Unified Modeling Language (UML) Fundamentals & Intro to Profiles

Margaret Goodrich
Cell: +1-903-477-7176
E-Mail: margaret@pcitek.com

Topics

- Information Modeling Methodology & Definitions
- UML Notation – A diagramming tool
  - Definitions
  - Introduction
- Generalization, Specialization, Inheritance
- Associations/Relationships
  - Simple
  - Aggregation/Composition
- UML Conventions
- Introduction to Profiles
- Questions

IM Methodology

- Information Modeling is a disciplined approach to the development of industrial grade software based on the object oriented paradigm.
- Projects using this methodology have produced manufacturing and process control applications, intelligent instruments and peripherals, telecommunications and defense applications.

PCItek
TECHNOLOGY - EXCELLENCE - KNOWLEDGE

IM Methodology

- An abstract representation of real-world objects
- Gives unique names, definition and meaning to each object to avoid confusion
- Describes relationships between objects
- Not tied to a particular application's view of the world
  - But permits the same model to be used by all applications to facilitate information sharing between applications
  - Provides consistent view of the world by operators regardless of which application user interface they are using

IM Components

- An Information Model is composed of Packages that describe distinct subjects.
- Each of the Packages describes the classes and the relationship of the classes for that subject.
- A Class describes an object, its properties or attributes and the relationships with other objects or classes
- An attribute is an element of the class. For example, the r value is an attribute of an AC Line Segment
- A relationship defines the association of the class to other classes or attributes:
  - transformers are contained within substations
  - transformers have names, voltages, ratings, etc.

UML Notation

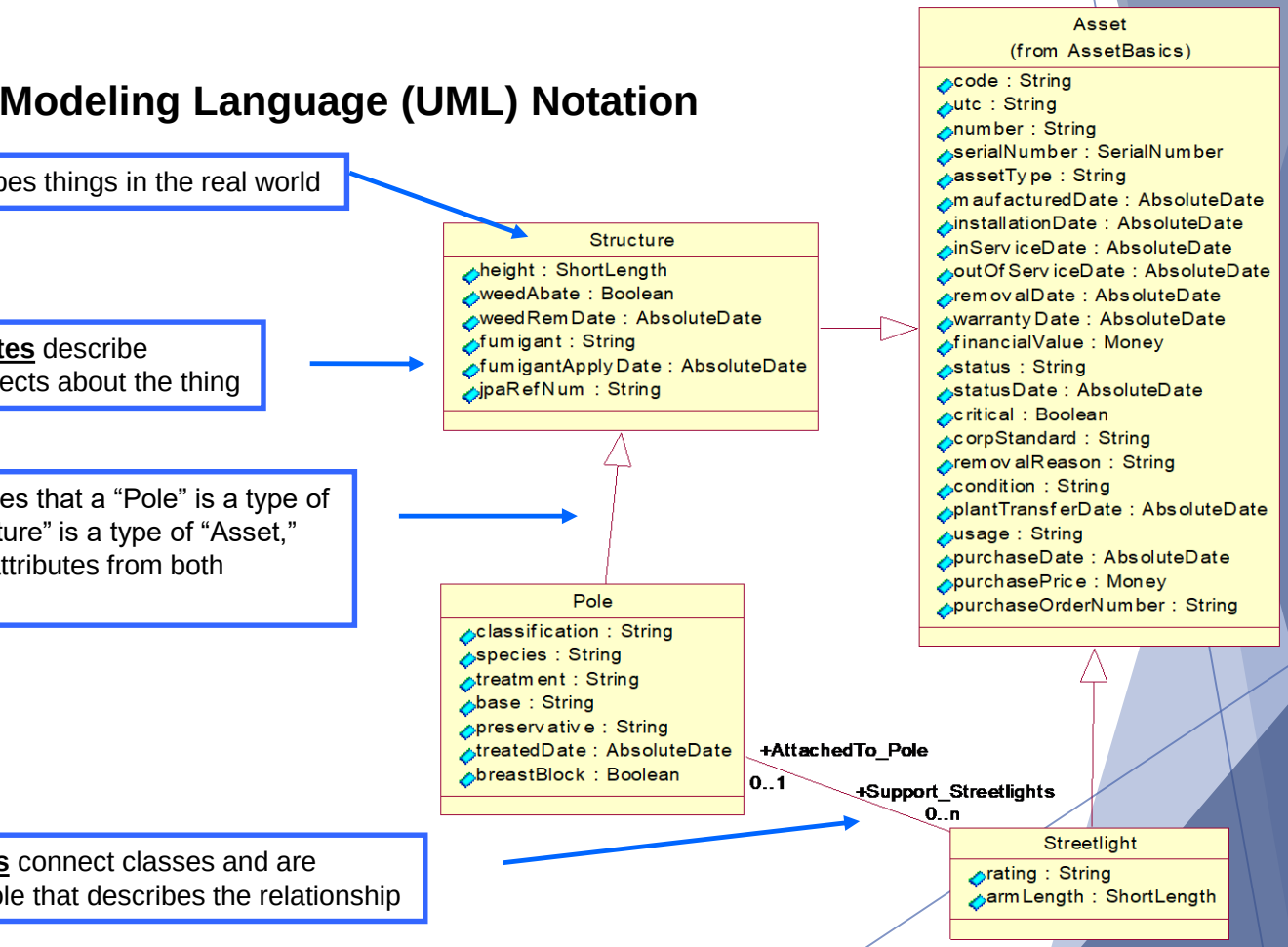
A Bit On Unified Modeling Language (UML) Notation

