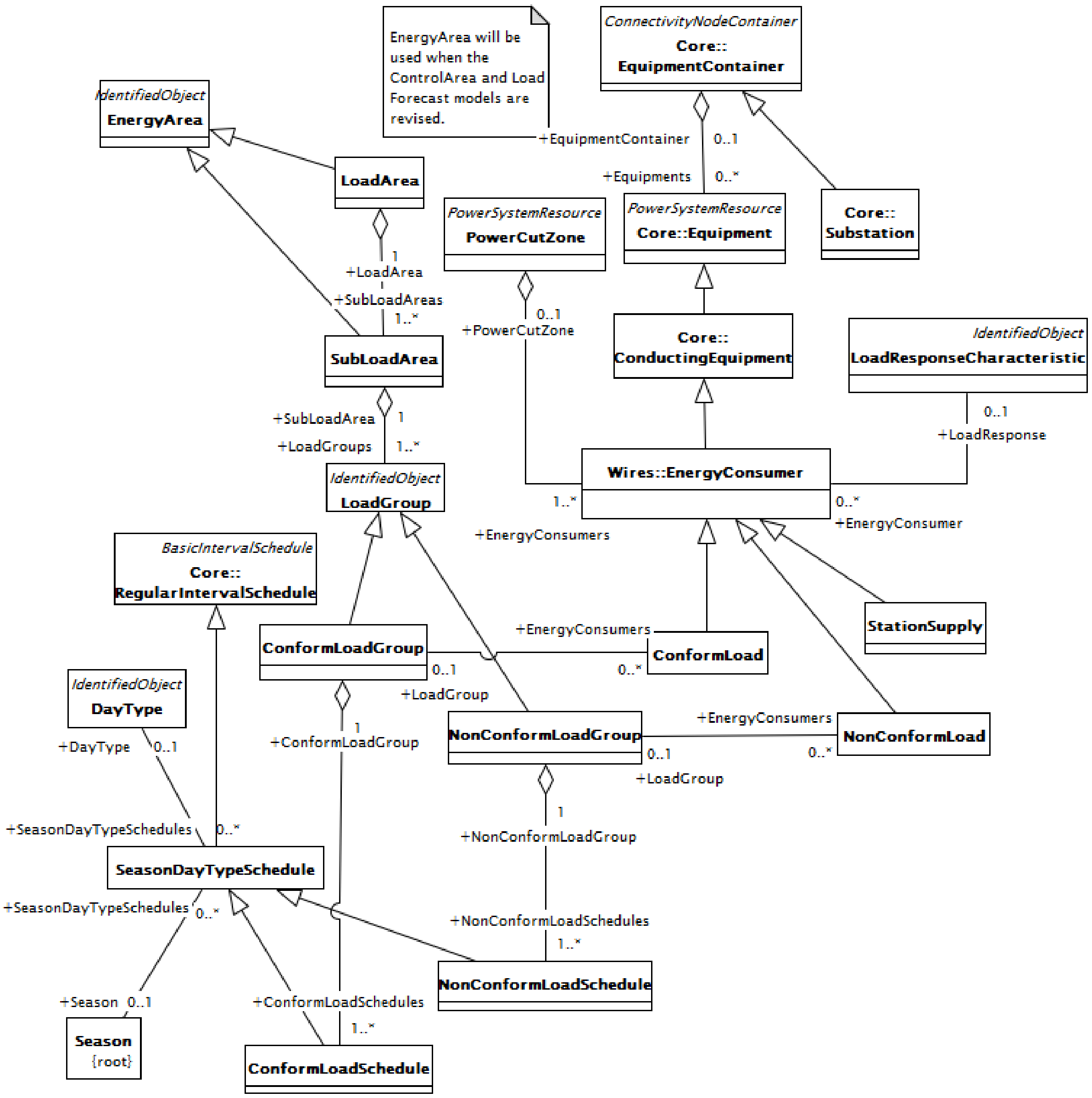

Figure 68 – Class diagram LoadModel::Main

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

Figure 69 shows class diagram Datatypes.

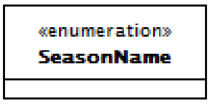

Figure 69 – Class diagram LoadModel::Datatypes

Shows datatypes for load model.

6.10.2 ConformLoad

ConformLoad represent loads that follow a daily load change pattern where the pattern can be used to scale the load with a system load.

Table 430 shows all attributes of ConformLoad.

Table 430 – Attributes of LoadModel::ConformLoad

name	type	description
customerCount	Integer	inherited from: EnergyConsumer
grounded	WindingConnection	inherited from: EnergyConsumer
pfixed	ActivePower	inherited from: EnergyConsumer
pfixedPct	PerCent	inherited from: EnergyConsumer
phaseConnection	PhaseShuntConnectionKind	inherited from: EnergyConsumer
qfixed	ReactivePower	inherited from: EnergyConsumer
qfixedPct	PerCent	inherited from: EnergyConsumer
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 431 shows all association ends of ConformLoad with other classes.

Table 431 – Association ends of LoadModel::ConformLoad with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] LoadGroup	ConformLoadGroup	Group of this ConformLoad.
[0..*]	[0..1] LoadResponse	LoadResponseCharacteristic	inherited from: EnergyConsumer
[1..1]	[0..*] EnergyConsumerPhase	EnergyConsumerPhase	inherited from: EnergyConsumer
[1..*]	[0..1] PowerCutZone	PowerCutZone	inherited from: EnergyConsumer
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.10.3 ConformLoadGroup

A group of loads conforming to an allocation pattern.

Table 432 shows all attributes of ConformLoadGroup.

Table 432 – Attributes of LoadModel::ConformLoadGroup

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

Table 433 shows all association ends of ConformLoadGroup with other classes.

Table 433 – Association ends of LoadModel::ConformLoadGroup with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] EnergyConsumers	ConformLoad	Conform loads assigned to this ConformLoadGroup.
[1..1]	[1..*] ConformLoadSchedules	ConformLoadSchedule	The ConformLoadSchedules in the ConformLoadGroup.
[1..*]	[1..1] SubLoadArea	SubLoadArea	inherited from: LoadGroup
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.10.4 ConformLoadSchedule

A curve of load versus time (X-axis) showing the active power values (Y1-axis) and reactive power (Y2-axis) for each unit of the period covered. This curve represents a typical pattern of load over the time period for a given day type and season.

Table 434 shows all attributes of ConformLoadSchedule.

Table 434 – Attributes of LoadModel::ConformLoadSchedule

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

Table 435 shows all association ends of ConformLoadSchedule with other classes.

Table 435 – Association ends of LoadModel::ConformLoadSchedule with other classes

[mult from]	[mult to] name	type	description
[1..*]	[1..1] ConformLoadGroup	ConformLoadGroup	The ConformLoadGroup where the ConformLoadSchedule belongs.
[0..*]	[0..1] DayType	DayType	inherited from: SeasonDayTypeSchedule
[0..*]	[0..1] Season	Season	inherited from: SeasonDayTypeSchedule
[1..1]	[1..*] TimePoints	RegularTimePoint	inherited from: RegularIntervalSchedule
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.10.5 DayType

Group of similar days. For example it could be used to represent weekdays, weekend, or holidays.

Table 436 shows all attributes of DayType.

Table 436 – Attributes of LoadModel::DayType

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

Table 437 shows all association ends of DayType with other classes.

Table 437 – Association ends of LoadModel::DayType with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] SeasonDayTypeSchedules	SeasonDayTypeSchedule	Schedules that use this DayType.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.10.6 EnergyArea

The class describes an area having energy production or consumption.

Table 438 shows all attributes of EnergyArea.

Table 438 – Attributes of LoadModel::EnergyArea

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

Table 439 shows all association ends of EnergyArea with other classes.

Table 439 – Association ends of LoadModel::EnergyArea with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..1] ControlArea	ControlArea	The control area specification that is used for the load forecast.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.10.7 LoadArea

The class is the root or first level in a hierarchical structure for grouping of loads for the purpose of load flow load scaling.

Table 440 shows all attributes of LoadArea.

Table 440 – Attributes of LoadModel::LoadArea

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

Table 441 shows all association ends of LoadArea with other classes.

Table 441 – Association ends of LoadModel::LoadArea with other classes

[mult from]	[mult to] name	type	description
[1..1]	[1..*] SubLoadAreas	SubLoadArea	The SubLoadAreas in the LoadArea.
[0..1]	[0..1] ControlArea	ControlArea	inherited from: EnergyArea
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.10.8 LoadGroup

The class is the third level in a hierarchical structure for grouping of loads for the purpose of load flow load scaling.

Table 442 shows all attributes of LoadGroup.

Table 442 – Attributes of LoadModel::LoadGroup

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

Table 443 shows all association ends of LoadGroup with other classes.

Table 443 – Association ends of LoadModel::LoadGroup with other classes

[mult from]	[mult to] name	type	description
[1..*]	[1..1] SubLoadArea	SubLoadArea	The SubLoadArea where the Loadgroup belongs.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.10.9 LoadResponseCharacteristic

Models the characteristic response of the load demand due to to changes in system conditions such as voltage and frequency. This is not related to demand response.

If LoadResponseCharacteristic.exponentModel is True, the voltage exponents are specified and used as to calculate:

Active power component = Pnominal × (Voltage/cim:BaseVoltage.nominalVoltage) ** cim:LoadResponseCharacteristic.pVoltageExponent

Reactive power component = Qnominal × (Voltage/cim:BaseVoltage.nominalVoltage)** cim:LoadResponseCharacteristic.qVoltageExponent

Where × means "multiply" and ** is "raised to power of".

Table 444 shows all attributes of LoadResponseCharacteristic.

Table 444 – Attributes of LoadModel::LoadResponseCharacteristic

name	type	description
exponentModel	Boolean	Indicates the exponential voltage dependency model (pVoltateExponent and qVoltageExponent) is to be used. If false, the coeficient model (consisting of pConstantImpedance, pConstantCurrent, pConstantPower, qConstantImpedance, qConstantCurrent, and qConstantPower) is to be used.
pConstantCurrent	Float	Portion of active power load modeled as constant current. Used only if the useExponentModel is false. This value is noralized against the sum of pZ, pI, and pP.
pConstantImpedance	Float	Portion of active power load modeled as constant impedance. Used only if the useExponentModel is false. This value is noralized against the sum of pZ, pI, and pP.
pConstantPower	Float	Portion of active power load modeled as constant power. Used only if the useExponentModel is false. This value is noralized against the sum of pZ, pI, and pP.
pFrequencyExponent	Float	Exponent of per unit frequency effecting active power.
pVoltageExponent	Float	Exponent of per unit voltage effecting real power. This model used only when "useExponentModel" is true.
qConstantCurrent	Float	Portion of reactive power load modeled as constant current. Used only if the useExponentModel is false. This value is noralized against the sum of qZ, qI, and qP.
qConstantImpedance	Float	Portion of reactive power load modeled as constant impedance. Used only if the useExponentModel is false. This value is noralized against the sum of qZ, qI, and qP.
qConstantPower	Float	Portion of reactive power load modeled as constant power. Used only if the useExponentModel is false. This value is noralized against the sum of qZ, qI, and qP.
qFrequencyExponent	Float	Exponent of per unit frequency effecting reactive power.
qVoltageExponent	Float	Exponent of per unit voltage effecting reactive power. This model used only when "useExponentModel" is true.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 445 shows all association ends of LoadResponseCharacteristic with other classes.

Table 445 – Association ends of LoadModel::LoadResponseCharacteristic with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] EnergyConsumer	EnergyConsumer	The set of loads that have the response characteristics.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.10.10 NonConformLoad

NonConformLoad represent loads that do not follow a daily load change pattern and changes are not correlated with the daily load change pattern.

Table 446 shows all attributes of NonConformLoad.

Table 446 – Attributes of LoadModel::NonConformLoad

name	type	description
customerCount	Integer	inherited from: EnergyConsumer
grounded	WindingConnection	inherited from: EnergyConsumer
pfixed	ActivePower	inherited from: EnergyConsumer
pfixedPct	PerCent	inherited from: EnergyConsumer
phaseConnection	PhaseShuntConnectionKind	inherited from: EnergyConsumer
qfixed	ReactivePower	inherited from: EnergyConsumer
qfixedPct	PerCent	inherited from: EnergyConsumer
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 447 shows all association ends of NonConformLoad with other classes.

Table 447 – Association ends of LoadModel::NonConformLoad with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] LoadGroup	NonConformLoadGroup	Group of this ConformLoad.
[0..*]	[0..1] LoadResponse	LoadResponseCharacteristic	inherited from: EnergyConsumer
[1..1]	[0..*] EnergyConsumerPhase	EnergyConsumerPhase	inherited from: EnergyConsumer
[1..*]	[0..1] PowerCutZone	PowerCutZone	inherited from: EnergyConsumer
[0..*]	[0..1] BaseVoltage	BaseVoltage	inherited from: ConductingEquipment
[1..1]	[0..*] Terminals	Terminal	inherited from: ConductingEquipment
[0..*]	[0..*] ProtectionEquipments	ProtectionEquipment	inherited from: ConductingEquipment
[1..1]	[0..*] ClearanceTags	ClearanceTag	inherited from: ConductingEquipment
[1..1]	[0..1] SvStatus	SvStatus	inherited from: ConductingEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.10.11 NonConformLoadGroup

Loads that do not follow a daily and seasonal load variation pattern.

Table 448 shows all attributes of NonConformLoadGroup.

Table 448 – Attributes of LoadModel::NonConformLoadGroup

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

Table 449 shows all association ends of NonConformLoadGroup with other classes.

Table 449 – Association ends of LoadModel::NonConformLoadGroup with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] EnergyConsumers	NonConformLoad	Conform loads assigned to this ConformLoadGroup.
[1..1]	[1..*] NonConformLoadSchedules	NonConformLoadSchedule	The NonConformLoadSchedules in the NonConformLoadGroup.
[1..*]	[1..1] SubLoadArea	SubLoadArea	inherited from: LoadGroup
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.10.12 NonConformLoadSchedule

An active power (Y1-axis) and reactive power (Y2-axis) schedule (curves) versus time (X-axis) for non-conforming loads, e.g., large industrial load or power station service (where modeled).

Table 450 shows all attributes of NonConformLoadSchedule.

Table 450 – Attributes of LoadModel::NonConformLoadSchedule

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

Table 451 shows all association ends of NonConformLoadSchedule with other classes.

Table 451 – Association ends of LoadModel::NonConformLoadSchedule with other classes

[mult from]	[mult to] name	type	description
[1..*]	[1..1] NonConformLoadGroup	NonConformLoadGroup	The NonConformLoadGroup where the NonConformLoadSchedule belongs.
[0..*]	[0..1] DayType	DayType	inherited from: SeasonDayTypeSchedule
[0..*]	[0..1] Season	Season	inherited from: SeasonDayTypeSchedule
[1..1]	[1..*] TimePoints	RegularTimePoint	inherited from: RegularIntervalSchedule
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.10.13 PowerCutZone

An area or zone of the power system which is used for load shedding purposes.

Table 452 shows all attributes of PowerCutZone.

