PowerSystemDataModel

Release 1.0.1-SNAPSHOT

Sep 14, 2020

Contents:

Welcome to the documentation of the PowerSystemDataModel. It provides an extensive data model capable of modelling energy systems with high granularity e.g. for bottom-up simulations. Additionally, useful functions to process, augment and furnish model i/o information is provided. Effective handling of geographic information related to power grids is also possible.

CHAPTER 1

Getting started

Welcome, dear fellow of bottom up power system modelling! This section is meant to give you some help getting hands on our project. If you feel, something is missing, please contact us!

1.1 Requirements

Java > v 1.8

1.2 Where to get

Checkout latest from GitHub or use maven for dependency management:

1.2.1 Stable releases

On Maven central:

```
<dependency>
  <groupId>com.github.ie3-institute</groupId>
  <artifactId>PowerSystemDataModel</artifactId>
  <version>1.1.0</version>
</dependency>
```

1.2.2 Snapshot releases

Available on OSS Sonatype. Add the correct repository:

```xml
<repositories>
  <repository>http://oss.sonatype.org/content/repositories/snapshots</repository>
</repositories>
```

and add the dependency:

```xml
<dependency>
  <groupId>com.github.ie3-institute</groupId>
  <artifactId>PowerSystemDataModel</artifactId>
  <version>2.0-SNAPSHOT</version>
</dependency>
```

CHAPTER 2

Available models

This page gives an overview about all available models in *PowerSystemDataModel*. They are basically grouped into three groups:

1. *Input* models may be used to describe input data for a power system simulation
2. *Result* models denote results of such a simulation
3. *Time Series* may serve both as input or output

All those models are designed with some assumptions and goals in mind. To assist you in applying them as intended, we will give you some general remarks:

Uniqueness All models have a `uuid` field as universal unique identifier. There shouldn't be any two elements with the same `uuid` in your grid data set, better in your whole collection of data sets.

Immutability We designed the models in a way, that does not allow for adaptions of the represented data after instantiation of the objects. Thereby you can be sure, that your models are *thread-safe* and no unwanted or unobserved changes are made.

Copyable With the general design principle of immutability, entity modifications (e.g. updates of field values) can become hard and annoying. To avoid generating methods to update each field value, we provide an adapted version of the builder pattern to make entity modifications as easy as possible. Each entity holds it's own copy builder class, which follows the same inheritance as the entity class itself. With a call of *.copy()* on an entity instance a builder instance is returned, that allows for modification of fields and can be terminated with *.build()* which will return an instance of the entity with modified field values as required. For the moment, this pattern is only implemented for a small amount of *AssetInput* entities (all entities held by a *GridContainer* except thermal units to be precise), but we plan to extend this capability to all input entities in the future.

Single Point of Truth Throughout all models you can be sure, that no information is given twice, reducing the possibility to have ambiguous information in your simulation set up. "Missing" information can be received through the grids relational information - e.g. if you intend to model a wind energy converter in detail, you may find information of it's geographical location in the model of it's common coupling point (*node*).

Harmonized Units System As our models are representations of physical elements, we introduced a harmonized system of units. The standard units, the models are served with, is given on each element's page. Thereby you can be sure, that all information are treated the same. As most (database) sources do not support physical units, make sure, you have your input data transferred to correct units before. Same applies for interpreting the obtained results. In all models physical values are transferred to standard units on instantiation.

2.1 Input

Model classes you can use to describe a data set as input to power system simulations.

2.1.1 Operator

This is a simple identifier object, representing a natural or legal person that is the owner or responsible person having control over one or more physical entitites.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	
id	–	Human readable identifier

Application example

```
OperatorInput profBroccoli = new OperatorInput(
        UUID.fromString("f15105c4-a2de-4ab8-a621-4bc98e372d92"),
        "Univ.-Prof. Dr. rer. hort. Klaus-Dieter Brokkoli"
      )
```

Caveats

Nothing - at least not known. If you found something, please contact us!

2.1.2 Grid Related Models

Node

Representation of an electrical node, with no further distinction into bus bar, auxiliary node or others.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	
id	–	Human readable identifier
operator	–	
operationTime	–	Timely restriction of operation
vTarget	p.u.	Target voltage magnitude to be used by voltage regulation entities
slack	–	Boolean indicator, if this nodes serves as a slack node in power flow calculation
geoPosition	–	Geographical location
voltLvl	–	Information of the voltage level (id and nominal voltage)
subnet	–	Sub grid number

Caveats

System participants, that need to have geographical locations, inherit the position from the node. If the overall location does not play a big role, you are able to use the default location with `NodeInput#DEFAULT_GEO_POSITION` being located on TU Dortmund university's campus (See on OpenStreetMaps).