Class Name usually describes things in the real world

Class Attributes describe significant aspects about the thing

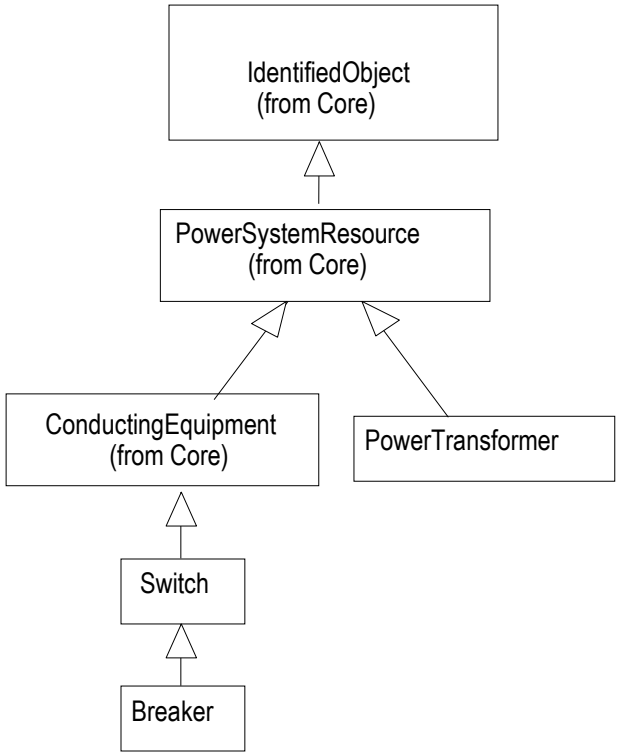
This **Specialization** indicates that a "Pole" is a type of "Structure." Since a "Structure" is a type of "Asset," the Pole inherits all of the attributes from both Structure and Asset

Associations connect classes and are assigned a role that describes the relationship



PCItek
TECHNOLOGY - EXCELLENCE - KNOWLEDGE

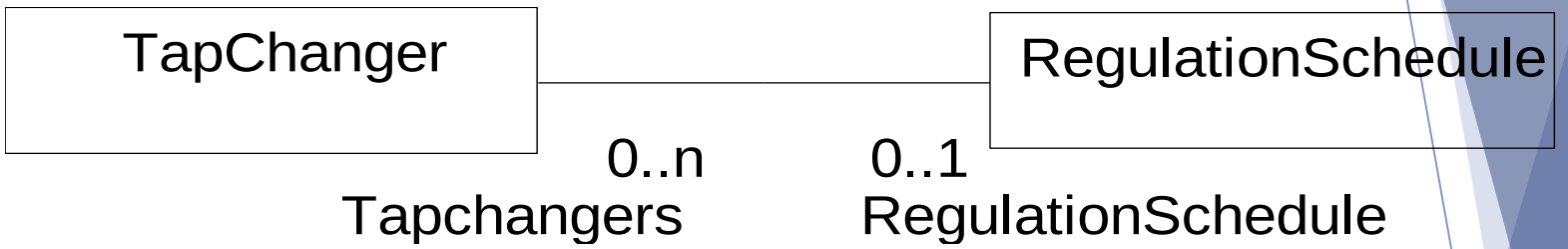
UML Concepts:



- Breaker: Specialization of Switch
- Switch: Specialization of Conducting Equipment
- ConductingEquipment: Specialization of PowerSystem Resource
- Going is reverse, these are referred to as Generalizations
- Each specialization inherits all the attributes and associations of the generalized object above it.

PCItek
TECHNOLOGY - EXCELLENCE - KNOWLEDGE

UML Concepts: Simple Association (aka Relationship)

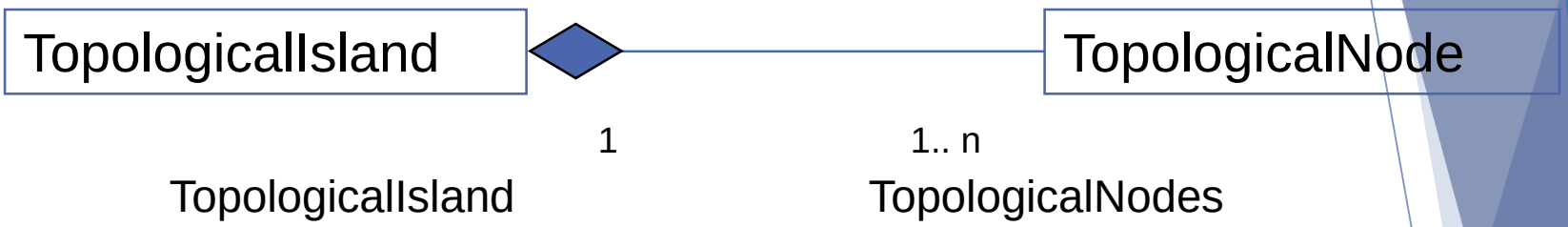


- Tap Changer has 0 or 1 regulation schedules
- Regulation Schedule applies to 0 to n Tap Changers

This is a one-to-many relationship/association

PCItek
TECHNOLOGY - EXCELLENCE - KNOWLEDGE

UML Concepts: Aggregation/Composition (AKA Containment)



A Topological Island comprises 1 to n Topological Nodes
Topological Nodes are Contained within Topological Islands
(This is how a bus is defined in the CIM)

Black Diamond – Composition

Child cannot exist without parent

White Diamond – Aggregation

Child can exist without parent

Diamonds always point to the parent or "whole" part of the relationship. Implied multiplicity of Parent is 0..1 unless specified as 1. The Child is fully contained within the parent – it is a "part" of the whole.