Table 452 – Attributes of LoadModel::PowerCutZone

name	type	description
cutLevel1	PerCent	First level (amount) of load to cut as a percentage of total zone load.
cutLevel2	PerCent	Second level (amount) of load to cut as a percentage of total zone load.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 453 shows all association ends of PowerCutZone with other classes.

Table 453 – Association ends of LoadModel::PowerCutZone with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..*] EnergyConsumers	EnergyConsumer	Energy consumer is assigned to the power cut zone.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.10.14 Season root class

A specified time period of the year.

Table 454 shows all attributes of Season.

Table 454 – Attributes of LoadModel::Season

name	type	description
endDate	DateTime	Date season ends.
name	SeasonName	Name of the season.
startDate	DateTime	Date season starts.

Table 455 shows all association ends of Season with other classes.

Table 455 – Association ends of LoadModel::Season with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] SeasonDayTypeSchedules	SeasonDayTypeSchedule	Schedules that use this Season.

6.10.15 SeasonDayTypeSchedule

A time schedule covering a 24 hour period, with curve data for a specific type of season and day.

Table 456 shows all attributes of SeasonDayTypeSchedule.

Table 456 – Attributes of LoadModel::SeasonDayTypeSchedule

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

Table 457 shows all association ends of SeasonDayTypeSchedule with other classes.

Table 457 – Association ends of LoadModel::SeasonDayTypeSchedule with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] DayType	DayType	DayType for the Schedule.
[0..*]	[0..1] Season	Season	Season for the Schedule.
[1..1]	[1..*] TimePoints	RegularTimePoint	inherited from: RegularIntervalSchedule
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.10.16 SeasonName enumeration

Name of season.

Table 458 shows all literals of SeasonName.

Table 458 – Literals of LoadModel::SeasonName

literal	description
winter	Winter.
spring	Spring.
summer	Summer.
fall	Fall.

6.10.17 StationSupply

Station supply with load derived from the station output.

Table 459 shows all attributes of StationSupply.

Table 459 – Attributes of LoadModel::StationSupply

name	type	description
customerCount	Integer	inherited from: EnergyConsumer
grounded	WindingConnection	inherited from: EnergyConsumer
pfixed	ActivePower	inherited from: EnergyConsumer
pfixedPct	PerCent	inherited from: EnergyConsumer
phaseConnection	PhaseShuntConnectionKind	inherited from: EnergyConsumer
qfixed	ReactivePower	inherited from: EnergyConsumer
qfixedPct	PerCent	inherited from: EnergyConsumer
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 460 shows all association ends of StationSupply with other classes.

Table 460 – Association ends of LoadModel::StationSupply with other classes

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

6.10.18 SubLoadArea

The class is the second level in a hierarchical structure for grouping of loads for the purpose of load flow load scaling.

Table 461 shows all attributes of SubLoadArea.

Table 461 – Attributes of LoadModel::SubLoadArea

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

Table 462 shows all association ends of SubLoadArea with other classes.

Table 462 – Association ends of LoadModel::SubLoadArea with other classes

[mult from]	[mult to] name	type	description
[1..1]	[1..*] LoadGroups	LoadGroup	The Loadgroups in the SubLoadArea.
[1..*]	[1..1] LoadArea	LoadArea	The LoadArea where the SubLoadArea belongs.
[0..1]	[0..1] ControlArea	ControlArea	inherited from: EnergyArea
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.11 Package Outage

6.11.1 General

An extension to the Core and Wires packages that models information on the current and planned network configuration. These entities are optional within typical network applications.

Figure 70 shows class diagram Datatypes.

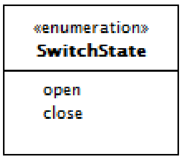

Figure 70 – Class diagram Outage::Datatypes

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

Figure 71 shows class diagram Main.

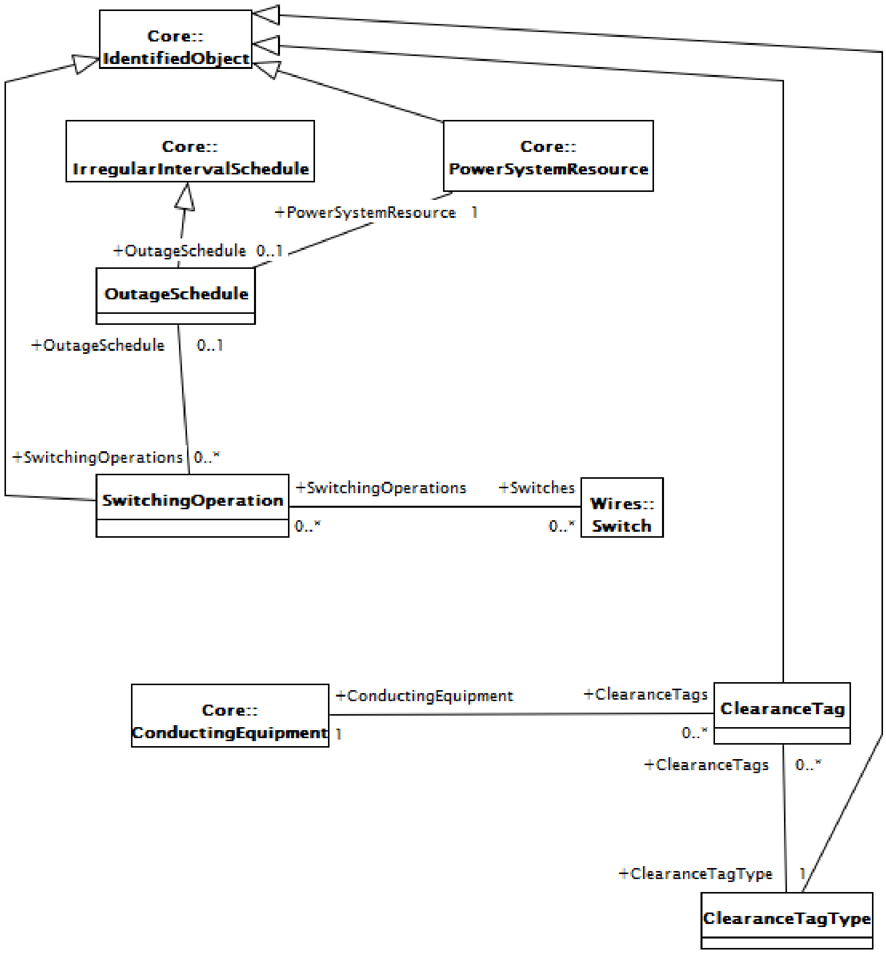

Figure 71 – Class diagram Outage::Main

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

6.11.2 ClearanceTag

A clearance tag that is used to authorize and schedule work on conducting equipment in the field. Tagged equipment is not available for commercial service.

Table 463 shows all attributes of ClearanceTag.

Table 463 – Attributes of Outage::ClearanceTag

name	type	description
authorityName	String	The name of the person who is authorized to issue the tag.
deenergizeReqFlag	Boolean	Set true if equipment must be deenergized.
groundReqFlag	Boolean	Set true if equipment must be grounded.
phaseCheckReqFlag	Boolean	Set true if equipment phasing must be checked.
tagIssueTime	DateTime	The time at which the clearance tag was issued.
workDescription	String	Description of the work to be performed.
workEndTime	DateTime	The time at which the clearance tag is scheduled to be removed.
workStartTime	DateTime	The time at which the clearance tag is scheduled to be set.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 464 shows all association ends of ClearanceTag with other classes.

Table 464 – Association ends of Outage::ClearanceTag with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] ConductingEquipment	ConductingEquipment	Conducting equipment of this clearance tags for authorized field work.
[0..*]	[1..1] ClearanceTagType	ClearanceTagType	The type of tag, depending on the purpose of the work to be performed and/or the type of supervisory control allowed.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.11.3 ClearanceTagType

Type of ClearanceTag. Could indicate the type of work to be performed and/or the type of supervisory control.

Table 465 shows all attributes of ClearanceTagType.

Table 465 – Attributes of Outage::ClearanceTagType

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

Table 466 shows all association ends of ClearanceTagType with other classes.

Table 466 – Association ends of Outage::ClearanceTagType with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] ClearanceTags	ClearanceTag	The ClearanceTags currently being defined for this type.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.11.4 OutageSchedule

The period of time that a piece of equipment is out of service, for example, for maintenance or testing; including the equipment's active power rating while under maintenance. The X-axis represents absolute time and the Y-axis represents the equipment's available rating while out of service.

Table 467 shows all attributes of OutageSchedule.

Table 467 – Attributes of Outage::OutageSchedule

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

Table 468 shows all association ends of OutageSchedule with other classes.

Table 468 – Association ends of Outage::OutageSchedule with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] PowerSystemResource	PowerSystemResource	A power system resource may have an outage schedule.
[0..1]	[0..*] SwitchingOperations	SwitchingOperation	An OutageSchedule may operate many switches.
[1..1]	[1..*] TimePoints	IrregularTimePoint	inherited from: IrregularIntervalSchedule
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.11.5 SwitchingOperation

A SwitchingOperation is used to define individual switch operations for an OutageSchedule. This OutageSchedule may be associated with another item of Substation such as a Transformer, Line, or Generator; or with the Switch itself as a PowerSystemResource. A Switch may be referenced by many OutageSchedules.

Table 469 shows all attributes of SwitchingOperation.

Table 469 – Attributes of Outage::SwitchingOperation

name	type	description
newState	SwitchState	The switch position that shall result from this SwitchingOperation.
operationTime	DateTime	Time of operation in same units as OutageSchedule.xAxixUnits.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 470 shows all association ends of SwitchingOperation with other classes.

Table 470 – Association ends of Outage::SwitchingOperation with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] Switches	Switch	A switch may be operated by many schedules.
[0..*]	[0..1] OutageSchedule	OutageSchedule	An OutageSchedule may operate many switches.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.11.6 SwitchState enumeration

Possible states for a switch.

Table 471 shows all literals of SwitchState.

Table 471 – Literals of Outage::SwitchState

literal	description
open	Switch is open.
close	Switch is closed.

6.12 Package AuxiliaryEquipment

6.12.1 General

Contains equipment which is not normal conducting equipment such as sensors, fault locators, and surge protectors. These devices do not define power carrying topological connections as conducting equipment, but are associated to terminals of other conducting equipment.

Figure 72 shows class diagram AuxiliaryEquipment.

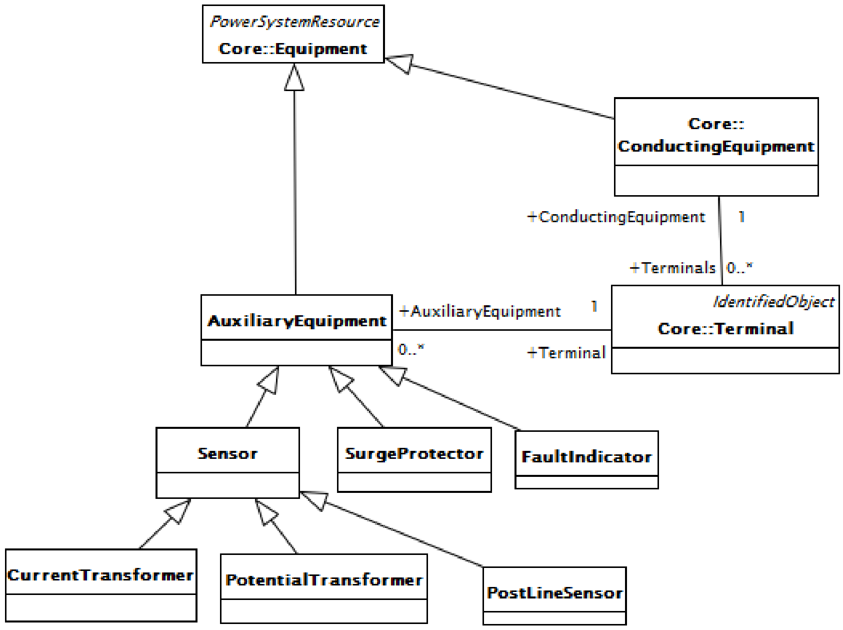

Figure 72 – Class diagram AuxiliaryEquipment::AuxiliaryEquipment

Inheritance and associations for auxiliary equipment.

6.12.2 AuxiliaryEquipment

AuxiliaryEquipment describe equipment that is not performing any primary functions but support for the equipment performing the primary function.

AuxiliaryEquipment is attached to primary eqipment via an association with Terminal.

Table 472 shows all attributes of AuxiliaryEquipment.

Table 472 – Attributes of AuxiliaryEquipment::AuxiliaryEquipment

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

Table 473 shows all association ends of AuxiliaryEquipment with other classes.

Table 473 – Association ends of AuxiliaryEquipment::AuxiliaryEquipment with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Terminal	Terminal	The Terminal at the equipment where the AuxiliaryEquipment is attached.
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.12.3 CurrentTransformer

Instrument transformer used to measure electrical qualities of the circuit that is being protected and/or monitored. Typically used as current transducer for the purpose of metering or protection. A typical secondary current rating would be 5 A.

Table 474 shows all attributes of CurrentTransformer.

Table 474 – Attributes of AuxiliaryEquipment::CurrentTransformer

name	type	description
accuracyClass	String	CT accuracy classification.
accuracyLimit	PerCent	Percent of rated current for which the CT remains accurate within specified limits.
coreCount	Integer	Number of cores.
ctClass	String	CT classification; i.e. class 10P.
maxRatio	Float	For multi-ratio CT's, the maximum permissable ratio attainable.
usage	String	Intended usage of the CT; i.e. metering, protection.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 475 shows all association ends of CurrentTransformer with other classes.

Table 475 – Association ends of AuxiliaryEquipment::CurrentTransformer with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Terminal	Terminal	inherited from: AuxiliaryEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.12.4 FaultIndicator

A FaultIndicator is typically only an indicator (which may or may not be remotely monitored), and not a piece of equipment that actually initiates a protection event. It is used for FLISR (Fault Location, Isolation and Restoration) purposes, assisting with the dispatch of crews to "most likely" part of the network (i.e. assists with determining circuit section where the fault most likely happened).

Table 476 shows all attributes of FaultIndicator.

Table 476 – Attributes of AuxiliaryEquipment::FaultIndicator

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

Table 477 shows all association ends of FaultIndicator with other classes.

Table 477 – Association ends of AuxiliaryEquipment::FaultIndicator with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Terminal	Terminal	inherited from: AuxiliaryEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.12.5 PostLineSensor

A sensor used mainly in overhead distribution networks as the source of both current and voltage measurements.

Table 478 shows all attributes of PostLineSensor.

Table 478 – Attributes of AuxiliaryEquipment::PostLineSensor

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

Table 479 shows all association ends of PostLineSensor with other classes.

Table 479 – Association ends of AuxiliaryEquipment::PostLineSensor with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Terminal	Terminal	inherited from: AuxiliaryEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.12.6 PotentialTransformer

Instrument transformer (also known as Voltage Transformer) used to measure electrical qualities of the circuit that is being protected and/or monitored. Typically used as voltage transducer for the purpose of metering, protection, or sometimes auxiliary substation supply. A typical secondary voltage rating would be 120 V.