Schematic Node Graphic

Schematic drawing information for a node model.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	
graphicLayer	–	Human readable identifier of the graphic layer to draw this element on
path	–	Line string of coordinates describing the drawing, e.g. for bus bars
point	–	Alternative to line string, only drawing a point coordinate
node	–	Reference to the physical node model

Caveats

Nothing - at least not known. If you found something, please contact us!

Line

Representation of an AC line.

Attributes, Units and Remarks

Type Model

Attribute	Unit	Remarks
uuid	–	
id	–	Human readable identifier
r	Ω / km	Phase resistance per length
x	Ω / km	Phase reactance per length
g	µS / km	Phase-to-ground conductance per length
b	µS / km	Phase-to-ground susceptance per length
iMax	A	Maximum permissible current
vRated	kV	Rated voltage

Entity Model

Attribute	Unit	Remarks
uuid	–	
id	–	Human readable identifier
operator	–	
operationTime	–	Timely restriction of operation
nodeA	–	
nodeB	–	
parallelDevices	–	Amount of parallel devices of same attributes
type	–	
length	km	
geoPosition	–	Line string of geographical locations describing the position of the line
olmCharacteristic	–	Characteristic of possible overhead line monitoring Can be given in the form of *olm:{<List of Pairs>}*. The pairs are wind velocity in x and permissible loading in y.

Caveats

Nothing - at least not known. If you found something, please contact us!

Schematic Line Graphic

Schematic drawing information for a line model.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	
graphicLayer	–	Human readable identifier of the graphic layer to draw this element on
path	–	Line string of coordinates describing the drawing
line	–	Reference to the physical line model

Caveats

Nothing - at least not known. If you found something, please contact us!

Switch

Model of an ideal switch connecting two node models of the same voltage level

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	
id	–	Human readable identifier
operator	–	
operationTime	–	Timely restriction of operation
nodeA	–	
nodeB	–	
closed	–	true, if the switch is closed

Caveats

Nothing - at least not known. If you found something, please contact us!

Two Winding Transformer

Model of a two winding transformer. It is assumed, that node A is the node with higher voltage.

Attributes, Units and Remarks

Type Model

All impedances and admittances are given with respect to the higher voltage side. As obvious, the parameter can be used in T- as in -equivalent circuit representations.

Attribute	Unit	Remarks
uuid		
id		Human readable identifier
rSc	Ω	Short circuit resistance
xSc	Ω	Short circuit impedance
gM	nS	No load conductance
bM	nS	No load susceptance
sRated	kVA	Rated apparent power
vRatedA	kV	Rated voltage at higher voltage terminal
vRatedB	kV	Rated voltage at lower voltage terminal
dV	%	Voltage magnitude increase per tap position
dPhi	°	Voltage angle increase per tap position
tapSide		true, if tap changer is installed on lower voltage side
tapNeutr		Neutral tap position
tapMin		Minimum tap position
tapMax		Maximum tap position

Entity Model

Attribute	Unit	Remarks
uuid	–	
id	–	Human readable identifier
operator	–	
operationTime	–	Timely restriction of operation
nodeA	–	Higher voltage node
nodeB	–	Lower voltage node
parallelDevices	–	Amount of parallel devices of same attributes
type	–	
tapPos	–	Current position of the tap changer
autoTap	–	true, if there is a tap regulation apparent and active

Caveats

Nothing - at least not known. If you found something, please contact us!

Three Winding Transformer

Model of a three winding transformer. It is assumed, that node A is the node with highest, node B with intermediate and node C with lowest voltage.

The assumed mathematical model is inspired by *ABB Schaltanlagenhanbuch* [Gremmel1999], but with the addition of a central phase-to-ground admittance, cf. following picture.