PCItek
TECHNOLOGY - EXCELLENCE - KNOWLEDGE

Naming Conventions

- Use English language for names
- Names for packages, classes and association ends start with upper case letter
- Names for attributes and enumeration literals start with lower case letter
- Use camel notation (= 'attributeName' or 'ClassName')
- Try to avoid underscore
- Never use space in these tokens
- Refer to sub-packages other than Informative under IEC 61968

UML Element Conventions

- Packages
  - Before creating a package, search for intended package name and add a prefix if the package already exists
  - Create a package that is peer (sibling) to the IEC* packages, e.g., MyPackage
  - Create a class within that package called MyPackageCIMVersion and copy date and version attributes from IEC61970CIMVersion into it and update the values to fit your context
  - If there are a large number of class extensions, create sub-packages within MyPackage
  - Document each package with a sentence or two

UML Element Conventions

- Classes
  - before creating a class, search for the intended class name and add a prefix to the extension if a class already exists
  - class names start with upper case and should not end with 's‘
  - document the purpose of the class
  - Use TagValues or Stereotypes: enumeration, Datatype or Compound (you should not be defining other Primitive types than standard ones)
  - Stereotyped classes are types that never participate in relationships (i.e., no associations, no inheritance), but are used as types for attributes

UML Element Conventions

- Classes (Continued)
  - Datatype may have attributes whose types are Primitive or enumeration (example: IEC61970::Domain::ActivePower)
  - Compound may have attributes whose types are Primitive, enumeration, Datatype or Compound (example: IEC61968::Common::StreetAddress)
  - Never use association class (too exotic UML feature, poor tool support)

UML Element Conventions

- Attributes
  - Attribute names start with lower case and should not end with 's‘
  - Document the purpose of the attribute
  - Change default multiplicity - all attributes in CIM are optional
  - Set default Private visibility to Public - all attributes in CIM are public
  - When choosing type, ensure you select it from the list of types. Attention on Primitives: use CIM Primitive datatypes (e.g., String, Boolean), not default UML ones (string, boolean)
  - Inherent dependency between the attribute's type and the class owning the attribute is uni-directional: Class knows about the attribute's type, but not vice versa

UML Element Conventions

- Associations
  - Stereotyped classes never participate in association
  - Important: draw the association from your extension class towards a standard CIM class and never vice versa
  - Create an association between two NON-stereotyped classes, with unspecified direction, and with NO name and NO doc
  - Ensure you specify multiplicity for both association ends ([1], [0..1], [1..*] or [0..*])
  - Ensure both ends have navigability 'unspecified‘
  - Give names starting with upper case to both association ends; if multiplicity ends with * (i.e., more than one), append 's' to that association end name

UML Element Conventions

- Associations (Continued)
  - Document the purpose of each association end (avoid "A is related to one or more B" - UML shows it already)
  - For WG14: avoid aggregation (we currently have none); for WG13: aggregation has special meaning in the context of naming hierarchy for PowerSystemResources
  - Inherent dependency between the two classes participating in association is bi-directional: Both classes know about each other
  - Minimize associations from extension classes to high-level standard CIM classes - try to add association between your extension classes that derive from standard CIM classes

UML Element Conventions

- Inheritance
  - Stereotyped classes never participate in inheritance
  - CIM does not allow multiple inheritance
  - Existing standard CIM classes can be used as supertypes for your extension classes, never vice versa
  - Inheritance is the strongest possible dependency; it is often misused and should be used with care (note: everything that can be expressed through inheritance could also be expressed through composition)

Profile (aka Contextual Profile) Definition

- The Profile defines the data required to meet the objective defined in a Use Case or by the End User of the Profile.
- It defines the model “requirements” for the Use Case.
- A Profile is a proper subset of the CIM UML model.
- As a subset, it may not contain any classes, attributes or associations that are not already part of the CIM UML model.
- If extensions are added to the CIM UML model, these may also be contained in the Profile.
- A Profile may be expressed in Word, HTML, RDFS, XSD, UML and may be provided in all the above formats