Table 480 shows all attributes of PotentialTransformer.

Table 480 – Attributes of AuxiliaryEquipment::PotentialTransformer

name	type	description
accuracyClass	String	PT accuracy classification.
nominalRatio	Float	Nominal ratio between the primary and secondary voltage.
ptClass	String	Potential transformer (PT) classification.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 481 shows all association ends of PotentialTransformer with other classes.

Table 481 – Association ends of AuxiliaryEquipment::PotentialTransformer with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Terminal	Terminal	inherited from: AuxiliaryEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.12.7 Sensor

This class describe devices that transform a measured quantity into signals that can be presented at displays, used in control or be recorded.

Table 482 shows all attributes of Sensor.

Table 482 – Attributes of AuxiliaryEquipment::Sensor

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

Table 483 shows all association ends of Sensor with other classes.

Table 483 – Association ends of AuxiliaryEquipment::Sensor with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Terminal	Terminal	inherited from: AuxiliaryEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.12.8 SurgeProtector

Shunt device, installed on the network, usually in the proximity of electrical equipment in order to protect the said equipment against transient voltage spikes caused by lightning or switching activity.

Table 484 shows all attributes of SurgeProtector.

Table 484 – Attributes of AuxiliaryEquipment::SurgeProtector

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

Table 485 shows all association ends of SurgeProtector with other classes.

Table 485 – Association ends of AuxiliaryEquipment::SurgeProtector with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Terminal	Terminal	inherited from: AuxiliaryEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.13 Package Protection

6.13.1 General

An extension to the Core and Wires packages that models information for protection equipment such as relays. These entities are used within training simulators and distribution network fault location applications.

Figure 73 shows class diagram Main.

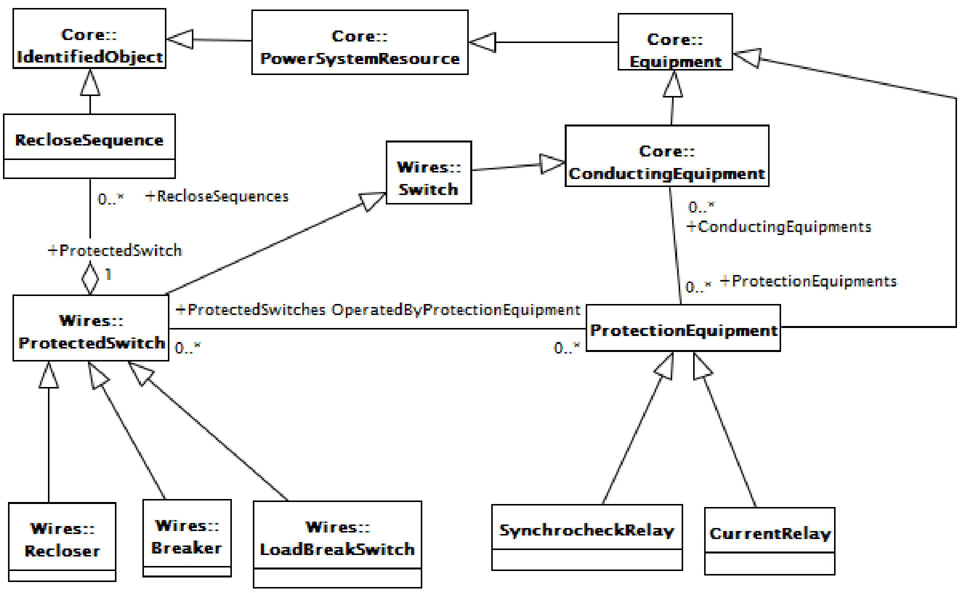

Figure 73 – Class diagram Protection::Main

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

6.13.2 CurrentRelay

A device that checks current flow values in any direction or designated direction.

Table 486 shows all attributes of CurrentRelay.

Table 486 – Attributes of Protection::CurrentRelay

name	type	description
currentLimit1	CurrentFlow	Current limit number 1 for inverse time pickup.
currentLimit2	CurrentFlow	Current limit number 2 for inverse time pickup.
currentLimit3	CurrentFlow	Current limit number 3 for inverse time pickup.
inverseTimeFlag	Boolean	Set true if the current relay has inverse time characteristic.
timeDelay1	Seconds	Inverse time delay number 1 for current limit number 1.
timeDelay2	Seconds	Inverse time delay number 2 for current limit number 2.
timeDelay3	Seconds	Inverse time delay number 3 for current limit number 3.
relayDelayTime	Seconds	inherited from: ProtectionEquipment
highLimit	Float	inherited from: ProtectionEquipment
lowLimit	Float	inherited from: ProtectionEquipment
powerDirectionFlag	Boolean	inherited from: ProtectionEquipment
unitSymbol	UnitSymbol	inherited from: ProtectionEquipment
unitMultiplier	UnitMultiplier	inherited from: ProtectionEquipment
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 487 shows all association ends of CurrentRelay with other classes.

Table 487 – Association ends of Protection::CurrentRelay with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] ConductingEquipments	ConductingEquipment	inherited from: ProtectionEquipment
[0..*]	[0..*] ProtectedSwitches	ProtectedSwitch	inherited from: ProtectionEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.13.3 ProtectionEquipment

An electrical device designed to respond to input conditions in a prescribed manner and after specified conditions are met to cause contact operation or similar abrupt change in associated electric control circuits, or simply to display the detected condition. Protection equipment are associated with conducting equipment and usually operate circuit breakers.

Table 488 shows all attributes of ProtectionEquipment.

Table 488 – Attributes of Protection::ProtectionEquipment

name	type	description
relayDelayTime	Seconds	The time delay from detection of abnormal conditions to relay operation.
highLimit	Float	The maximum allowable value.
lowLimit	Float	The minimum allowable value.
powerDirectionFlag	Boolean	Direction same as positive active power flow value.
unitSymbol	UnitSymbol	The unit of measure of the value.
unitMultiplier	UnitMultiplier	The unit multiplier of the value.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 489 shows all association ends of ProtectionEquipment with other classes.

Table 489 – Association ends of Protection::ProtectionEquipment with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] ConductingEquipments	ConductingEquipment	Protection equipment may be used to protect specific conducting equipment.
[0..*]	[0..*] ProtectedSwitches	ProtectedSwitch	Protected switches operated by this ProtectionEquipment.
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.13.4 RecloseSequence

A reclose sequence (open and close) is defined for each possible reclosure of a breaker.

Table 490 shows all attributes of RecloseSequence.

Table 490 – Attributes of Protection::RecloseSequence

name	type	description
recloseDelay	Seconds	Indicates the time lapse before the reclose step will execute a reclose.
recloseStep	Integer	Indicates the ordinal position of the reclose step relative to other steps in the sequence.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 491 shows all association ends of RecloseSequence with other classes.

Table 491 – Association ends of Protection::RecloseSequence with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] ProtectedSwitch	ProtectedSwitch	A breaker may have zero or more automatic reclosures after a trip occurs.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.13.5 SynchrocheckRelay

A device that operates when two AC circuits are within the desired limits of frequency, phase angle, and voltage, to permit or to cause the paralleling of these two circuits. Used to prevent the paralleling of non-synchronous topological islands.

Table 492 shows all attributes of SynchrocheckRelay.

Table 492 – Attributes of Protection::SynchrocheckRelay

name	type	description
maxAngleDiff	AngleRadians	The maximum allowable voltage vector phase angle difference across the open device.
maxFreqDiff	Frequency	The maximum allowable frequency difference across the open device.
maxVoltDiff	Voltage	The maximum allowable difference voltage across the open device.
relayDelayTime	Seconds	inherited from: ProtectionEquipment
highLimit	Float	inherited from: ProtectionEquipment
lowLimit	Float	inherited from: ProtectionEquipment
powerDirectionFlag	Boolean	inherited from: ProtectionEquipment
unitSymbol	UnitSymbol	inherited from: ProtectionEquipment
unitMultiplier	UnitMultiplier	inherited from: ProtectionEquipment
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 493 shows all association ends of SynchrocheckRelay with other classes.

Table 493 – Association ends of Protection::SynchrocheckRelay with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..*] ConductingEquipments	ConductingEquipment	inherited from: ProtectionEquipment
[0..*]	[0..*] ProtectedSwitches	ProtectedSwitch	inherited from: ProtectionEquipment
[0..*]	[0..1] EquipmentContainer	EquipmentContainer	inherited from: Equipment
[1..1]	[0..*] ContingencyEquipment	ContingencyEquipment	inherited from: Equipment
[0..1]	[0..*] OperationalLimitSet	OperationalLimitSet	inherited from: Equipment
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.14 Package Equivalents

6.14.1 General

The equivalents package models equivalent networks.

Figure 74 shows class diagram Main.

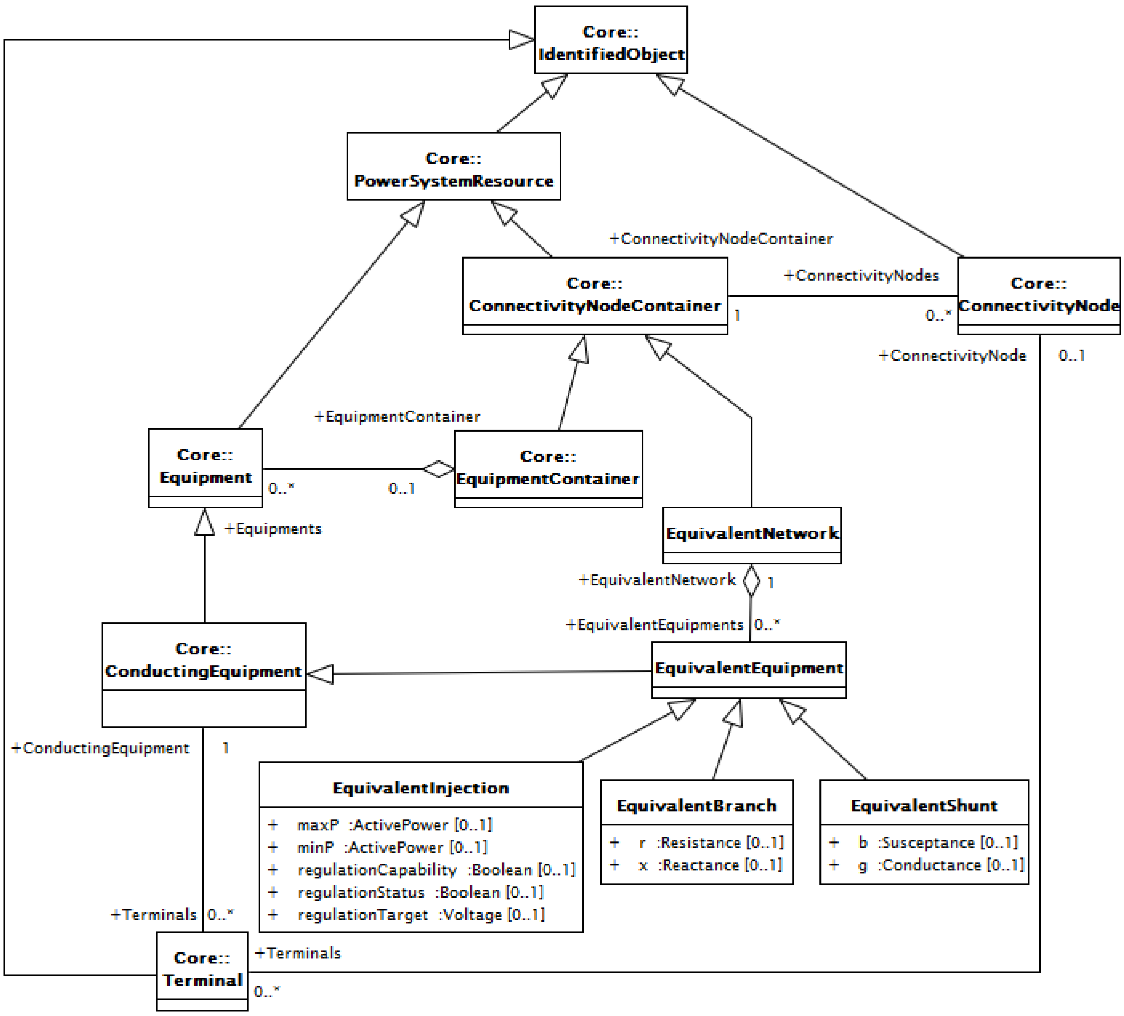

Figure 74 – Class diagram Equivalents::Main

Shows model for equivalent networks.

6.14.2 EquivalentBranch

The class represents equivalent branches.

Table 494 shows all attributes of EquivalentBranch.

Table 494 – Attributes of Equivalents::EquivalentBranch

name	type	description
r	Resistance	Positive sequence series resistance of the reduced branch.
x	Reactance	Positive sequence series reactance of the reduced branch.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 495 shows all association ends of EquivalentBranch with other classes.

Table 495 – Association ends of Equivalents::EquivalentBranch with other classes

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

6.14.3 EquivalentEquipment

The class represents equivalent objects that are the result of a network reduction. The class is the base for equivalent objects of different types.

Table 496 shows all attributes of EquivalentEquipment.

Table 496 – Attributes of Equivalents::EquivalentEquipment

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

Table 497 shows all association ends of EquivalentEquipment with other classes.

Table 497 – Association ends of Equivalents::EquivalentEquipment with other classes

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

6.14.4 EquivalentInjection

This class represents equivalent injections (generation or load). Voltage regulation is allowed only at the point of connection.

Table 498 shows all attributes of EquivalentInjection.

Table 498 – Attributes of Equivalents::EquivalentInjection

name	type	description
maxP	ActivePower	Minimum active power of the injection.
minP	ActivePower	Maximum active power of the injection.
regulationCapability	Boolean	Specifies whether or not the EquivalentInjection has the capability to regulate the local voltage.
regulationStatus	Boolean	Specifies the default regulation status of the EquivalentInjection. True is regulating. False is not regulating.
regulationTarget	Voltage	The target voltage for voltage regulation.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 499 shows all association ends of EquivalentInjection with other classes.

Table 499 – Association ends of Equivalents::EquivalentInjection with other classes

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

6.14.5 EquivalentNetwork

A class that represents an external meshed network that has been reduced to an electrically equivalent model. The ConnectivityNodes contained in the equivalent are intended to reflect internal nodes of the equivalent. The boundary Connectivity nodes where the equivalent connects outside itself are NOT contained by the equivalent.

Table 500 shows all attributes of EquivalentNetwork.

Table 500 – Attributes of Equivalents::EquivalentNetwork

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

Table 501 shows all association ends of EquivalentNetwork with other classes.

Table 501 – Association ends of Equivalents::EquivalentNetwork with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] EquivalentEquipments	EquivalentEquipment	The associated reduced equivalents.
[1..1]	[0..*] ConnectivityNodes	ConnectivityNode	inherited from: ConnectivityNodeContainer
[0..1]	[0..*] TopologicalNode	TopologicalNode	inherited from: ConnectivityNodeContainer
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.14.6 EquivalentShunt

The class represents equivalent shunts.

Table 502 shows all attributes of EquivalentShunt.