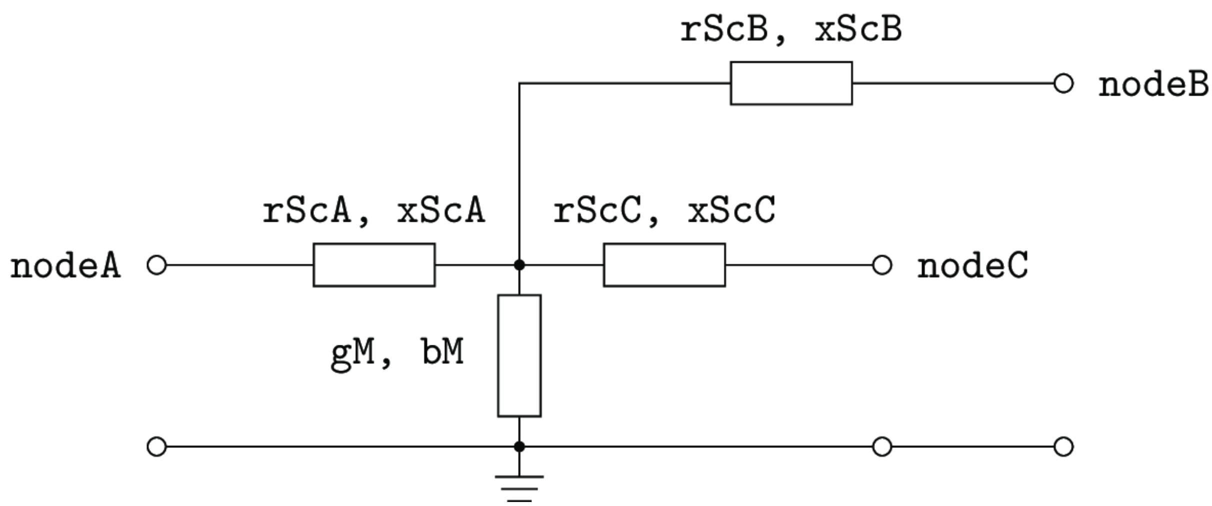


Fig. 1: "Star like" T-equivalent circuit diagram of a three winding transformer

Attributes, Units and Remarks

Type Model

All impedances and admittances are given with respect to the higher voltage side.

Attribute	Unit	Remarks
uuid		
id		Human readable identifier
rScA	Ω	Short circuit resistance in branch A
rScB	Ω	Short circuit resistance in branch B
rScC	Ω	Short circuit resistance in branch C
xScA	Ω	Short circuit impedance in branch A
xScB	Ω	Short circuit impedance in branch B
xScC	Ω	Short circuit impedance in branch C
gM	nS	No load conductance
bM	nS	No load susceptance
sRatedA	kVA	Rated apparent power of branch A
sRatedB	kVA	Rated apparent power of branch B
sRatedC	kVA	Rated apparent power of branch C
vRatedA	kV	Rated voltage at higher node A
vRatedB	kV	Rated voltage at higher node B
vRatedC	kV	Rated voltage at higher node C
dV	%	Voltage magnitude increase per tap position
dPhi	°	Voltage angle increase per tap position
tapNeutr		Neutral tap position
tapMin		Minimum tap position
tapMax		Maximum tap position

Entity Model

Attribute	Unit	Remarks
uuid	–	
id	–	Human readable identifier
operator	–	
operationTime	–	Timely restriction of operation
nodeA	–	Higher voltage node
nodeB	–	Intermediate voltage node
nodeC	–	Lowest voltage node
parallelDevices	–	Amount of parallel devices of same attributes
type	–	
tapPos	–	Current position of the tap changer
autoTap	–	true, if there is a tap regulation apparent and active

Caveats

Nothing - at least not known. If you found something, please contact us!

Measurement Unit

Representation of a measurement unit placed at a node. It can be used to mark restrictive access to simulation results to e.g. control algorithms. The measured information are indicated by boolean fields.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	
id	–	Human readable identifier
operator	–	
operationTime	–	Timely restriction of operation
node	–	
vMag	–	Voltage magnitude measurements are available
vAng	–	Voltage angle measurements are available
p	–	Active power measurements are available
q	–	Reactive power measurements are available

Caveats

Nothing - at least not known. If you found something, please contact us!

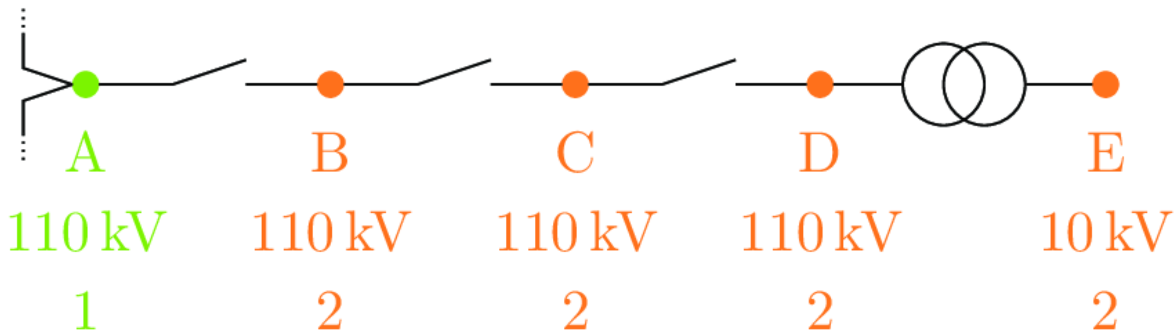
Grid Container

The grid container groups all entities that are able to form a full grid model. Two types of grid containers are available:

JointGridContainer This one is able to hold a grid model spanning several voltage levels. On instantiation, a sub grid topology graph is built. This graph holds `SubGridContainers` as vertices and transformer models as edges. Thereby, you are able to discover the topology of galvanically separated sub grids and access those sub models directly.

and

SubGridContainer This one is meant to hold all models, that form a galvanically separated sub grid. In contrast to the `JointGridContainer` it only covers one voltage level and therefore has an additional field for the predominant voltage level apparent in the container. Why predominant? As of convention, the `SubGridContainers` hold also reference to the transformers leading to higher sub grids and their higher voltage coupling point.