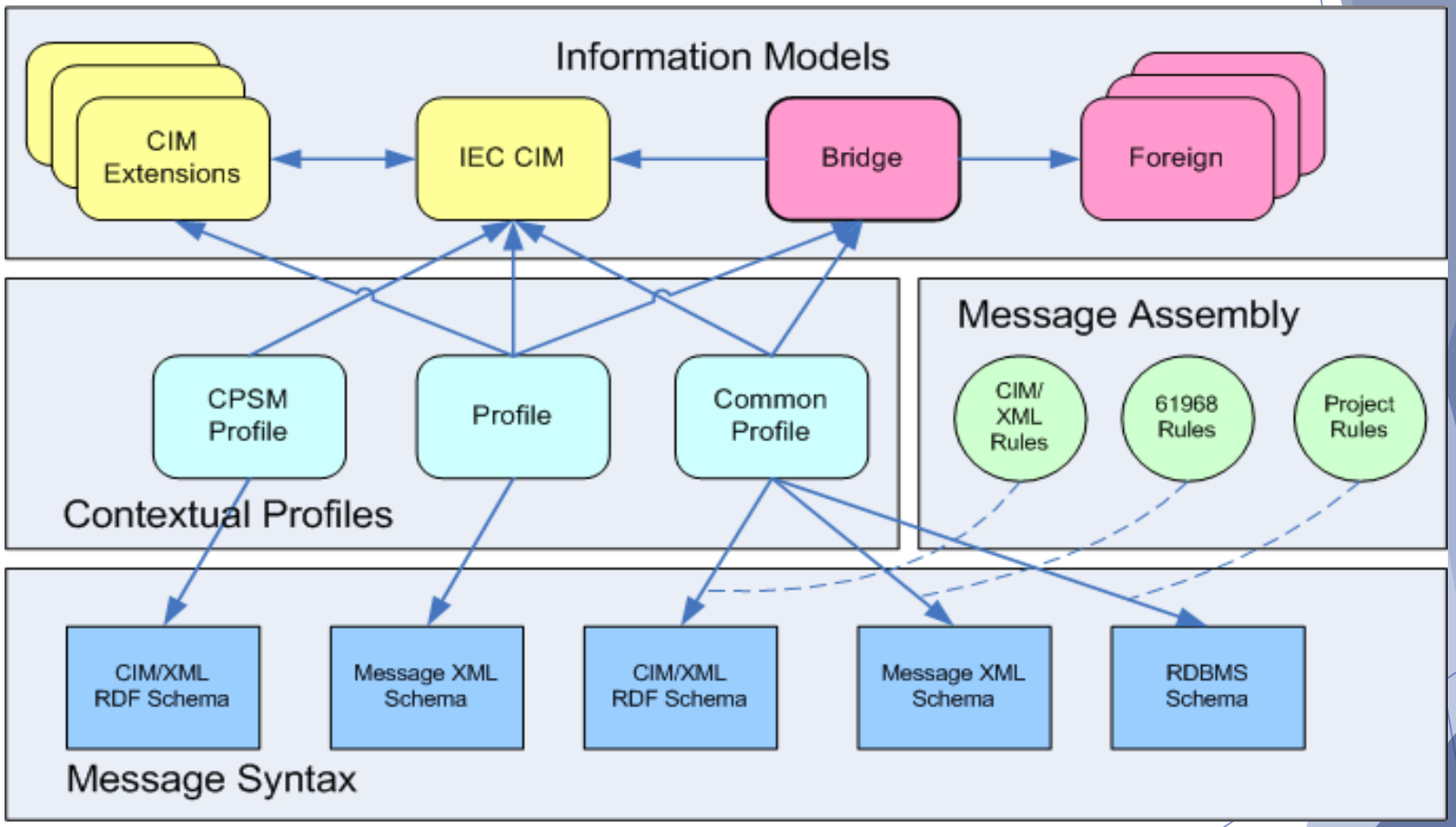
PCItek
TECHNOLOGY - EXCELLENCE - KNOWLEDGE

Contextual Profiles

- If a Profile is submitted as a standard to the IEC, it must be in Word format using the IEC template.
- A profile is often given a name (e.g. CPSM)
- Each profile will have an assigned namespace
- Profiles are also known as ‘contextual models’