Table 502 – Attributes of Equivalents::EquivalentShunt

name	type	description
b	Susceptance	Positive sequence shunt susceptance.
g	Conductance	Positive sequence shunt conductance.
aggregate	Boolean	inherited from: Equipment
normallyInService	Boolean	inherited from: Equipment
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 503 shows all association ends of EquivalentShunt with other classes.

Table 503 – Association ends of Equivalents::EquivalentShunt with other classes

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

6.15 Package Meas

6.15.1 General

Contains entities that describe dynamic measurement data exchanged between applications.

Figure 75 shows class diagram Datatypes.

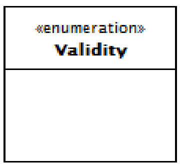

Figure 75 – Class diagram Meas::Datatypes

Shows datatypes for measurements.

Figure 76 shows class diagram Control.

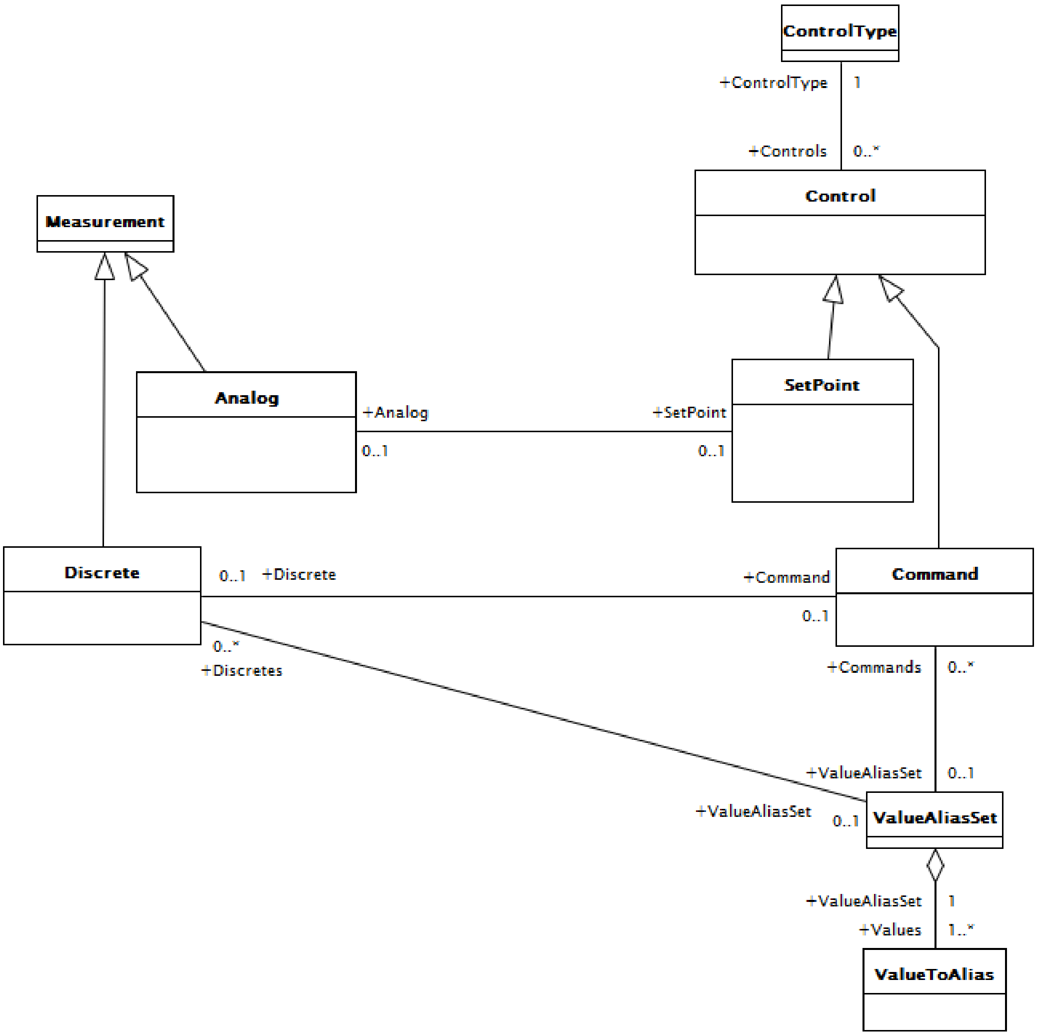

Figure 76 – Class diagram Meas::Control

This diagram shows all classes included in the Meas package.

Figure 77 shows class diagram Measurement.

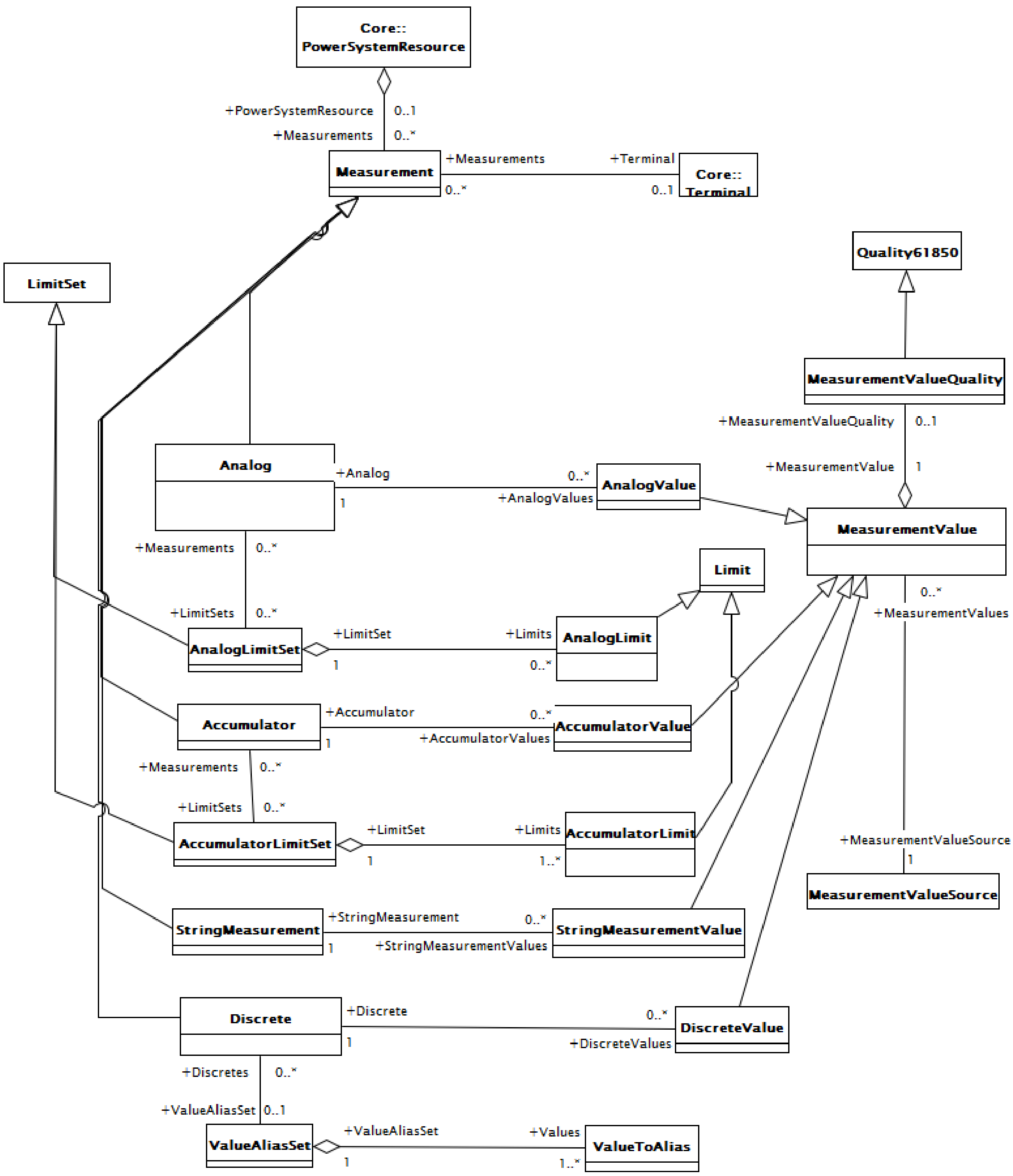

Figure 77 – Class diagram Meas::Measurement

This diagram shows classes central to the Measurement package and connections to some external classes.

Figure 78 shows class diagram MeasurementInheritance.

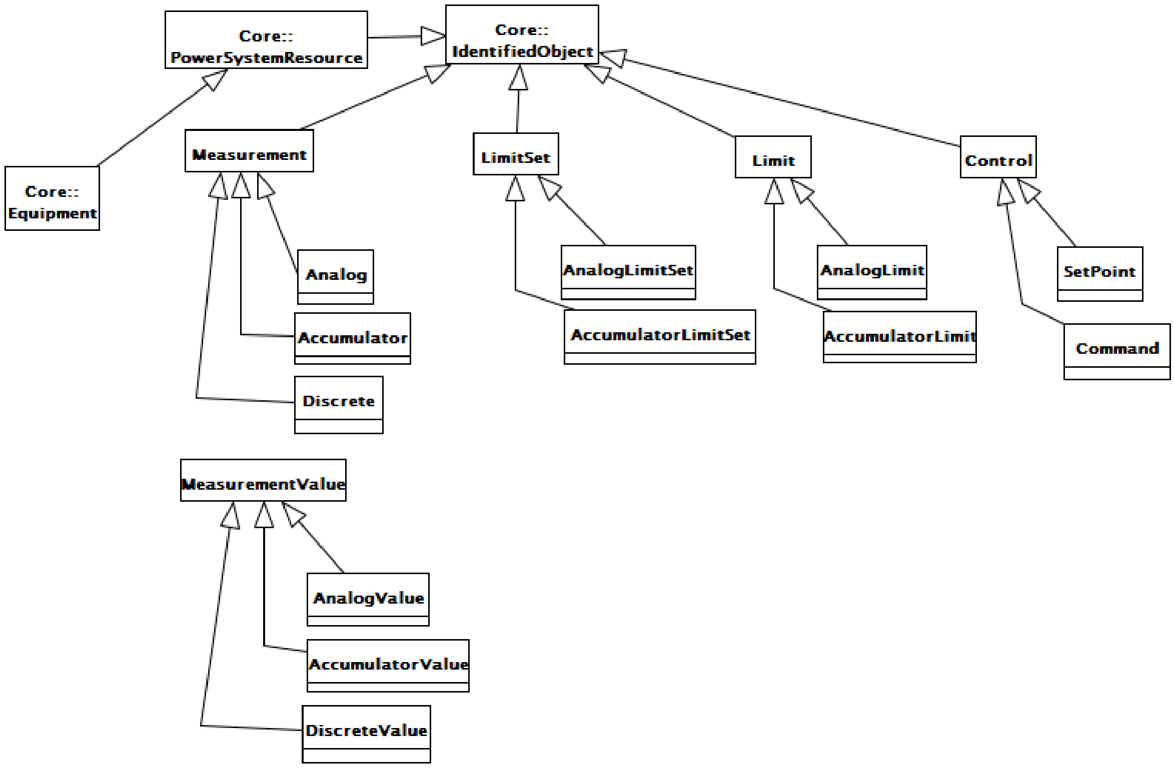

Figure 78 – Class diagram Meas::MeasurementInheritance

This diagram shows the measurement classes inheritance.

Figure 79 shows class diagram Quality.

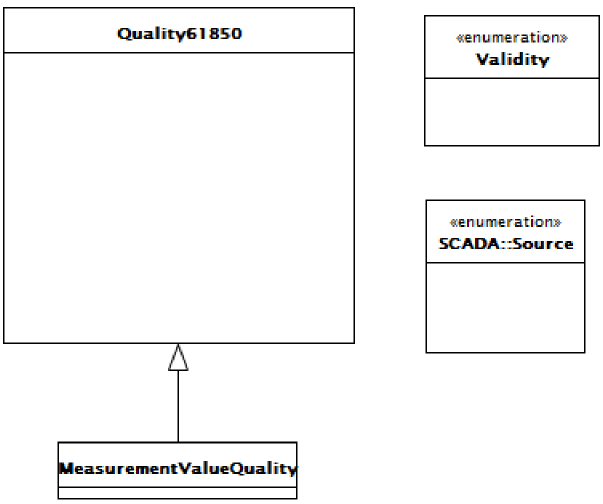

Figure 79 – Class diagram Meas::Quality

This diagram shows the details of the quality codes. The quality flags can be used also in other packages not only specific to MeasurementValues. A generalized Quality class is added and inherited into MeasurementValueQuality.

6.15.2 Accumulator

Accumulator represents a accumulated (counted) Measurement, e.g. an energy value.

Table 504 shows all attributes of Accumulator.

Table 504 – Attributes of Meas::Accumulator

name	type	description
maxValue	Integer	Normal value range maximum for any of the MeasurementValue.values. Used for scaling, e.g. in bar graphs or of telemetered raw values.
measurementType	String	inherited from: Measurement
phases	PhaseCode	inherited from: Measurement
unitMultiplier	UnitMultiplier	inherited from: Measurement
unitSymbol	UnitSymbol	inherited from: Measurement
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 505 shows all association ends of Accumulator with other classes.

Table 505 – Association ends of Meas::Accumulator with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] AccumulatorValues	AccumulatorValue	The values connected to this measurement.
[0..*]	[0..*] LimitSets	AccumulatorLimitSet	A measurement may have zero or more limit ranges defined for it.
[0..*]	[0..1] Terminal	Terminal	inherited from: Measurement
[0..*]	[0..1] PowerSystemResource	PowerSystemResource	inherited from: Measurement
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.3 AccumulatorLimit

Limit values for Accumulator measurements.

Table 506 shows all attributes of AccumulatorLimit.

Table 506 – Attributes of Meas::AccumulatorLimit

name	type	description
value	Integer	The value to supervise against. The value is positive.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 507 shows all association ends of AccumulatorLimit with other classes.

Table 507 – Association ends of Meas::AccumulatorLimit with other classes

[mult from]	[mult to] name	type	description
[1..*]	[1..1] LimitSet	AccumulatorLimitSet	The set of limits.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.4 AccumulatorLimitSet

An AccumulatorLimitSet specifies a set of Limits that are associated with an Accumulator measurement.

Table 508 shows all attributes of AccumulatorLimitSet.

Table 508 – Attributes of Meas::AccumulatorLimitSet

name	type	description
isPercentageLimits	Boolean	inherited from: LimitSet
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 509 shows all association ends of AccumulatorLimitSet with other classes.

Table 509 – Association ends of Meas::AccumulatorLimitSet with other classes

[mult from]	[mult to] name	type	description
[1..1]	[1..*] Limits	AccumulatorLimit	The limit values used for supervision of Measurements.
[0..*]	[0..*] Measurements	Accumulator	The Measurements using the LimitSet.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.5 AccumulatorValue

AccumulatorValue represents an accumulated (counted) MeasurementValue.

Table 510 shows all attributes of AccumulatorValue.