Let's shed a more detailed light on the boundaries of a sub grid as of our definition. This especially is important, if the switchgear of the transformer is modeled in detail. We defined, that all nodes in upstream direction of the transformer, that are connected by switches *only* (therefore are within the switchgear) are counted towards the inferior sub grid structure (here "2"), although they belong to a different voltage level. This decision is taken, because we assume, that the interest to operate on the given switchgear will most likely be placed in the inferior grid structure.

The "real" coupling node A is not comprised in the sub grids node collection, but obviously has reference through the switch between nodes A and B.

A synoptic overview of both classes' attributes is given here:

Attributes, Units and Remarks

Attribute	Unit	Remarks
gridName	–	Human readable identifier
rawGrid	–	see below
systemParticipants	–	see below
graphics	–	see below
subGridTopologyGraph	–	topology of sub grids - only `JointGridContainer`
predominantVoltageLevel	–	main voltage level - only `SubGridContainer`
subnet	–	sub grid number - only `SubGridContainer`

RawGridElements

This sub container simply holds:

- *nodes*
- *lines*

- *switches*
- *two winding transformers*
- *three winding transformers*
- *measurement units*

SystemParticipants

This sub container simply holds:

- *biomass plants*
- *combined heat and power plants*
- *electric vehicles*
- *electric vehicle charging stations*
- *fixed feed in facilities*
- *heat pumps*
- *loads*
- *photovoltaic power plants*
- *electrical energy storages*
- *wind energy converters*

and the needed nested thermal models.

Graphics

This sub container simply holds:

- *schematic node graphics*
- *schematic line graphics*

Caveats

Nothing - at least not known. If you found something, please contact us!

2.1.3 Participant Related Models

General Remarks on Participant Models

Reactive Power Characteristics

Reactive power characteristics are designed to describe reactive power control behaviour of the models. In Germany, system operators can require system participants to follow certain characteristics specified in the operators technical requirements and individually selected per connected asset.

Currently three different characteristics are implemented:

Fixed Power Factor

Active and reactive power are coupled by a time-independent power factor. It can be parsed from `cosPhiFixed:{(0.0, 0.95)}` (exemplary).

Active Power Dependent Power Factor

The power factor is determined based on the current active power feed in or consumption. The characteristic in the figure below would be described by the three coordinates (0.0, 1.0), (0.9,1.0) and (1.0, 0.95). Alternatively it can be parsed from `cosPhiP:{(0.0, 1.0),(0.9,1.0),(1.0, 0.95)}`.