PCItek
TECHNOLOGY - EXCELLENCE - KNOWLEDGE

Information Models and Profiles



PCItek
TECHNOLOGY - EXCELLENCE - KNOWLEDGE

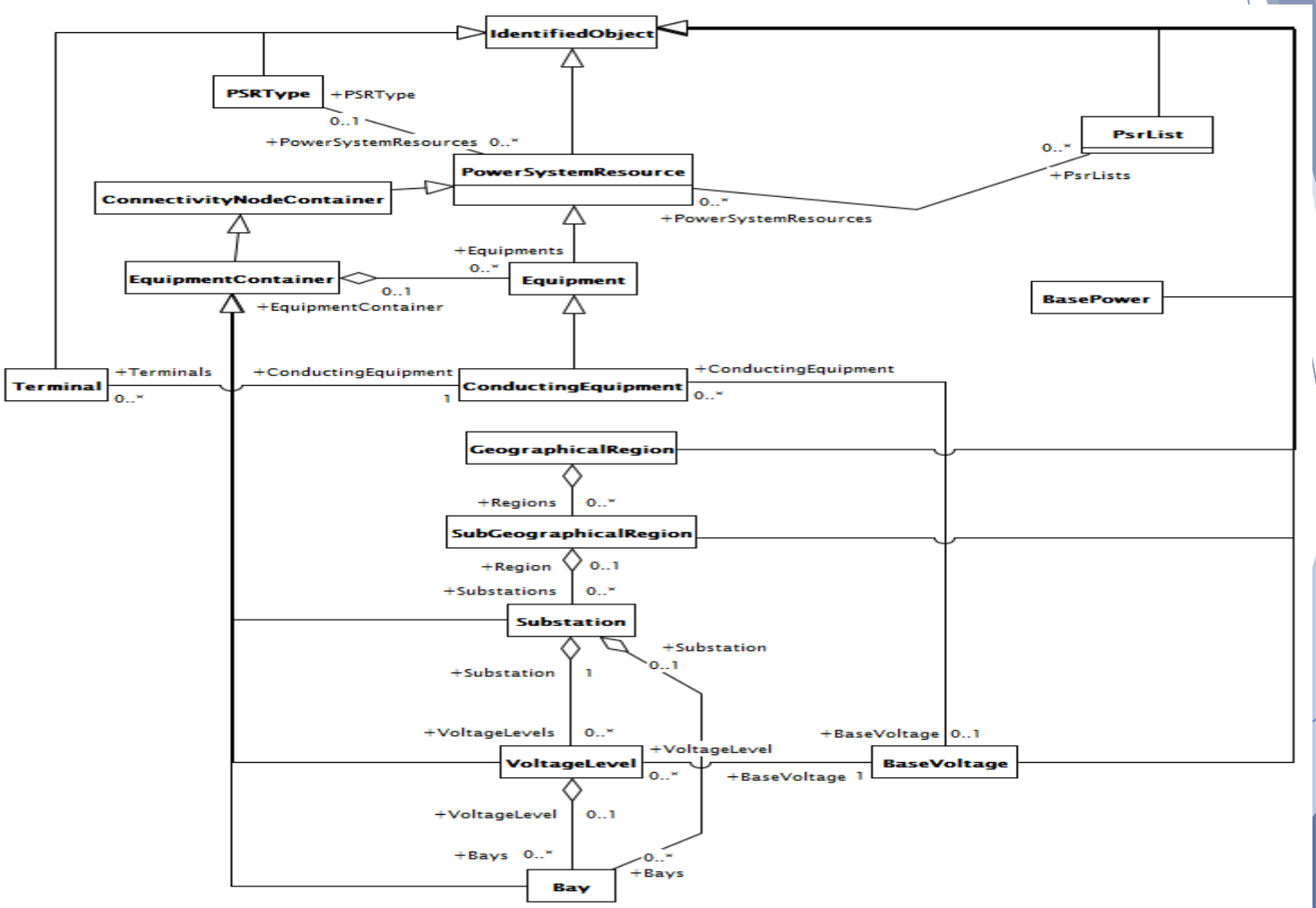
Profile Definition

- A Profile may restrict the CIM but not expand it:
  - Multiplicity can be changed from 1..n to 1..1 but not the other way.
  - The set of Attributes in a class in the CIM do not have to be contained in the Profile; that is, is a class has 10 attributes but you only need 4 of those Attributes to satisfy your goal, you only need to include those 4 attributes in the Profile
  - Not all classes in the CIM must be included in the Profile
  - Entire Packages may be left out of the Profile if they are not needed.

Thanks!

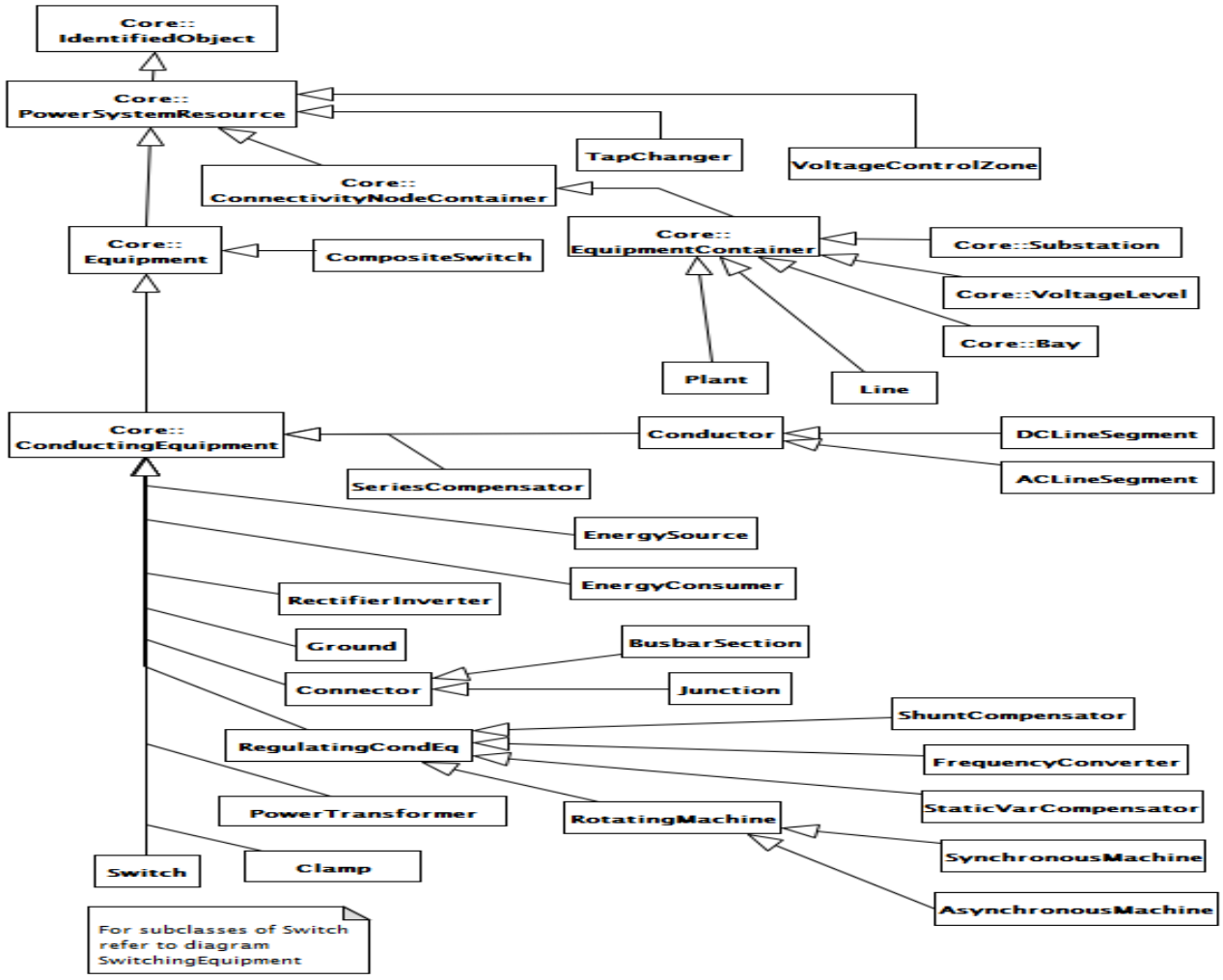
Margaret E Goodrich
Project Consultants, LLC
margaret@pcitek.com
+1-903-477-7176