Table 510 – Attributes of Meas::AccumulatorValue

name	type	description
value	Integer	The value to supervise. The value is positive.
timeStamp	DateTime	inherited from: MeasurementValue
sensorAccuracy	PerCent	inherited from: MeasurementValue
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 511 shows all association ends of AccumulatorValue with other classes.

Table 511 – Association ends of Meas::AccumulatorValue with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Accumulator	Accumulator	Measurement to which this value is connected.
[1..1]	[0..1] RemoteSource	RemoteSource	inherited from: MeasurementValue
[1..1]	[0..1] MeasurementValueQuality	MeasurementValueQualit y	inherited from: MeasurementValue
[0..*]	[1..1] MeasurementValueSource	MeasurementValueSourc e	inherited from: MeasurementValue
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.6 Analog

Analog represents an analog Measurement.

Table 512 shows all attributes of Analog.

Table 512 – Attributes of Meas::Analog

name	type	description
maxValue	Float	Normal value range maximum for any of the MeasurementValue.values. Used for scaling, e.g. in bar graphs or of telemetered raw values.
minValue	Float	Normal value range minimum for any of the MeasurementValue.values. Used for scaling, e.g. in bar graphs or of telemetered raw values.
normalValue	Float	Normal measurement value, e.g., used for percentage calculations.
positiveFlowIn	Boolean	If true then this measurement is an active power, reactive power or current with the convention that a positive value measured at the Terminal means power is flowing into the related PowerSystemResource.
measurementType	String	inherited from: Measurement
phases	PhaseCode	inherited from: Measurement
unitMultiplier	UnitMultiplier	inherited from: Measurement
unitSymbol	UnitSymbol	inherited from: Measurement
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 513 shows all association ends of Analog with other classes.

Table 513 – Association ends of Meas::Analog with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] AnalogValues	AnalogValue	The values connected to this measurement.
[0..*]	[0..*] LimitSets	AnalogLimitSet	A measurement may have zero or more limit ranges defined for it.
[0..1]	[0..1] SetPoint	SetPoint	The Control variable associated with the Measurement.
[0..*]	[0..1] Terminal	Terminal	inherited from: Measurement
[0..*]	[0..1] PowerSystemResource	PowerSystemResource	inherited from: Measurement
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.7 AnalogLimit

Limit values for Analog measurements.

Table 514 shows all attributes of AnalogLimit.

Table 514 – Attributes of Meas::AnalogLimit

name	type	description
value	Float	The value to supervise against.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 515 shows all association ends of AnalogLimit with other classes.

Table 515 – Association ends of Meas::AnalogLimit with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] LimitSet	AnalogLimitSet	The set of limits.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.8 AnalogLimitSet

An AnalogLimitSet specifies a set of Limits that are associated with an Analog measurement.

Table 516 shows all attributes of AnalogLimitSet.

Table 516 – Attributes of Meas::AnalogLimitSet

name	type	description
isPercentageLimits	Boolean	inherited from: LimitSet
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 517 shows all association ends of AnalogLimitSet with other classes.

Table 517 – Association ends of Meas::AnalogLimitSet with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] Limits	AnalogLimit	The limit values used for supervision of Measurements.
[0..*]	[0..*] Measurements	Analog	The Measurements using the LimitSet.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.9 AnalogValue

AnalogValue represents an analog MeasurementValue.

Table 518 shows all attributes of AnalogValue.

Table 518 – Attributes of Meas::AnalogValue

name	type	description
value	Float	The value to supervise.
timeStamp	DateTime	inherited from: MeasurementValue
sensorAccuracy	PerCent	inherited from: MeasurementValue
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 519 shows all association ends of AnalogValue with other classes.

Table 519 – Association ends of Meas::AnalogValue with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Analog	Analog	Measurement to which this value is connected.
[1..1]	[0..*] AltGeneratingUnit	AltGeneratingUnitMeas	The alternate generating unit for which this measurement value applies.
[1..1]	[0..*] AltTieMeas	AltTieMeas	The usage of the measurement within the control area specification.
[1..1]	[0..1] RemoteSource	RemoteSource	inherited from: MeasurementValue
[1..1]	[0..1] MeasurementValueQuality	MeasurementValueQuality	inherited from: MeasurementValue
[0..*]	[1..1] MeasurementValueSource	MeasurementValueSource	inherited from: MeasurementValue
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.10 Command

A Command is a discrete control used for supervisory control.

Table 520 shows all attributes of Command.

Table 520 – Attributes of Meas::Command

name	type	description
normalValue	Integer	Normal value for Control.value e.g. used for percentage scaling.
value	Integer	The value representing the actuator output.
operationInProgress	Boolean	inherited from: Control
timeStamp	DateTime	inherited from: Control
unitMultiplier	UnitMultiplier	inherited from: Control
unitSymbol	UnitSymbol	inherited from: Control
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 521 shows all association ends of Command with other classes.

Table 521 – Association ends of Meas::Command with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] ValueAliasSet	ValueAliasSet	The Commands using the set for translation.
[0..1]	[0..1] Discrete	Discrete	The Measurement variable used for control.
[1..*]	[0..1] RegulatingCondEq	RegulatingCondEq	inherited from: Control
[1..1]	[0..1] RemoteControl	RemoteControl	inherited from: Control
[0..*]	[1..1] ControlType	ControlType	inherited from: Control
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.11 Control

Control is used for supervisory/device control. It represents control outputs that are used to change the state in a process, e.g. close or open breaker, a set point value or a raise lower command.

Table 522 shows all attributes of Control.

Table 522 – Attributes of Meas::Control

name	type	description
operationInProgress	Boolean	Indicates that a client is currently sending control commands that has not completed.
timeStamp	DateTime	The last time a control output was sent.
unitMultiplier	UnitMultiplier	The unit multiplier of the controlled quantity.
unitSymbol	UnitSymbol	The unit of measure of the controlled quantity.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 523 shows all association ends of Control with other classes.

Table 523 – Association ends of Meas::Control with other classes

[mult from]	[mult to] name	type	description
[1..*]	[0..1] RegulatingCondEq	RegulatingCondEq	Regulating device governed by this control output.
[1..1]	[0..1] RemoteControl	RemoteControl	The remote point controlling the physical actuator.
[0..*]	[1..1] ControlType	ControlType	The type of control.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.12 ControlType

Specifies the type of Control, e.g. BreakerOn/Off, GeneratorVoltageSetPoint, TieLineFlow, etc. The ControlType.name shall be unique among all specified types and describe the type. The ControlType.aliasName is meant to be used for localization.

Table 524 shows all attributes of ControlType.

Table 524 – Attributes of Meas::ControlType

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

Table 525 shows all association ends of ControlType with other classes.

Table 525 – Association ends of Meas::ControlType with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] Controls	Control	The controls having the control type.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.13 Discrete

Discrete represents a discrete Measurement, i.e. a Measurement reprsenting discrete values, e.g. a Breaker position.

Table 526 shows all attributes of Discrete.

Table 526 – Attributes of Meas::Discrete

name	type	description
maxValue	Integer	Normal value range maximum for any of the MeasurementValue.values. Used for scaling, e.g. in bar graphs or of telemetered raw values.
minValue	Integer	Normal value range minimum for any of the MeasurementValue.values. Used for scaling, e.g. in bar graphs or of telemetered raw values.
normalValue	Integer	Normal measurement value, e.g., used for percentage calculations.
measurementType	String	inherited from: Measurement
phases	PhaseCode	inherited from: Measurement
unitMultiplier	UnitMultiplier	inherited from: Measurement
unitSymbol	UnitSymbol	inherited from: Measurement
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 527 shows all association ends of Discrete with other classes.

Table 527 – Association ends of Meas::Discrete with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..1] Command	Command	The Control variable associated with the Measurement.
[1..1]	[0..*] DiscreteValues	DiscreteValue	The values connected to this measurement.
[0..*]	[0..1] ValueAliasSet	ValueAliasSet	The ValueAliasSet used for translation of a MeasurementValue.value to a name.
[0..*]	[0..1] Terminal	Terminal	inherited from: Measurement
[0..*]	[0..1] PowerSystemResource	PowerSystemResource	inherited from: Measurement
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.14 DiscreteValue

DiscreteValue represents a discrete MeasurementValue.

Table 528 shows all attributes of DiscreteValue.

Table 528 – Attributes of Meas::DiscreteValue

name	type	description
value	Integer	The value to supervise.
timeStamp	DateTime	inherited from: MeasurementValue
sensorAccuracy	PerCent	inherited from: MeasurementValue
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 529 shows all association ends of DiscreteValue with other classes.

Table 529 – Association ends of Meas::DiscreteValue with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Discrete	Discrete	Measurement to which this value is connected.
[1..1]	[0..1] RemoteSource	RemoteSource	inherited from: MeasurementValue
[1..1]	[0..1] MeasurementValueQuality	MeasurementValueQualit y	inherited from: MeasurementValue
[0..*]	[1..1] MeasurementValueSource	MeasurementValueSourc e	inherited from: MeasurementValue
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.15 Limit

Specifies one limit value for a Measurement. A Measurement typically has several limits that are kept together by the LimitSet class. The actual meaning and use of a Limit instance (i.e., if it is an alarm or warning limit or if it is a high or low limit) is not captured in the Limit class. However the name of a Limit instance may indicate both meaning and use.

Table 530 shows all attributes of Limit.

Table 530 – Attributes of Meas::Limit

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

Table 531 shows all association ends of Limit with other classes.

Table 531 – Association ends of Meas::Limit with other classes

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

6.15.16 LimitSet

Specifies a set of Limits that are associated with a Measurement. A Measurement may have several LimitSets corresponding to seasonal or other changing conditions. The condition is captured in the name and description attributes. The same LimitSet may be used for several Measurements. In particular percentage limits are used this way.

Table 532 shows all attributes of LimitSet.

Table 532 – Attributes of Meas::LimitSet

name	type	description
isPercentageLimits	Boolean	Tells if the limit values are in percentage of normalValue or the specified Unit for Measurements and Controls.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 533 shows all association ends of LimitSet with other classes.

Table 533 – Association ends of Meas::LimitSet with other classes

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

6.15.17 Measurement

A Measurement represents any measured, calculated or non-measured non-calculated quantity. Any piece of equipment may contain Measurements, e.g. a substation may have temperature measurements and door open indications, a transformer may have oil temperature and tank pressure measurements, a bay may contain a number of power flow measurements and a Breaker may contain a switch status measurement.

The PSR – Measurement association is intended to capture this use of Measurement and is included in the naming hierarchy based on EquipmentContainer. The naming hierarchy typically has Measurements as leafs, e.g. Substation-VoltageLevel-Bay-Switch-Measurement.

Some Measurements represent quantities related to a particular sensor location in the network, e.g. a voltage transformer (PT) at a busbar or a current transformer (CT) at the bar between a breaker and an isolator. The sensing position is not captured in the PSR – Measurement association. Instead it is captured by the Measurement – Terminal association that is used to define the sensing location in the network topology. The location is defined by the connection of the Terminal to ConductingEquipment.

If both a Terminal and PSR are associated, and the PSR is of type ConductingEquipment, the associated Terminal should belong to that ConductingEquipment instance.

When the sensor location is needed both Measurement-PSR and Measurement-Terminal are used. The Measurement-Terminal association is never used alone.

Table 534 shows all attributes of Measurement.

Table 534 – Attributes of Meas::Measurement

name	type	description
measurementType	String	Specifies the type of measurement. For example, this specifies if the measurement represents an indoor temperature, outdoor temperature, bus voltage, line flow, etc.
phases	PhaseCode	Indicates to which phases the measurement applies and avoids the need to use 'measurementType' to also encode phase information (which would explode the types). Since Controls have associations with Measurements, they will have the capability to handle each phase. The phase information in Measurement, along with 'measurementType' and 'phases' uniquely defines a Measurement for a device, based on normal network phase. Their meaning will not change when the computed energizing phasing is changed due to jumpers or other reasons.
unitMultiplier	UnitMultiplier	The unit multiplier of the measured quantity.
unitSymbol	UnitSymbol	The unit of measure of the measured quantity.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 535 shows all association ends of Measurement with other classes.

Table 535 – Association ends of Meas::Measurement with other classes

[mult from]	[mult to] name	type	description
[0..*]	[0..1] Terminal	Terminal	One or more measurements may be associated with a terminal in the network.
[0..*]	[0..1] PowerSystemResource	PowerSystemResource	The power system resource that contains the measurement.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.18 MeasurementValue

The current state for a measurement. A state value is an instance of a measurement from a specific source. Measurements can be associated with many state values, each representing a different source for the measurement.

Table 536 shows all attributes of MeasurementValue.

Table 536 – Attributes of Meas::MeasurementValue

name	type	description
timeStamp	DateTime	The time when the value was last updated
sensorAccuracy	PerCent	The limit, expressed as a percentage of the sensor maximum, that errors will not exceed when the sensor is used under reference conditions.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 537 shows all association ends of MeasurementValue with other classes.

Table 537 – Association ends of Meas::MeasurementValue with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..1] RemoteSource	RemoteSource	Link to the physical telemetered point associated with this measurement.
[1..1]	[0..1] MeasurementValueQuality	MeasurementValueQualit y	A MeasurementValue has a MeasurementValueQuality associated with it.
[0..*]	[1..1] MeasurementValueSource	MeasurementValueSourc e	A reference to the type of source that updates the MeasurementValue, e.g. SCADA, CCLink, manual, etc. User conventions for the names of sources are contained in the introduction to IEC 61970-301.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.19 MeasurementValueQuality

Measurement quality flags. Bits 0-10 are defined for substation automation in IEC 61850-7-3. Bits 11-15 are reserved for future expansion by that document. Bits 16-31 are reserved for EMS applications.

Table 538 shows all attributes of MeasurementValueQuality.

Table 538 – Attributes of Meas::MeasurementValueQuality

name	type	description
badReference	Boolean	inherited from: Quality61850
estimatorReplaced	Boolean	inherited from: Quality61850
failure	Boolean	inherited from: Quality61850
oldData	Boolean	inherited from: Quality61850
operatorBlocked	Boolean	inherited from: Quality61850
oscillatory	Boolean	inherited from: Quality61850
outOfRange	Boolean	inherited from: Quality61850
overFlow	Boolean	inherited from: Quality61850
source	Source	inherited from: Quality61850
suspect	Boolean	inherited from: Quality61850
test	Boolean	inherited from: Quality61850
validity	Validity	inherited from: Quality61850

Table 539 shows all association ends of MeasurementValueQuality with other classes.

Table 539 – Association ends of Meas::MeasurementValueQuality with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] MeasurementValue	MeasurementValue	A MeasurementValue has a MeasurementValueQuality associated with it.

6.15.20 MeasurementValueSource

MeasurementValueSource describes the alternative sources updating a MeasurementValue. User conventions for how to use the MeasurementValueSource attributes are described in the introduction to this standard.

Table 540 shows all attributes of MeasurementValueSource.

Table 540 – Attributes of Meas::MeasurementValueSource

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

Table 541 shows all association ends of MeasurementValueSource with other classes.