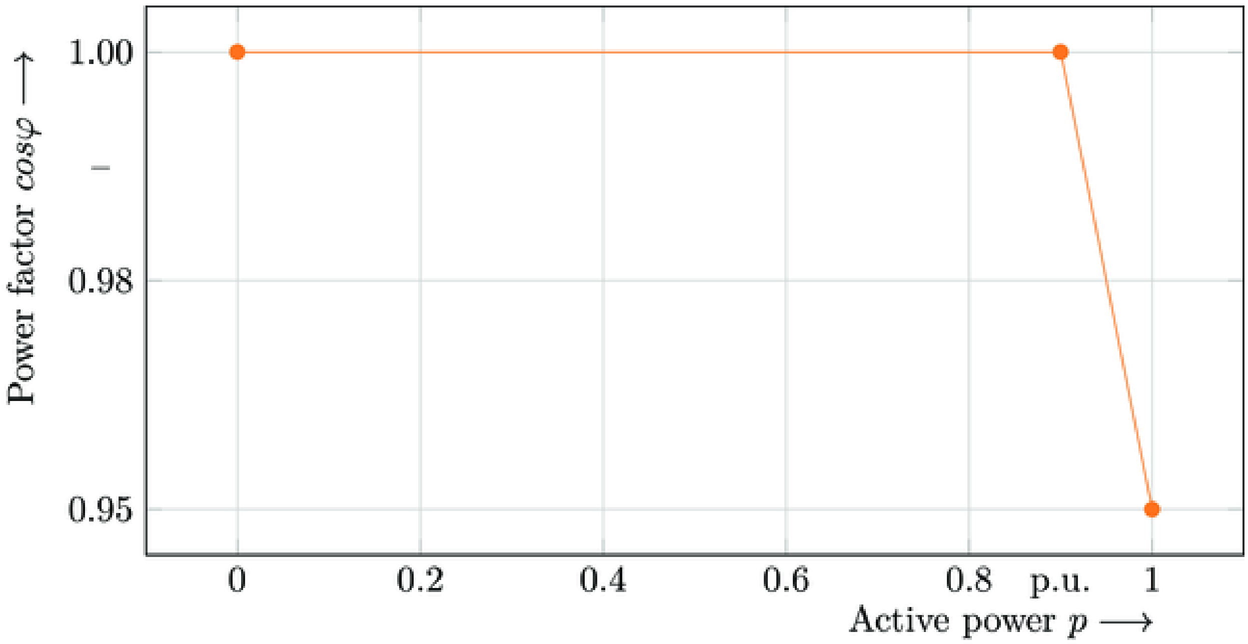


Fig. 2: Exemplary active power dependent power factor

Reactive Power as Function of Nodal Voltage Magnitude

The reactive power is directly derived in accordance to the nodal voltage magnitude. The characteristic in the figure below would be described by the three coordinates (0.92, -1), (0.97, 0.0), (1.03, 0.0) and (1.08, 1.0). Alternatively it can be parsed from `qV:{(0.92, -1),(0.97, 0.0),(1.03, 0.0),(1.08, 1.0)}`.

Biomass plant

Model of a biomass power plant.

Attributes, Units and Remarks

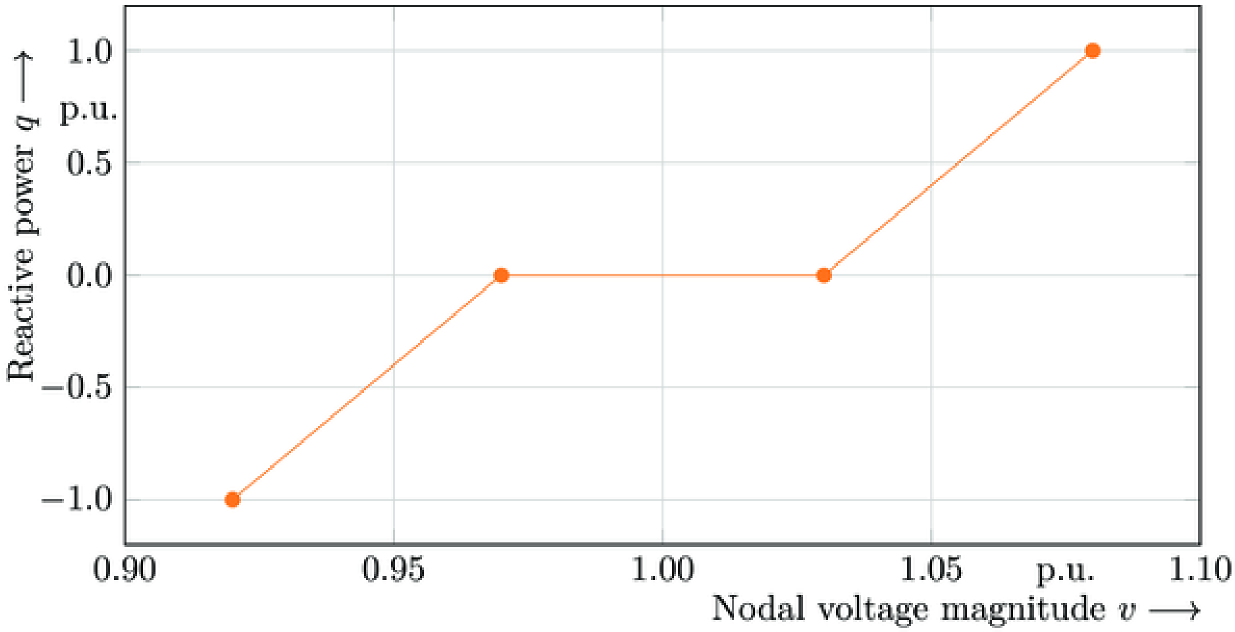


Fig. 3: Exemplary reactive power as function of nodal voltage magnitude

Type Model

Attribute	Unit	Remarks
uuid	–	
id	–	Human readable identifier
capex	€	Capital expenditure to purchase one entity of this type
opex	€ / MWh	Operational expenditure to operate one entity of this type
activePowerGradient	% / h	Maximum permissible rate of change of power
sRated	kVA	Rated apparent power
cosphiRated	–	Rated power factor
etaConv	%	Efficiency of the assets inverter

Entity Model

Attribute	Unit	Remarks
uuid	–	
id	–	Human readable identifier
operator	–	
operationTime	–	Timely restriction of operation
node	–	
qCharacteristics	–	*Reactive power characteristic* to follow
type	–	
marketReaction	–	Whether to adapt output based on (volatile) market price or not
costControlled	–	Whether to adapt output based on the difference between production costs and fixed feed in tariff or not
feedInTariff	€ / MWh	Fixed feed in tariff

Caveats

Nothing - at least not known. If you found something, please contact us!