Core Diagram



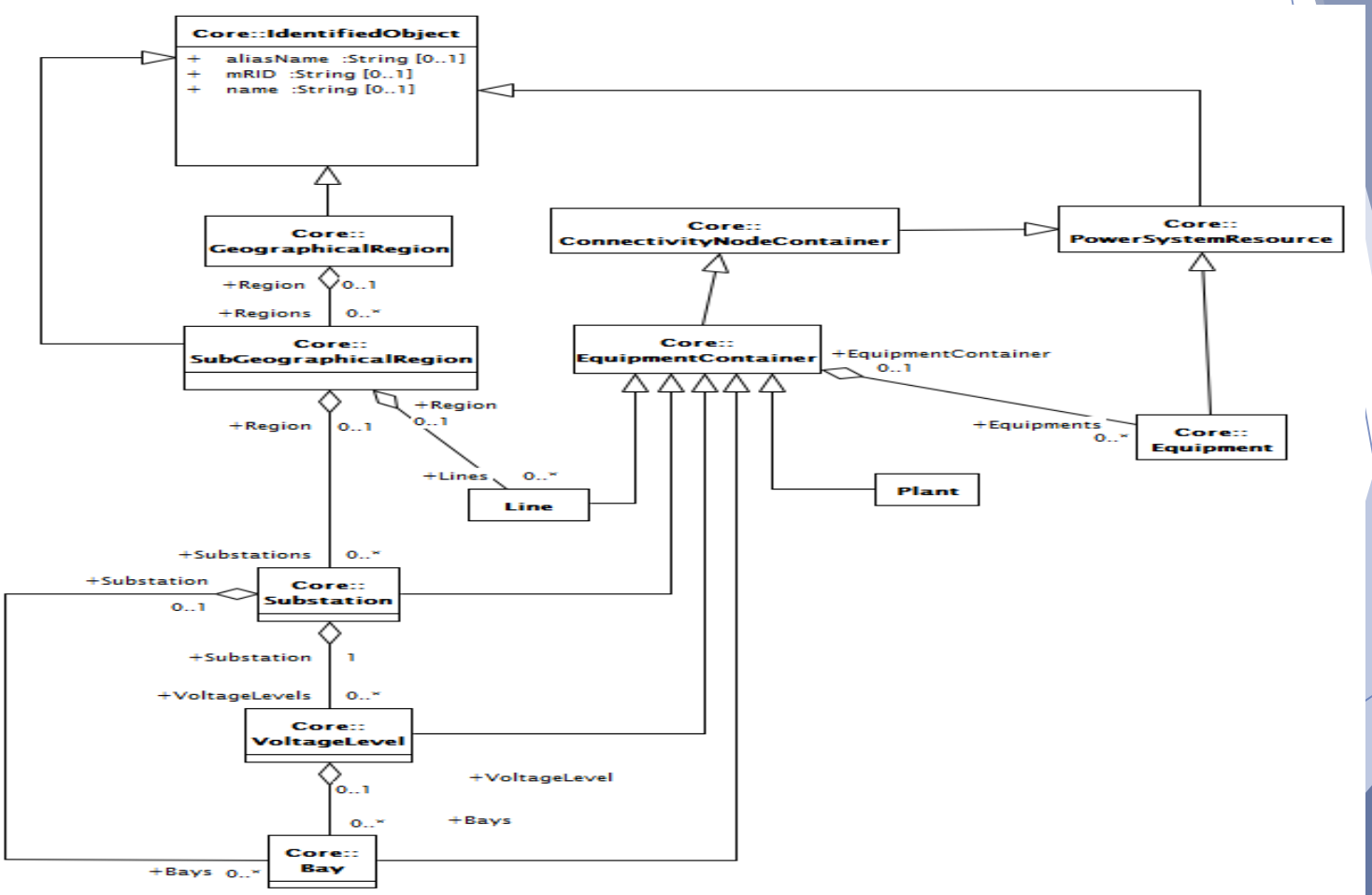
PCItek
TECHNOLOGY - EXCELLENCE - KNOWLEDGE