Table 541 – Association ends of Meas::MeasurementValueSource with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] MeasurementValues	MeasurementValue	The MeasurementValues updated by the source.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.21 Quality61850 root class

Quality flags in this class are as defined in IEC 61850-7-3, except for estimatorReplaced, which has been included in this class for convenience.

Table 542 shows all attributes of Quality61850.

Table 542 – Attributes of Meas::Quality61850

name	type	description
badReference	Boolean	Measurement value may be incorrect due to a reference being out of calibration.
estimatorReplaced	Boolean	Value has been replaced by State Estimator. estimatorReplaced is not an IEC61850 quality bit but has been put in this class for convenience.
failure	Boolean	This identifier indicates that a supervision function has detected an internal or external failure, e.g. communication failure.
oldData	Boolean	Measurement value is old and possibly invalid, as it has not been successfully updated during a specified time interval.
operatorBlocked	Boolean	Measurement value is blocked and hence unavailable for transmission.
oscillatory	Boolean	To prevent some overload of the communication it is sensible to detect and suppress oscillating (fast changing) binary inputs. If a signal changes in a defined time (tosc) twice in the same direction (from 0 to 1 or from 1 to 0) then oscillation is detected and the detail quality identifier "oscillatory" is set. If it is detected a configured number of transient changes could be passed by. In this time the validity status "questionable" is set. If after this defined numbers of changes the signal is still in the oscillating state the value shall be set either to the opposite state of the previous stable value or to a defined default value. In this case the validity status "questionable" is reset and "invalid" is set as long as the signal is oscillating. If it is configured such that no transient changes should be passed by then the validity status "invalid" is set immediately in addition to the detail quality identifier "oscillatory" (used for status information only).
outOfRange	Boolean	Measurement value is beyond a predefined range of value.
overFlow	Boolean	Measurement value is beyond the capability of being represented properly. For example, a counter value overflows from maximum count back to a value of zero.
source	Source	Source gives information related to the origin of a value. The value may be acquired from the process, defaulted or substituted.
suspect	Boolean	A correlation function has detected that the value is not consitent with other values. Typically set by a network State Estimator.
test	Boolean	Measurement value is transmitted for test purposes.
validity	Validity	Validity of the measurement value.

6.15.22 SetPoint

A SetPoint is an analog control used for supervisory control.

Table 543 shows all attributes of SetPoint.

Table 543 – Attributes of Meas::SetPoint

name	type	description
maxValue	Float	Normal value range maximum for any of the Control.value. Used for scaling, e.g. in bar graphs.
minValue	Float	Normal value range minimum for any of the Control.value. Used for scaling, e.g. in bar graphs.
normalValue	Float	Normal value for Control.value e.g. used for percentage scaling.
value	Float	The value representing the actuator output.
operationInProgress	Boolean	inherited from: Control
timeStamp	DateTime	inherited from: Control
unitMultiplier	UnitMultiplier	inherited from: Control
unitSymbol	UnitSymbol	inherited from: Control
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 544 shows all association ends of SetPoint with other classes.

Table 544 – Association ends of Meas::SetPoint with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..1] Analog	Analog	The Measurement variable used for control.
[1..*]	[0..1] RegulatingCondEq	RegulatingCondEq	inherited from: Control
[1..1]	[0..1] RemoteControl	RemoteControl	inherited from: Control
[0..*]	[1..1] ControlType	ControlType	inherited from: Control
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.23 StringMeasurement

StringMeasurement represents a measurement with values of type string.

Table 545 shows all attributes of StringMeasurement.

Table 545 – Attributes of Meas::StringMeasurement

name	type	description
measurementType	String	inherited from: Measurement
phases	PhaseCode	inherited from: Measurement
unitMultiplier	UnitMultiplier	inherited from: Measurement
unitSymbol	UnitSymbol	inherited from: Measurement
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 546 shows all association ends of StringMeasurement with other classes.

Table 546 – Association ends of Meas::StringMeasurement with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] StringMeasurementValues	StringMeasurementValue	The values connected to this measurement.
[0..*]	[0..1] Terminal	Terminal	inherited from: Measurement
[0..*]	[0..1] PowerSystemResource	PowerSystemResource	inherited from: Measurement
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.24 StringMeasurementValue

StringMeasurementValue represents a measurement value of type string.

Table 547 shows all attributes of StringMeasurementValue.

Table 547 – Attributes of Meas::StringMeasurementValue

name	type	description
value	String	The value to supervise.
timeStamp	DateTime	inherited from: MeasurementValue
sensorAccuracy	PerCent	inherited from: MeasurementValue
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 548 shows all association ends of StringMeasurementValue with other classes.

Table 548 – Association ends of Meas::StringMeasurementValue with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] StringMeasurement	StringMeasurement	Measurement to which this value is connected.
[1..1]	[0..1] RemoteSource	RemoteSource	inherited from: MeasurementValue
[1..1]	[0..1] MeasurementValueQuality	MeasurementValueQuality	inherited from: MeasurementValue
[0..*]	[1..1] MeasurementValueSource	MeasurementValueSource	inherited from: MeasurementValue
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.25 Validity enumeration

Validity for MeasurementValue.

Table 549 shows all literals of Validity.

Table 549 – Literals of Meas::Validity

literal	description
GOOD	The value is marked good if no abnormal condition of the acquisition function or the information source is detected.
QUESTIONABLE	The value is marked questionable if a supervision function detects an abnormal behaviour, however the value could still be valid. The client is responsible for determining whether or not values marked "questionable" should be used.
INVALID	The value is marked invalid when a supervision function recognises abnormal conditions of the acquisition function or the information source (missing or non-operating updating devices). The value is not defined under this condition. The mark invalid is used to indicate to the client that the value may be incorrect and shall not be used.

6.15.26 ValueAliasSet

Describes the translation of a set of values into a name and is intendend to facilitate cusom translations. Each ValueAliasSet has a name, description, etc. A specific Measurement may represent a discrete state like Open, Closed, Intermediate, etc. This requires a translation from the MeasurementValue.value number to a string, e.g. 0 = "Invalid", 1 = "Open", 2 = "Closed", 3 = "Intermediate". Each ValueToAlias member in ValueAliasSet.Value describe a mapping for one particular value to a name.

Table 550 shows all attributes of ValueAliasSet.

Table 550 – Attributes of Meas::ValueAliasSet

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

Table 551 shows all association ends of ValueAliasSet with other classes.

Table 551 – Association ends of Meas::ValueAliasSet with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..*] Discretes	Discrete	The Measurements using the set for translation.
[1..1]	[1..*] Values	ValueToAlias	The ValueToAlias mappings included in the set.
[0..1]	[0..*] Commands	Command	The ValueAliasSet used for translation of a Control value to a name.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.15.27 ValueToAlias

Describes the translation of one particular value into a name, e.g. 1 as "Open".

Table 552 shows all attributes of ValueToAlias.

Table 552 – Attributes of Meas::ValueToAlias

name	type	description
value	Integer	The value that is mapped.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 553 shows all association ends of ValueToAlias with other classes.

Table 553 – Association ends of Meas::ValueToAlias with other classes

[mult from]	[mult to] name	type	description
[1..*]	[1..1] ValueAliasSet	ValueAliasSet	The ValueAliasSet having the ValueToAlias mappings.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.16 Package SCADA

6.16.1 General

Contains entities to model information used by Supervisory Control and Data Acquisition (SCADA) applications. Supervisory control supports operator control of equipment, such as opening or closing a breaker. Data acquisition gathers telemetered data from various sources. The subtypes of the Telemetry entity deliberately match the UCA and IEC 61850 definitions.

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

Figure 80 shows class diagram Datatypes.

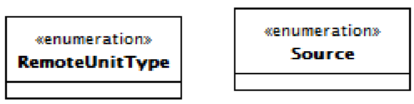

Figure 80 – Class diagram SCADA::Datatypes

Shows datatypes for SCADA model.

Figure 81 shows class diagram Main.

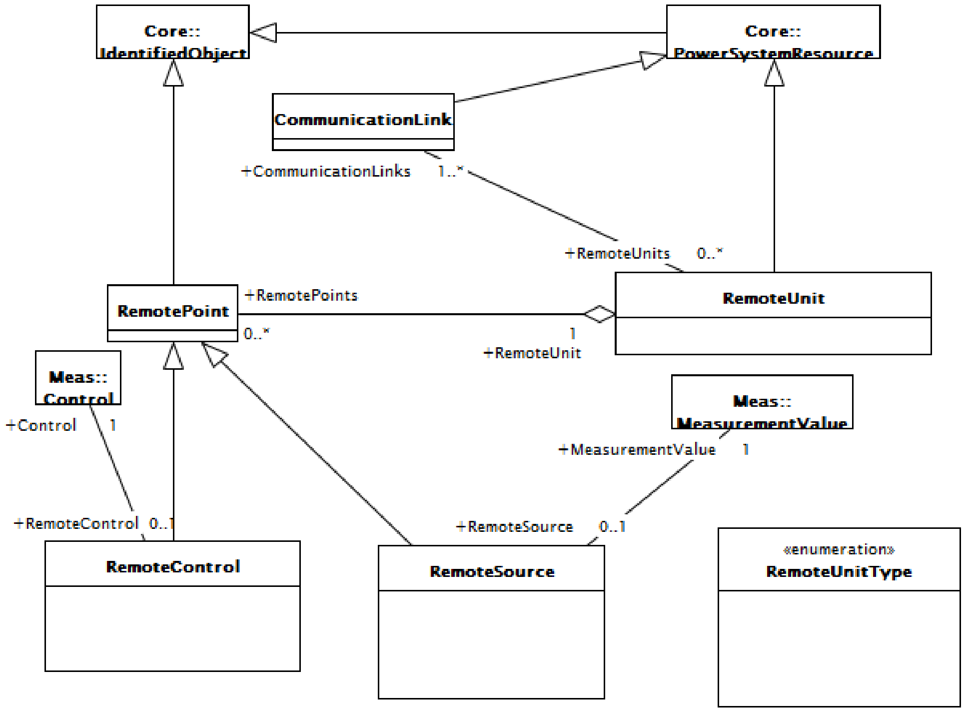

Figure 81 – Class diagram SCADA::Main

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

6.16.2 CommunicationLink

The connection to remote units is through one or more communication links. Reduntant links may exist. The CommunicationLink class inherit PowerSystemResource. The intention is to allow CommunicationLinks to have Measurements. These Measurements can be used to model link status as operational, out of service, unit failure, etc.

Table 554 shows all attributes of CommunicationLink.

Table 554 – Attributes of SCADA::CommunicationLink

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

Table 555 shows all association ends of CommunicationLink with other classes.

Table 555 – Association ends of SCADA::CommunicationLink with other classes

[mult from]	[mult to] name	type	description
[1..*]	[0..*] RemoteUnits	RemoteUnit	RTUs may be attached to communication links.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.16.3 RemoteControl

Remote controls are ouputs that are sent by the remote unit to actuators in the process.

Table 556 shows all attributes of RemoteControl.

Table 556 – Attributes of SCADA::RemoteControl

name	type	description
remoteControlled	Boolean	Set to true if the actuator is remotely controlled.
actuatorMaximum	Float	The maximum set point value accepted by the remote control point.
actuatorMinimum	Float	The minimum set point value accepted by the remote control point.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 557 shows all association ends of RemoteControl with other classes.

Table 557 – Association ends of SCADA::RemoteControl with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] Control	Control	The Control for the RemoteControl point.
[0..*]	[1..1] RemoteUnit	RemoteUnit	inherited from: RemotePoint
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.16.4 RemotePoint

For a RTU remote points correspond to telemetered values or control outputs. Other units (e.g. control centers) usually also contain calculated values.

Table 558 shows all attributes of RemotePoint.

Table 558 – Attributes of SCADA::RemotePoint

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

Table 559 shows all association ends of RemotePoint with other classes.

Table 559 – Association ends of SCADA::RemotePoint with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] RemoteUnit	RemoteUnit	Remote unit this point belongs to.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.16.5 RemoteSource

Remote sources are state variables that are telemetered or calculated within the remote unit.

Table 560 shows all attributes of RemoteSource.

Table 560 – Attributes of SCADA::RemoteSource

name	type	description
sensorMaximum	Float	The maximum value the telemetry item can return.
sensorMinimum	Float	The minimum value the telemetry item can return.
scanInterval	Seconds	The time interval between scans.
deadband	Float	The smallest change in value to be reported.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 561 shows all association ends of RemoteSource with other classes.

Table 561 – Association ends of SCADA::RemoteSource with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] MeasurementValue	MeasurementValue	Link to the physical telemetered point associated with this measurement.
[0..*]	[1..1] RemoteUnit	RemoteUnit	inherited from: RemotePoint
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.16.6 RemoteUnit

A remote unit can be a RTU, IED, substation control system, control center, etc. The communication with the remote unit can be through various standard protocols (e.g. IEC 61870, IEC 61850) or non standard protocols (e.g. DNP, RP570, etc.). A remote unit contains remote data points that might be telemetered, collected or calculated. The RemoteUnit class inherits from PowerSystemResource. The intention is to allow RemotUnits to have Measurements. These Measurements can be used to model unit status as operational, out of service, unit failure, etc.

Table 562 shows all attributes of RemoteUnit.

Table 562 – Attributes of SCADA::RemoteUnit

name	type	description
remoteUnitType	RemoteUnitType	Type of remote unit.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 563 shows all association ends of RemoteUnit with other classes.

Table 563 – Association ends of SCADA::RemoteUnit with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..*] CommunicationLinks	CommunicationLink	RTUs may be attached to communication links.
[1..1]	[0..*] RemotePoints	RemotePoint	Remote points this Remote unit contains.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.16.7 RemoteUnitType enumeration

Type of remote unit.

Table 564 shows all literals of RemoteUnitType.

Table 564 – Literals of SCADA::RemoteUnitType

literal	description
RTU	Remote terminal unit.
SubstationControlSystem	Substation control system.
ControlCenter	Control center.
IED	Intelligent electronic device (IED).

6.16.8 Source enumeration

Source gives information related to the origin of a value.

Table 565 shows all literals of Source.

Table 565 – Literals of SCADA::Source

literal	description
PROCESS	The value is provided by input from the process I/O or calculated from some function.
DEFAULTED	The value contains a default value.
SUBSTITUTED	The value is provided by input of an operator or by an automatic source.

6.17 Package ControlArea

6.17.1 General

The ControlArea package models area specifications which can be used for a variety of purposes. The package as a whole models potentially overlapping control area specifications for the purpose of actual generation control, load forecast area load capture, or powerflow based analysis.

Figure 82 shows class diagram ControlArea.