Combined Heat and Power Plant

Combined heat and power plant.

Attributes, Units and Remarks

Type Model

Attribute	Unit	Remarks
uuid	–	
id	–	Human readable identifier
capex	€	Capital expenditure to purchase one entity of this type
opex	€ / MWh	Operational expenditure to operate one entity of this type
etaEl	%	Efficiency of the electrical inverter
etaThermal	%	Thermal efficiency of the system
sRated	kVA	Rated apparent power
cosphiRated	–	Rated power factor
pThermal	kW	Rated thermal power (at rated electrical power)
pOwn	kW	Needed self-consumption

Entity Model

Attribute	Unit	Remarks
uuid	–	
id	–	Human readable identifier
operator	–	
operationTime	–	Timely restriction of operation
node	–	
thermalBus	–	Connection point to the thermal system
qCharacteristics	–	*Reactive power characteristic* to follow
type	–	
thermalStorage	–	Reference to thermal storage
marketReaction	–	Whether to adapt output based on (volatile) market price or not

Caveats

Nothing - at least not known. If you found something, please contact us!

Electric Vehicle

Model of an electric vehicle, that is occasionally connected to the grid via an *electric vehicle charging system*.

Attributes, Units and Remarks

Type Model

Attribute	Unit	Remarks
uuid	–	
id	–	Human readable identifier
capex	€	Capital expenditure to purchase one entity of this type
opex	€ / MWh	Operational expenditure to operate one entity of this type
eStorage	kWh	Available battery capacity
eCons	kWh / km	Energy consumption per driven kilometre
sRated	kVA	Rated apparent power
cosphiRated	–	Rated power factor

Entity Model

Attribute	Unit	Remarks
uuid	–	
id	–	Human readable identifier
operator	–	
operationTime	–	Timely restriction of operation
node	–	
type	–	

Caveats

The `node` attribute only marks the vehicles home connection point. The actual connection to the grid is always given through `EvcsInput`'s relation.

Electric Vehicle Charging System

This model is currently only a dummy implementation.

Fixed Feed In Facility

Model of a facility, that provides constant power feed in, as no further information about the actual behaviour of the model can be derived.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	
id	–	Human readable identifier
operator	–	
operationTime	–	Timely restriction of operation
node	–	
qCharacteristics	–	*Reactive power characteristic* to follow
sRated	kVA	Rated apparent power
cosphiRated	–	Rated power factor

Caveats

Nothing - at least not known. If you found something, please contact us!

Heat Pump

Model of a heat pump.

Attributes, Units and Remarks

Type Model

Attribute	Unit	Remarks
uuid	–	
id	–	Human readable identifier
capex	€	Capital expenditure to purchase one entity of this type
opex	€ / MWh	Operational expenditure to operate one entity of this type
sRated	kVA	Rated apparent power
cosphiRated	–	Rated power factor
pThermal	kW	Rated thermal power (at rated electrical power)

Entity Model

Attribute	Unit	Remarks
uuid	–	
id	–	Human readable identifier
operator	–	
operationTime	–	Timely restriction of operation
node	–	
thermalBus	–	Connection point to the thermal system
qCharacteristics	–	*Reactive power characteristic* to follow
type	–	

Caveats

Nothing - at least not known. If you found something, please contact us!

Load

Model of (mainly) domestic loads.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	
id	–	Human readable identifier
operator	–	
operationTime	–	Timely restriction of operation
node	–	
qCharacteristics	–	*Reactive power characteristic* to follow
standardLoadProfile	–	*Standard load profile* as model behaviour
dsm	–	Whether the load is able to follow demand side management signals
eConsAnnual	kWh	Annual energy consumption
sRated	kVA	Rated apparent power
cosphiRated	–	Rated power factor

Caveats

Nothing - at least not known. If you found something, please contact us!

Standard Load Profiles

The `StandardLoadProfile` is an interface, that forces it's implementing classes to have a `String` *key* and being able to parse a `String` to an `StandardLoadProfile`. Its only purpose is to give note, which standard load profile has to be used by the simulation. The actual profile has to be provided by the simulation itself. If no matching standard load profile is known, `StandardLoadProfile#NO_STANDARD_LOAD_PROFILE` can be used.

To assist the user in marking the desired load profile, the enum `BdewLoadProfile` provides a collection of commonly known German standard electricity load profiles, defined by the bdew (Bundesverband der Energie- und Wasserwirtschaft; engl. Federal Association of the Energy and Water Industry). For more details see the corresponding website (German only).