Inheritance Hierarchy



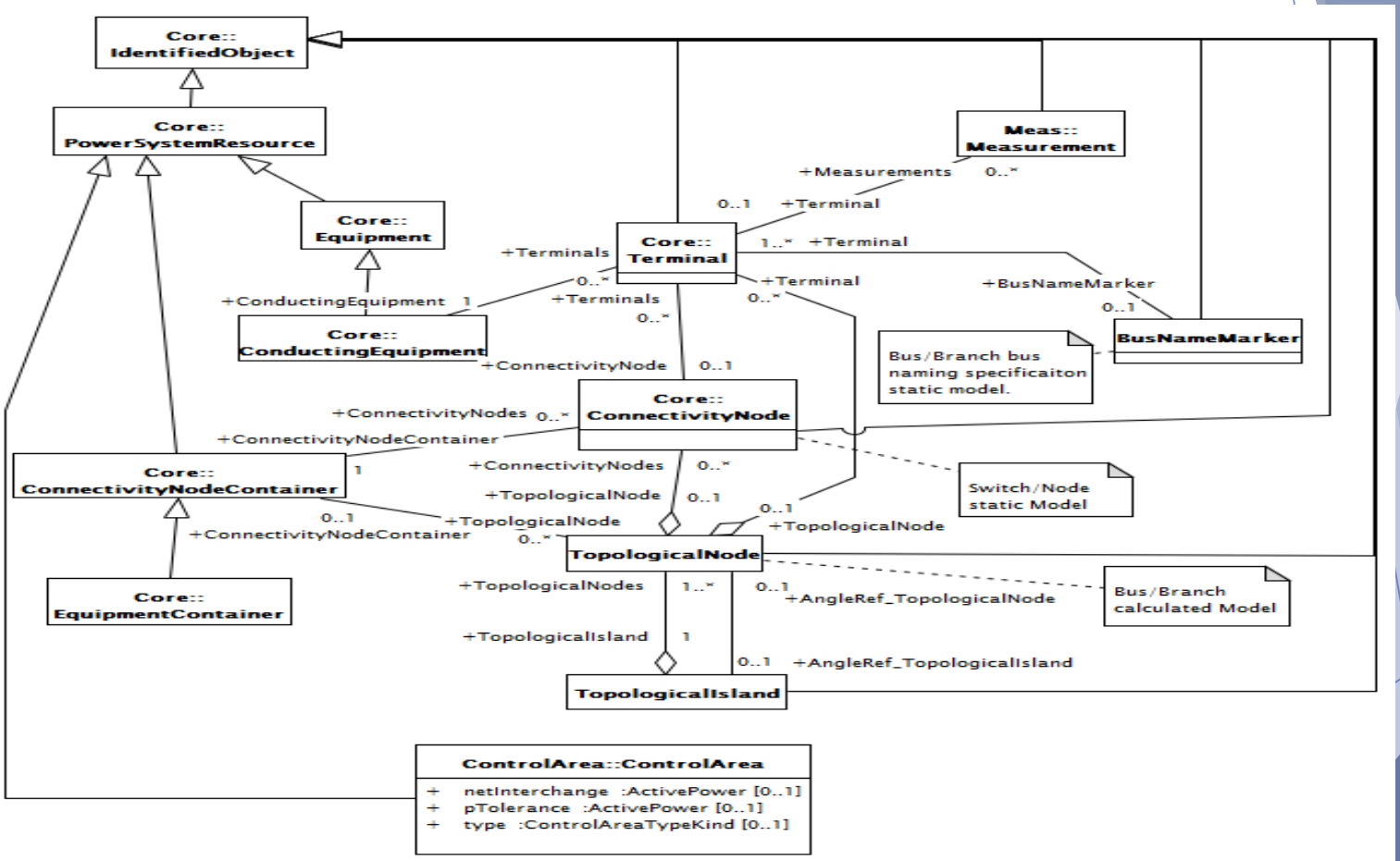
PCItek
TECHNOLOGY - EXCELLENCE - KNOWLEDGE

Equipment Containers (Containment)