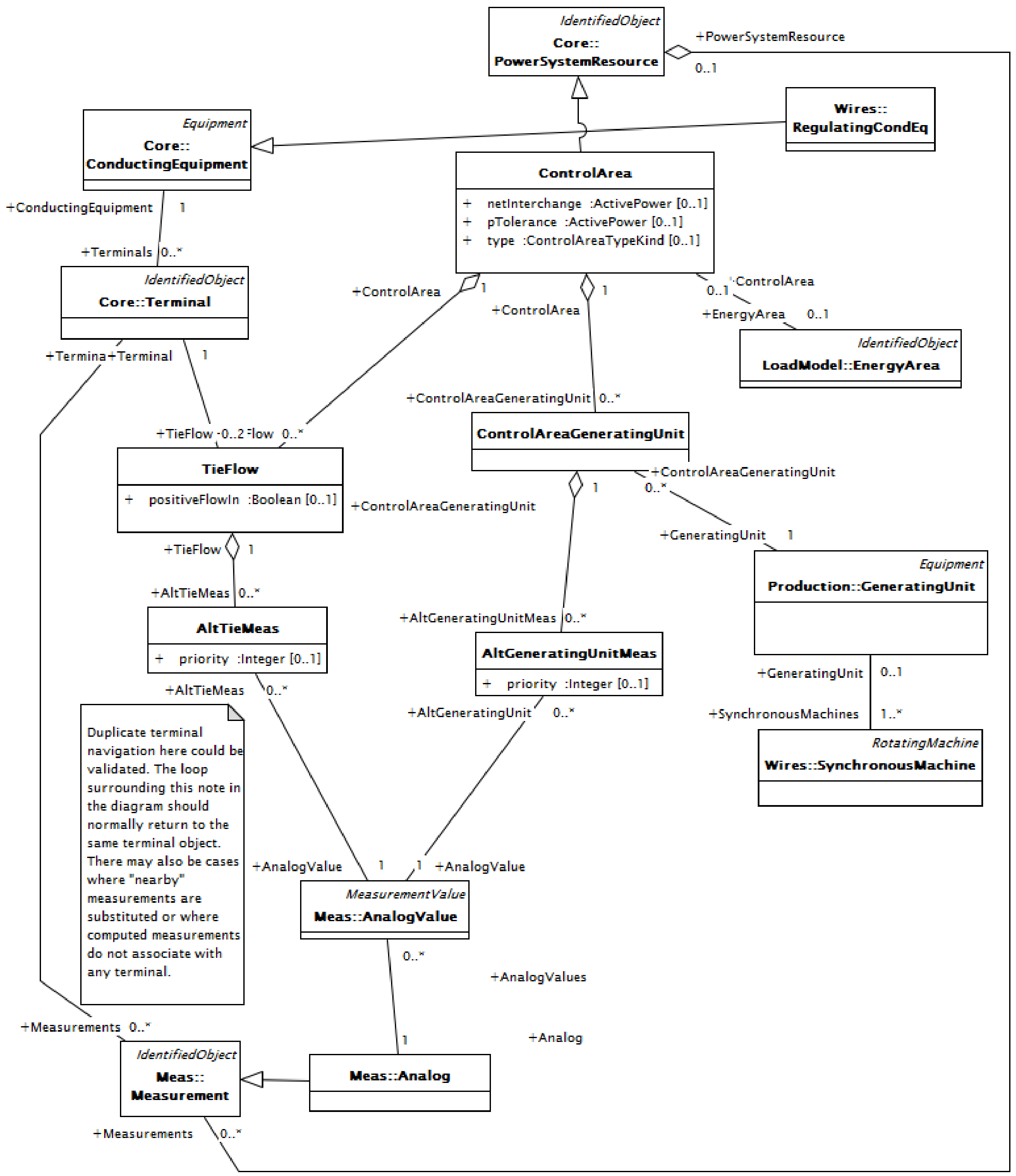

Figure 82 – Class diagram ControlArea::ControlArea

This diagram shows control area specification and some related classes. The Terminal to AnalogValue linkages are shown for clarity in understanding the control area specificaiton. The GeneratingUnit to Terminal linkages are also show to illustrate how generation flows are specifically tied to the network.

Figure 83 shows class diagram ControlAreaInheritance.

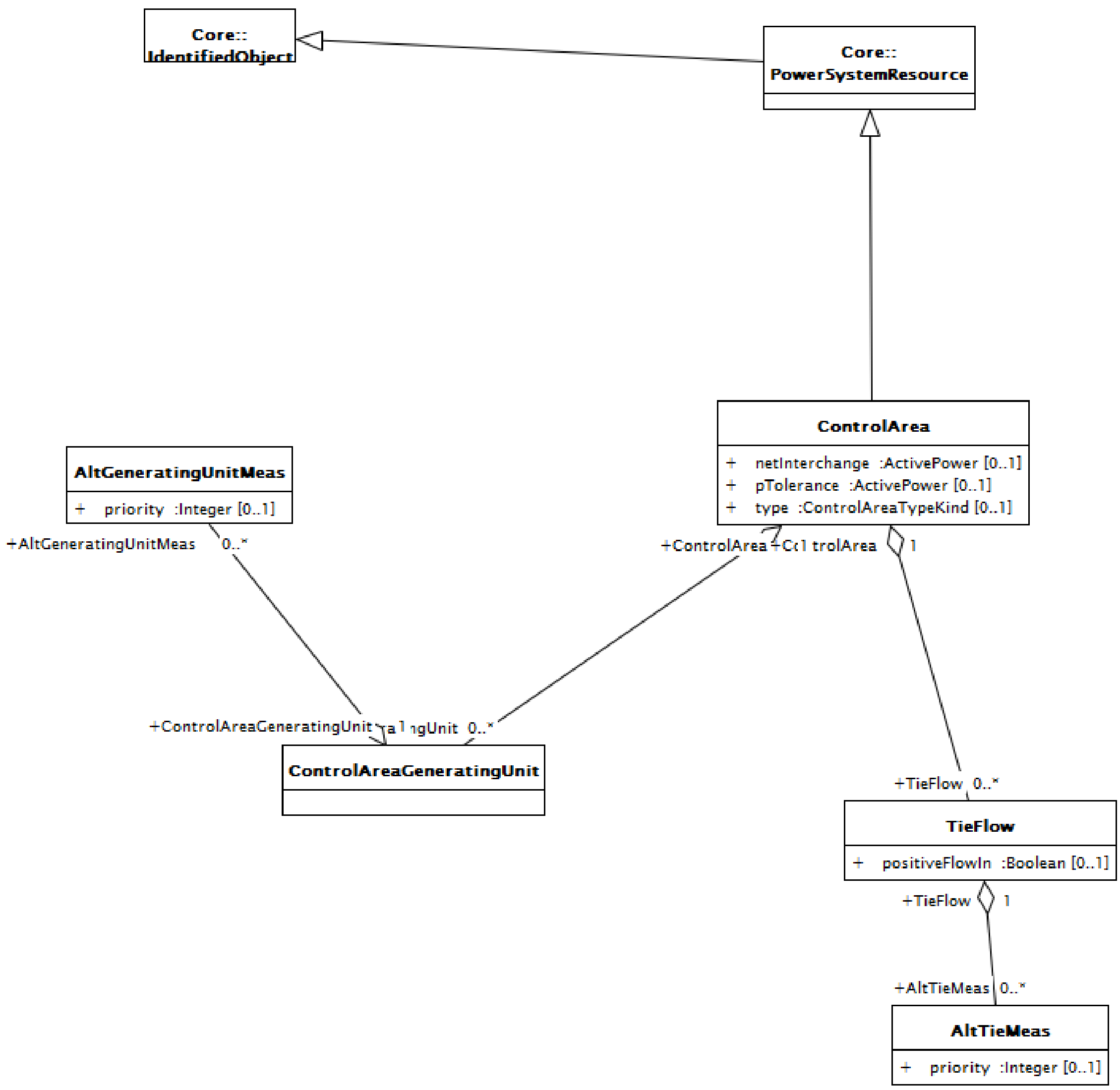

Figure 83 – Class diagram ControlArea::ControlAreaInheritance

Shows the inheritance of classes in this package.

Figure 84 shows class diagram Datatypes.

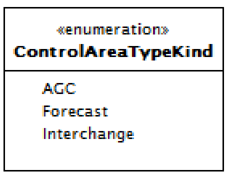

Figure 84 – Class diagram ControlArea::Datatypes

Shows datatypes for control areas.

6.17.2 AltGeneratingUnitMeas root class

A prioritized measurement to be used for the generating unit in the control area specificaiton.

Table 566 shows all attributes of AltGeneratingUnitMeas.

Table 566 – Attributes of ControlArea::AltGeneratingUnitMeas

name	type	description
priority	Integer	Priority of a measurement usage. Lower numbers have first priority.

Table 567 shows all association ends of AltGeneratingUnitMeas with other classes.

Table 567 – Association ends of ControlArea::AltGeneratingUnitMeas with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] AnalogValue	AnalogValue	The specific analog value used as a source.
[0..*]	[1..1] ControlAreaGeneratingUnit	ControlAreaGeneratingUnit	The control area generating unit to which the prioritized measurement assignment is applied.

6.17.3 AltTieMeas root class

A prioritized measurement to be used for the tie flow as part of the control area specification.

Table 568 shows all attributes of AltTieMeas.

Table 568 – Attributes of ControlArea::AltTieMeas

name	type	description
priority	Integer	Priority of a measurement usage. Lower numbers have first priority.

Table 569 shows all association ends of AltTieMeas with other classes.

Table 569 – Association ends of ControlArea::AltTieMeas with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] AnalogValue	AnalogValue	The specific analog value used as a source.
[0..*]	[1..1] TieFlow	TieFlow	The tie flow of the alternate measurements.

6.17.4 ControlArea

A control area is a grouping of generating units and/or loads and a cutset of tie lines (as terminals) which may be used for a variety of purposes including automatic generation control, powerflow solution area interchange control specification, and input to load forecasting. Note that any number of overlapping control area specifications can be superimposed on the physical model.

Table 570 shows all attributes of ControlArea.

Table 570 – Attributes of ControlArea::ControlArea

name	type	description
netInterchange	ActivePower	The specified positive net interchange into the control area.
pTolerance	ActivePower	Active power net interchange tolerance
type	ControlAreaTypeKind	The type of control area defintion used to determine if this is used for automatic generation control, for planning interchange control, or other purposes.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 571 shows all association ends of ControlArea with other classes.

Table 571 – Association ends of ControlArea::ControlArea with other classes

[mult from]	[mult to] name	type	description
[0..1]	[0..1] EnergyArea	EnergyArea	The energy area that is forecast from this control area specification.
[1..1]	[0..*] ControlAreaGeneratingUnit	ControlAreaGeneratingUnit	The generating unit specificaitons for the control area.
[1..1]	[0..*] TieFlow	TieFlow	The tie flows associated with the control area.
[0..*]	[0..1] PSRType	PSRType	inherited from: PowerSystemResource
[0..1]	[0..*] Measurements	Measurement	inherited from: PowerSystemResource
[1..1]	[0..*] OperatingShare	OperatingShare	inherited from: PowerSystemResource
[0..*]	[0..*] PsrLists	PsrList	inherited from: PowerSystemResource
[1..1]	[0..1] OutageSchedule	OutageSchedule	inherited from: PowerSystemResource
[0..*]	[0..*] ReportingGroup	ReportingGroup	inherited from: PowerSystemResource
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.17.5 ControlAreaGeneratingUnit root class

A control area generating unit. This class is needed so that alternate control area definitions may include the same generating unit. Note only one instance within a control area should reference a specific generating unit.

Table 572 shows all association ends of ControlAreaGeneratingUnit with other classes.

Table 572 – Association ends of ControlArea::ControlAreaGeneratingUnit with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] GeneratingUnit	GeneratingUnit	The generating unit specified for this control area. Note that a control area should include a GeneratingUnit only once.
[0..*]	[1..1] ControlArea	ControlArea	The parent control area for the generating unit specifications.
[1..1]	[0..*] AltGeneratingUnitMeas	AltGeneratingUnitMeas	The link to prioritized measurements for this GeneratingUnit.

6.17.6 ControlAreaTypeKind enumeration

The type of control area.

Table 573 shows all literals of ControlAreaTypeKind.

Table 573 – Literals of ControlArea::ControlAreaTypeKind

literal	description
AGC	Used for automatic generatoin control.
Forecast	Used for load forecast.
Interchange	Used for interchange specification or control.

6.17.7 TieFlow root class

A flow specification in terms of location and direction for a control area.

Table 574 shows all attributes of TieFlow.

Table 574 – Attributes of ControlArea::TieFlow

name	type	description
positiveFlowIn	Boolean	The flow is positive into the terminal. A flow is positive if it is an import into the control area.

Table 575 shows all association ends of TieFlow with other classes.

Table 575 – Association ends of ControlArea::TieFlow with other classes

[mult from]	[mult to] name	type	description
[0..2]	[1..1] Terminal	Terminal	The terminal to which this tie flow belongs.
[0..*]	[1..1] ControlArea	ControlArea	The control area of the tie flows.
[1..1]	[0..*] AltTieMeas	AltTieMeas	The primary and alternate tie flow measurements associated with the tie flow.

6.18 Package Contingency

6.18.1 General

Contingencies to be studied.

Figure 85 shows class diagram Contingency.

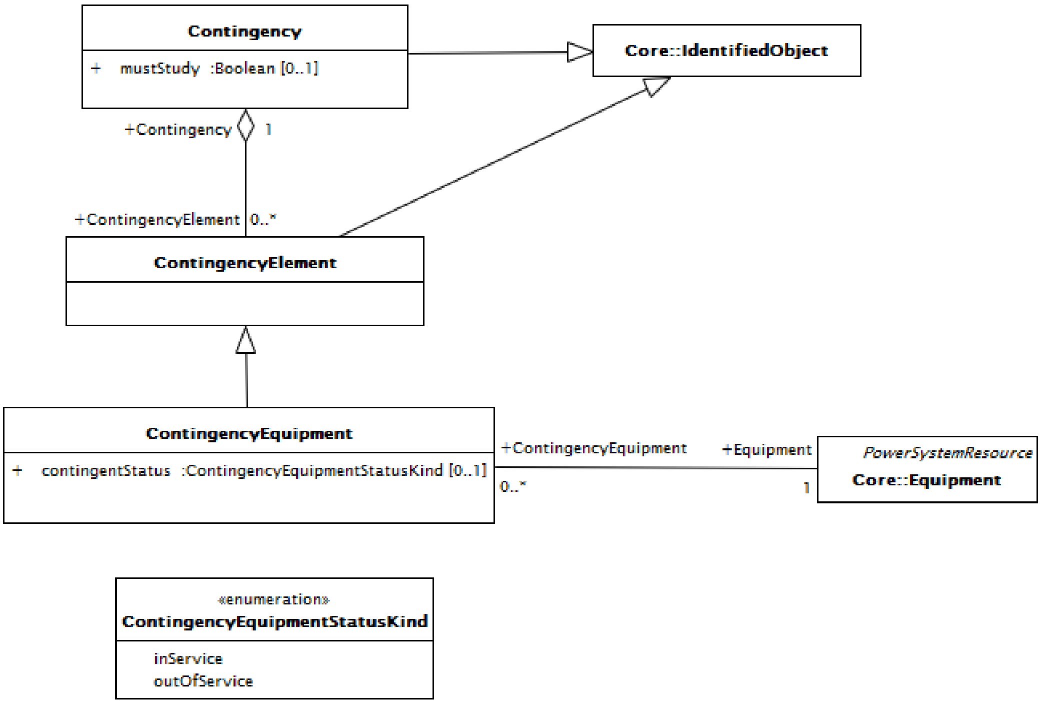

Figure 85 – Class diagram Contingency::Contingency

This diagram shows contingency model inheritance and associations.