Photovoltaic Power Plant

Detailed model of a photovoltaic power plant.

Attribute	Unit	Remarks
uuid	–	
id	–	Human readable identifier
operator	–	
operationTime	–	Timely restriction of operation
node	–	
qCharacteristics	–	*Reactive power characteristic* to follow
albedo	–	Albedo of the plant's surrounding
azimuth	°	Inclination in a compass direction South = 0°, West = 90°, East = -90°
etaConv	%	Efficiency of the assets inverter
height	°	Tilted inclination from horizontal [0°, 90°]
kG	–	Generator correction factor merging technical influences
kT	–	Temperature correction factor merging thermal influences
marketReaction	–	Whether to adapt output based on (volatile) market price or not
sRated	kVA	Rated apparent power
cosphiRated	–	Rated power factor

Caveats

Nothing - at least not known. If you found something, please contact us!

Electrical Energy Storage

Model of an ideal electrical battery energy storage.

Attributes, Units and Remarks

Type Model

Attribute	Unit	Remarks
uuid	–	
id	–	Human readable identifier
capex	€	Capital expenditure to purchase one entity of this type
opex	€ / MWh	Operational expenditure to operate one entity of this type
eStorage	kWh	Battery capacity
sRated	kVA	Rated apparent power
cosphiRated	–	Rated power factor
pMax	kW	Maximum permissible active power infeed or consumption
activePowerGradient	% / h	Maximum permissible rate of change of power
eta	%	Efficiency of the electrical inverter
dod	%	Maximum permissible depth of discharge. 80 % dod is equivalent to a state of charge of 20 %.
lifeTime	h	Permissible hours of full use
lifeCycle	–	Permissible amount of full cycles

Entity Model

Attribute	Unit	Remarks
uuid	–	
id	–	Human readable identifier
operator	–	
operationTime	–	Timely restriction of operation
node	–	
qCharacteristics	–	*Reactive power characteristic* to follow
type	–	
behaviour	–	Foreseen operation strategy of the storage. Eligible input: *"market"*, *"grid"*, *"self"*

Caveats

The field `behaviour` will be removed in version 1.x, as this is an information, that is only important to a smaller sub set of simulation applications.

Wind Energy Converter

Model of a wind energy converter.

Attributes, Units and Remarks

Type Model

Attribute	Unit	Remarks
uuid	–	
id	–	Human readable identifier
capex	€	Capital expenditure to purchase one entity of this type
opex	€ / MWh	Operational expenditure to operate one entity of this type
sRated	kVA	Rated apparent power
cosphiRated	–	Rated power factor
cpCharacteristic	–	Wind velocity dependent *Betz factors*.
etaConv	%	Efficiency of the assets inverter
rotorArea	m^2	Area the rotor covers
hubHeight	m	Height of the rotor hub

Entity Model

Attribute	Unit	Remarks
uuid	–	
id	–	Human readable identifier
operator	–	
operationTime	–	Timely restriction of operation
node	–	
qCharacteristics	–	*Reactive power characteristic* to follow
type	–	
marketReaction	–	Whether to adapt output based on (volatile) market price or not

Caveats

Nothing - at least not known. If you found something, please contact us!

Betz Characteristic

A collection of wind velocity to Betz factor pairs to be applied in Betz's law to determine the wind energy coming onto the rotor area.

Thermal Bus

A coupling point to thermal system - equivalent to *electrical node*.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	
id	–	Human readable identifier
operator	–	
operationTime	–	Timely restriction of operation
bus	–	Connection point to the thermal system

Caveats

Nothing - at least not known. If you found something, please contact us!

Thermal House Model

Model for the thermal behaviour of a building. This reflects a simple shoe box with transmission losses

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	
id	–	Human readable identifier
operator	–	
operationTime	–	Timely restriction of operation
ethLosses	kW / K	Thermal losses
ethCapa	kWh / K	Thermal capacity

Caveats

Nothing - at least not known. If you found something, please contact us!

Cylindrical Thermal Storage

Model of a cylindrical thermal storage using a fluent to store thermal energy.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	
id	–	Human readable identifier
operator	–	
operationTime	–	Timely restriction of operation
thermalBus	–	Connection point to the thermal system
storageVolumeLvl	m^3	Overall available storage volume
storageVolumeLvlMin	m^3	Minimum permissible storage volume
inletTemp	°C	Temperature of the inlet
returnTemp	°C	Temperature of the outlet
c	$kWh / (K \cdot m^3)$	Specific heat capacity of the storage medium

Caveats

Nothing - at least not known. If you found something, please contact us!

2.2 Result

Model classes you can use to describe the outcome of a power system simulation.