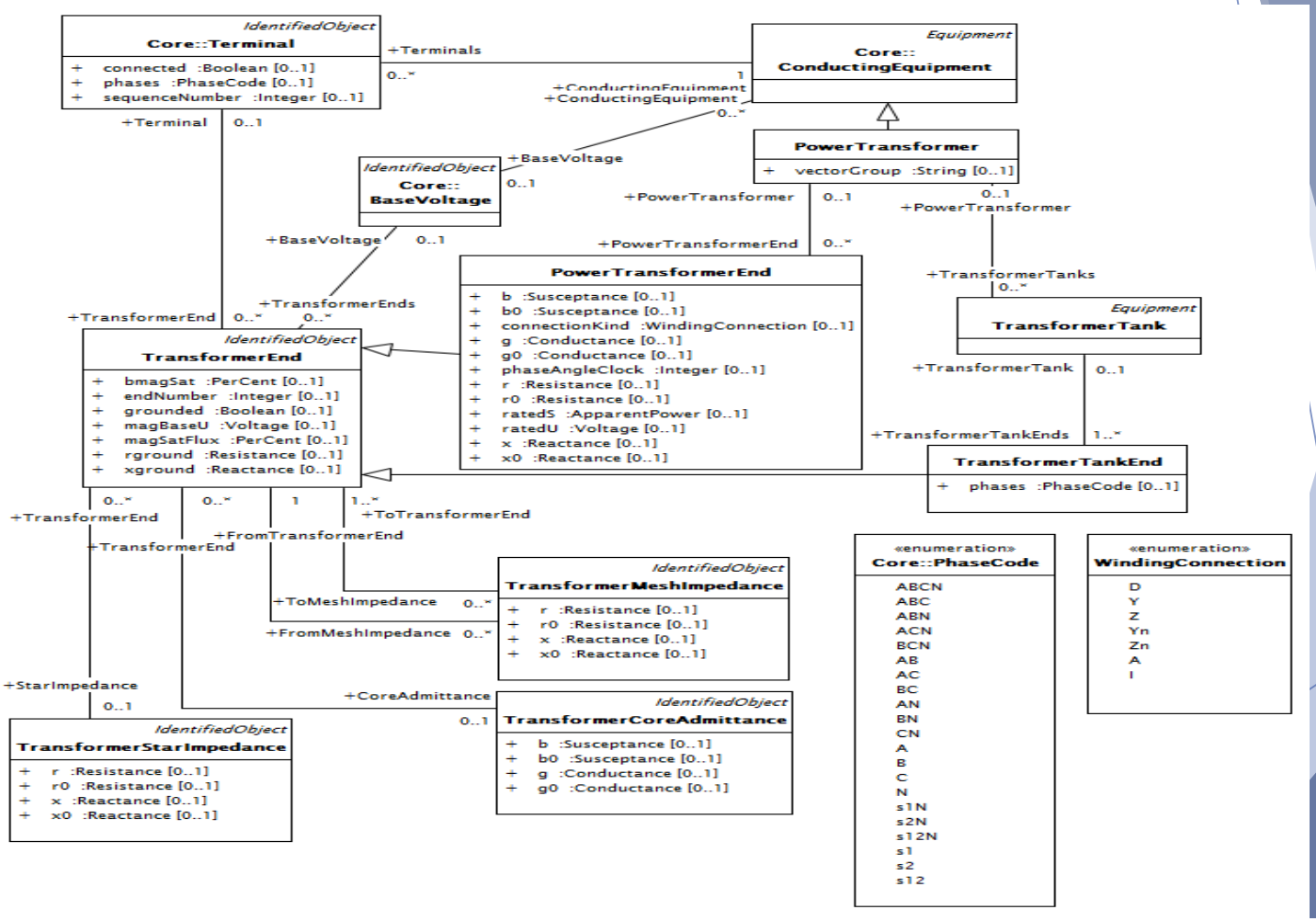
PCItek
TECHNOLOGY - EXCELLENCE - KNOWLEDGE

Connectivity Model



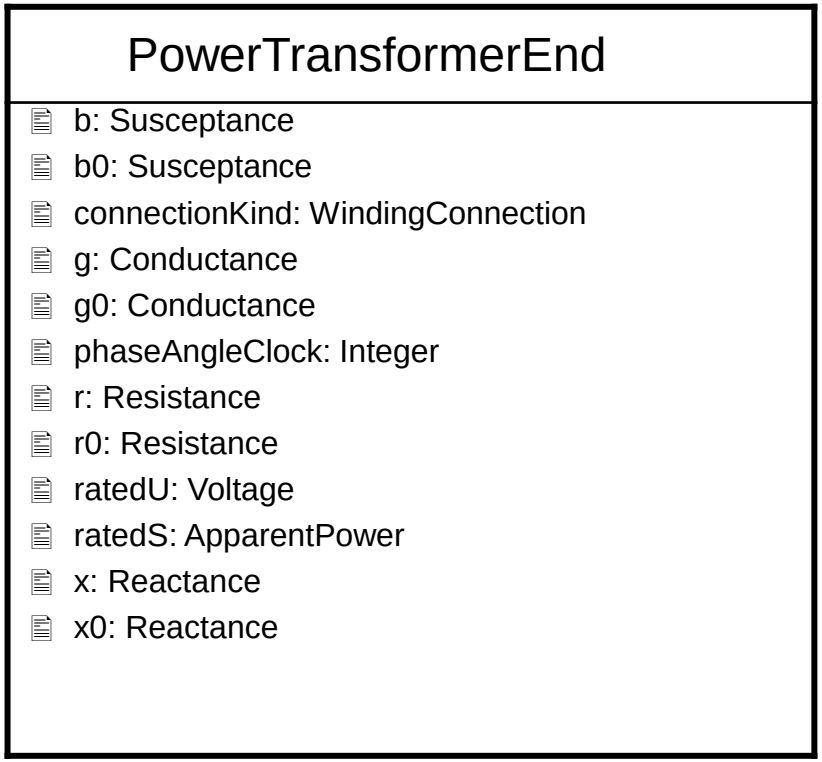
PCItek
TECHNOLOGY - EXCELLENCE - KNOWLEDGE

Transformer Class (Model) From CIM



PCItek
TECHNOLOGY - EXCELLENCE - KNOWLEDGE

PowerTransformerEnd Attributes



PCItek
TECHNOLOGY - EXCELLENCE - KNOWLEDGE