6.18.2 Contingency

An event threatening system reliability, consisting of one or more contingency elements.

Table 576 shows all attributes of Contingency.

Table 576 – Attributes of Contingency::Contingency

name	type	description
mustStudy	Boolean	Set true if must study this contingency.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 577 shows all association ends of Contingency with other classes.

Table 577 – Association ends of Contingency::Contingency with other classes

[mult from]	[mult to] name	type	description
[1..1]	[0..*] ContingencyElement	ContingencyElement	A contingency can have any number of contingency elements.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.18.3 ContingencyElement

An element of a system event to be studied by contingency analysis, representing a change in status of a single piece of equipment.

Table 578 shows all attributes of ContingencyElement.

Table 578 – Attributes of Contingency::ContingencyElement

name	type	description
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 579 shows all association ends of ContingencyElement with other classes.

Table 579 – Association ends of Contingency::ContingencyElement with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Contingency	Contingency	A contingency element belongs to one contingency.
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.18.4 ContingencyEquipment

A equipment to which the in service status is to change such as a power transformer or AC line segment.

Table 580 shows all attributes of ContingencyEquipment.

Table 580 – Attributes of Contingency::ContingencyEquipment

name	type	description
contingentStatus	ContingencyEquipmentStatusKind	The status for the associated equipment when in the contingency state. This status is independent of the case to which the contingency is originally applied, but defines the equipment status when the contingency is applied.
aliasName	String	inherited from: IdentifiedObject
mRID	String	inherited from: IdentifiedObject
name	String	inherited from: IdentifiedObject

Table 581 shows all association ends of ContingencyEquipment with other classes.

Table 581 – Association ends of Contingency::ContingencyEquipment with other classes

[mult from]	[mult to] name	type	description
[0..*]	[1..1] Equipment	Equipment	The single piece of equipment to which to apply the contingency.
[0..*]	[1..1] Contingency	Contingency	inherited from: ContingencyElement
[0..1]	[0..*] DiagramObjects	DiagramObject	inherited from: IdentifiedObject
[1..1]	[0..*] Names	Name	inherited from: IdentifiedObject

6.18.5 ContingencyEquipmentStatusKind enumeration

Indicates the state which the contingency equipment is to be in when the contingency is applied.

Table 582 shows all literals of ContingencyEquipmentStatusKind.

Table 582 – Literals of Contingency::ContingencyEquipmentStatusKind

literal	description
inService	The equipment is in service.
outOfService	The equipment is to be taken out of service.

6.19 Package StateVariables

6.19.1 General

State variables for analysis solutions such as powerflow.

Figure 86 shows class diagram StateVariables.

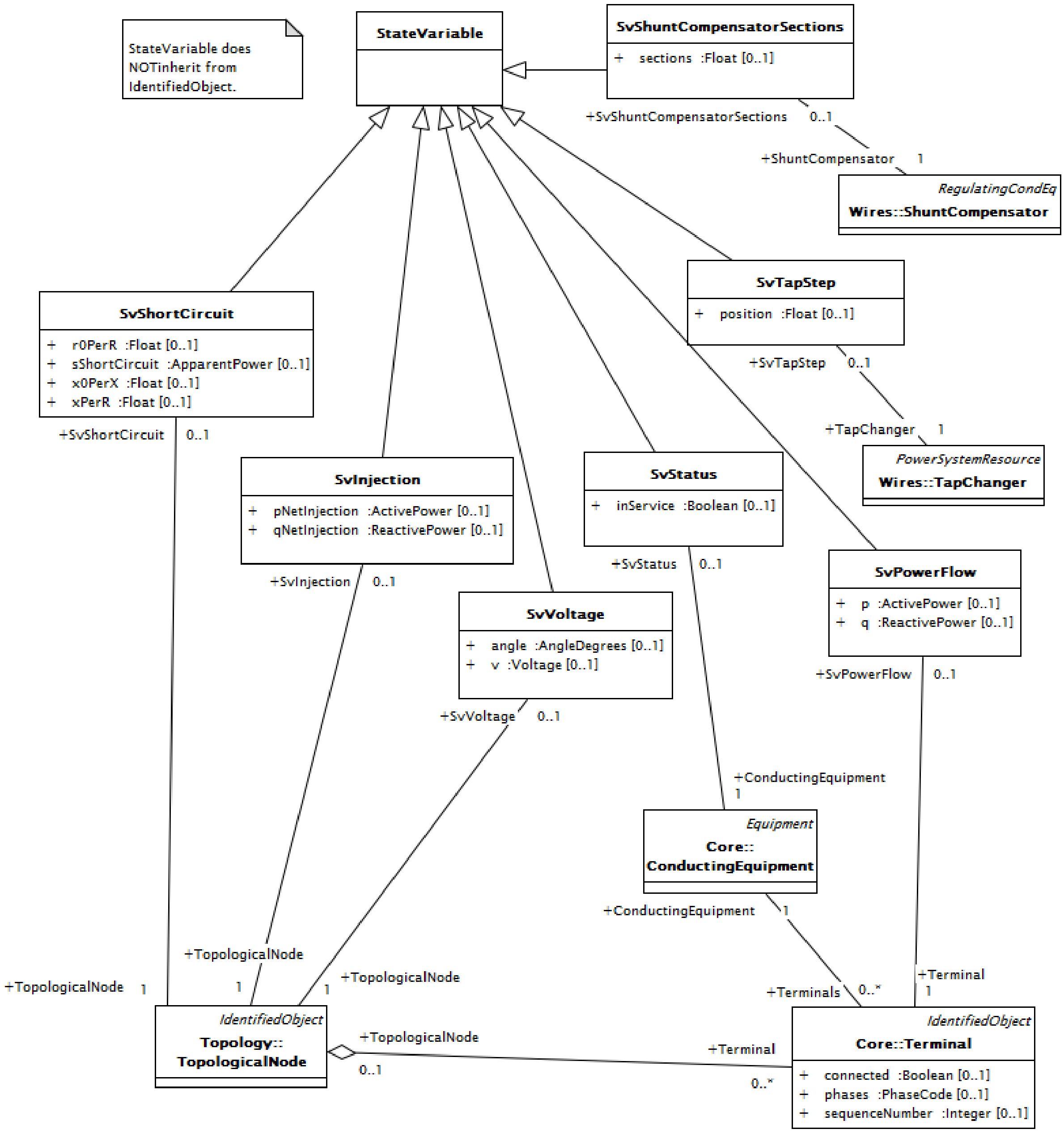

Figure 86 – Class diagram StateVariables::StateVariables

The state solution variables as derived from base StateVariable class. Note that specialized state variables are associated with objects derived from IdentifiedObject, but StateVariable does not derive from IdentifiedObject.

6.19.2 StateVariable root class

An abstract class for state variables.

6.19.3 SvInjection

Injection state variable. Used for addressing situations where the exchanged models have a Kirchof's law mismatch at a bus. This includes exchange of partial models with boundary flows and state estimator solutions with residual mismatch.

Table 583 shows all attributes of SvInjection.

Table 583 – Attributes of StateVariables::SvInjection

name	type	description
pNetInjection	ActivePower	The active power injected into the bus at this location. Positive sign means injection into the bus.
qNetInjection	ReactivePower	The reactive power injected into the bus at this location. Positive sign means injection into the bus.

Table 584 shows all association ends of SvInjection with other classes.

Table 584 – Association ends of StateVariables::SvInjection with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] TopologicalNode	TopologicalNode	The topological node associated with the flow injection state variable.

6.19.4 SvPowerFlow

State variable for power flow. Load convention is used for flow direction. This means flow out from the TopologicalNode into the equipment is positive.

Table 585 shows all attributes of SvPowerFlow.

Table 585 – Attributes of StateVariables::SvPowerFlow

name	type	description
p	ActivePower	The active power flow. Load sign convention is used, i.e. positive sign means flow out from a node.
q	ReactivePower	The reactive power flow. Load sign convention is used, i.e. positive sign means flow out from a node.

Table 586 shows all association ends of SvPowerFlow with other classes.

Table 586 – Association ends of StateVariables::SvPowerFlow with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] Terminal	Terminal	The terminal associated with the power flow state variable.

6.19.5 SvShortCircuit

State variable for short circuit.

Table 587 shows all attributes of SvShortCircuit.

Table 587 – Attributes of StateVariables::SvShortCircuit

name	type	description
r0PerR	Float	The ratio of zero sequence resistance to positive sequence resistance.
sShortCircuit	ApparentPower	The short circuit apparent power drawn at this node when faulted.
x0PerX	Float	The ratio of zero sequence reactance per positive sequence reactance.
xPerR	Float	Ratio of positive sequence reactance per postive sequence resistance.

Table 588 shows all association ends of SvShortCircuit with other classes.

Table 588 – Association ends of StateVariables::SvShortCircuit with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] TopologicalNode	TopologicalNode	The topological node associated with the state variables for short circuit.

6.19.6 SvShuntCompensatorSections

State variable for the number of sections in service for a shunt compensator.

Table 589 shows all attributes of SvShuntCompensatorSections.

Table 589 – Attributes of StateVariables::SvShuntCompensatorSections

name	type	description
sections	Float	The number of sections in service as a continous variable. To get integer value scale with ShuntCompensator.bPerSection.

Table 590 shows all association ends of SvShuntCompensatorSections with other classes.

Table 590 – Association ends of StateVariables::SvShuntCompensatorSections with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] ShuntCompensator	ShuntCompensator	The shunt compensator for which the state applies.

6.19.7 SvStatus

State variable for status.

Table 591 shows all attributes of SvStatus.

Table 591 – Attributes of StateVariables::SvStatus

name	type	description
inService	Boolean	The in service status as a result of topology processing.

Table 592 shows all association ends of SvStatus with other classes.

Table 592 – Association ends of StateVariables::SvStatus with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] ConductingEquipment	ConductingEquipment	The conducting equipment associated with the status state variable.

6.19.8 SvTapStep

State variable for transformer tap step. This class is to be used for taps of LTC (load tap changing) transformers, not fixed tap transformers. Normally a profile specifies only one of the attributes "position"or "tapRatio".

Table 593 shows all attributes of SvTapStep.

Table 593 – Attributes of StateVariables::SvTapStep

name	type	description
position	Float	The floating point tap position. To get integer value scale with range and round off.

Table 594 shows all association ends of SvTapStep with other classes.

Table 594 – Association ends of StateVariables::SvTapStep with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] TapChanger	TapChanger	The tap changer associated with the tap step state.

6.19.9 SvVoltage

State variable for voltage.

Table 595 shows all attributes of SvVoltage.

Table 595 – Attributes of StateVariables::SvVoltage

name	type	description
angle	AngleDegrees	The voltage angle of the topological node complex voltage with respect to system reference.
v	Voltage	The voltage magnitude of the topological node.

Table 596 shows all association ends of SvVoltage with other classes.

Table 596 – Association ends of StateVariables::SvVoltage with other classes

[mult from]	[mult to] name	type	description
[0..1]	[1..1] TopologicalNode	TopologicalNode	The topological node associated with the voltage state.

Bibliography

IEC 61968-11, *Application integration at electric utilities – System interfaces for distribution management – Part 11: Common Information Model (CIM) Extensions for Distribution*

IEC 61970-501, *Energy management system application program interface (EMS-API) – Part 501: Common Information Model Resource Description Framework (CIM RDF) schema*

ISO 8601:2004, *Data elements and interchange formats – Information interchange – Representation of dates and times*

RP-3654-1 EPRI Control Center API (CCAPI) research project.

CIMTool – Open source CIM profile management tool. http://www.cimtool.org

CIM Users Group – The community of CIM users. http://cimug.ucaiug.org

DNP (Distributed Network Protocol): Series of standards. http://www.dnp.org/

RP570 (RTU Protocol based on IEC 57 part 5-1 (present IEC 870)). http://library.abb.com/GLOBAL/SCOT/scot229.NSF/VerityDisplay/9A5C1896695487E6C2256A7200361578/$File/REC501RP570_EN_A.pdf

W3C: RDF/XML Syntax Specification http://www.w3.org/RDF/

W3C: Extensible Markup Language (XML) 1.0 http://www.w3.org/XML/

This page deliberately left blank

This page deliberately left blank

British Standards Institution (BSI)

BSI is the national body responsible for preparing British Standards and other standards-related publications, information and services.

BSI is incorporated by Royal Charter. British Standards and other standardization products are published by BSI Standards Limited.

About us

We bring together business, industry, government, consumers, innovators and others to shape their combined experience and expertise into standards -based solutions.

The knowledge embodied in our standards has been carefully assembled in a dependable format and refined through our open consultation process. Organizations of all sizes and across all sectors choose standards to help them achieve their goals.

Information on standards

We can provide you with the knowledge that your organization needs to succeed. Find out more about British Standards by visiting our website at bsigroup.com/standards or contacting our Customer Services team or Knowledge Centre.

Buying standards

You can buy and download PDF versions of BSI publications, including British and adopted European and international standards, through our website at bsigroup.com/shop, where hard copies can also be purchased.

If you need international and foreign standards from other Standards Development Organizations, hard copies can be ordered from our Customer Services team.

Subscriptions

Our range of subscription services are designed to make using standards easier for you. For further information on our subscription products go to bsigroup.com/subscriptions.

With **British Standards Online (BSOL)** you'll have instant access to over 55,000 British and adopted European and international standards from your desktop. It's available 24/7 and is refreshed daily so you'll always be up to date.

You can keep in touch with standards developments and receive substantial discounts on the purchase price of standards, both in single copy and subscription format, by becoming a **BSI Subscribing Member**.

PLUS is an updating service exclusive to BSI Subscribing Members. You will automatically receive the latest hard copy of your standards when they're revised or replaced.

To find out more about becoming a BSI Subscribing Member and the benefits of membership, please visit bsigroup.com/shop.

With a **Multi-User Network Licence (MUNL)** you are able to host standards publications on your intranet. Licences can cover as few or as many users as you wish. With updates supplied as soon as they're available, you can be sure your documentation is current. For further information, email bsmusales@bsigroup.com.

Revisions

Our British Standards and other publications are updated by amendment or revision.

We continually improve the quality of our products and services to benefit your business. If you find an inaccuracy or ambiguity within a British Standard or other BSI publication please inform the Knowledge Centre.

Copyright

Useful Contacts:

Customer Services
Tel: +44 845 086 9001
Email (orders): orders@bsigroup.com
Email (enquiries): cservices@bsigroup.com

Subscriptions
Tel: +44 845 086 9001
Email: subscriptions@bsigroup.com

Knowledge Centre
Tel: +44 20 8996 7004
Email: knowledgecentre@bsigroup.com

Copyright & Licensing
Tel: +44 20 8996 7070
Email: copyright@bsigroup.com

BSI Group Headquarters

389 Chiswick High Road London W4 4AL UK

bsi.

...making excellence a habit.™