2.2.1 Grid Related Models

Node

Representation of an electrical node, with no further distinction into bus bar, auxiliary node or others.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	uuid for the result entity
timeStamp	ZonedDateTime	date and time for the produced result
inputModel	–	uuid for the associated input model
vMag	p.u.	
vAng	degree	

Caveats

Nothing - at least not known. If you found something, please contact us!

Connector

Representation of all kinds of connectors.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	uuid for the result entity
timeStamp	ZonedDateTime	date and time for the produced result
inputModel	–	uuid for the associated input model
iAMag	ampere	A stands for sending node
iAAng	degree	
iBMag	ampere	B stands for receiving node
iBAng	degree	

Caveats

Groups all available connectors i.e. lines, switches and transformers

Line

Representation of an AC line.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	uuid for the result entity
timeStamp	ZonedDateTime	date and time for the produced result
inputModel	–	uuid for the associated input model
iAMag	ampere	A stands for sending node
iAAng	degree	
iBMag	ampere	B stands for receiving node
iBAng	degree	

Caveats

Nothing - at least not known. If you found something, please contact us!

Switch

Representation of electrical switches.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	uuid for the result entity
timeStamp	ZonedDateTime	date and time for the produced result
inputModel	–	uuid for the associated input model
iAMag	ampere	A stands for sending node
iAAng	degree	
iBMag	ampere	B stands for receiving node
iBAng	degree	
closed	boolean	status of the switching device

Caveats

Nothing - at least not known. If you found something, please contact us!

Transformer

Representation of transformers.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	uuid for the result entity
timeStamp	ZonedDateTime	date and time for the produced result
inputModel	–	uuid for the associated input model
iAMag	ampere	A stands for sending node
iAAng	degree	
iBMag	ampere	B stands for receiving node
iBAng	degree	
tapPos	–	

Caveats

Groups common information to both 2W and 3W transformers.

Two Winding Transformer

Representation of two winding transformers.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	uuid for the result entity
timeStamp	ZonedDateTime	date and time for the produced result
inputModel	–	uuid for the associated input model
iAMag	ampere	A stands for sending node
iAAng	degree	
iBMag	ampere	B stands for receiving node
iBAng	degree	
tapPos	–	

Caveats

Assumption: Node A is the node at higher voltage.

Three Winding Transformer

Representation of three winding transformers.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	uuid for the result entity
timeStamp	ZonedDateTime	date and time for the produced result
inputModel	–	uuid for the associated input model
iAMag	ampere	A stands for sending node
iAAng	degree	
iBMag	ampere	B stands for receiving node
iBAng	degree	
iCMag	ampere	B stands for receiving node
iCAng	degree	
tapPos	–	

Caveats

Assumption: Node A is the node at highest voltage and Node B is at intermediate voltage. For model specifications please check corresponding input model documentation.

2.2.2 Participant Related Models

Biomass plant

Result of a biomass power plant.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	uuid for the result entity
timeStamp	–	date and time for the produced result
inputModel	–	uuid for the associated input model
p	MW	
q	MVAr	

Caveats

Nothing - at least not known. If you found something, please contact us!

Combined Heat and Power Plant

Result of a combined heat and power plant.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	uuid for the result entity
timeStamp	–	date and time for the produced result
inputModel	–	uuid for the associated input model
p	MW	
q	MVAr	

Caveats

Nothing - at least not known. If you found something, please contact us!

Electric Vehicle

Result of an electric vehicle, that is occasionally connected to the grid via an *electric vehicle charging station*.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	uuid for the result entity
timeStamp	–	date and time for the produced result
inputModel	–	uuid for the associated input model
p	MW	
q	MVAr	
soc	–	

Caveats

Nothing - at least not known. If you found something, please contact us!

Electric Vehicle Charging Station

This model is currently only a dummy implementation of an electric vehicle charging station.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	uuid for the result entity
timeStamp	–	date and time for the produced result
inputModel	–	uuid for the associated input model
p	MW	
q	MVAr	

Caveats

Nothing - at least not known. If you found something, please contact us!

Fixed Feed In Facility

Result of a facility, that provides constant power feed in, as no further information about the actual behaviour of the model can be derived.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	uuid for the result entity
timeStamp	–	date and time for the produced result
inputModel	–	uuid for the associated input model
p	MW	
q	MVAr	

Caveats

Nothing - at least not known. If you found something, please contact us!

Load

Result of a heat pump.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	uuid for the result entity
timeStamp	–	date and time for the produced result
inputModel	–	uuid for the associated input model
p	MW	
q	MVAr	
qDot	MW	Thermal power

Caveats

Nothing - at least not known. If you found something, please contact us!

Load

Result of (mainly) domestic loads.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	uuid for the result entity
timeStamp	–	date and time for the produced result
inputModel	–	uuid for the associated input model
p	MW	
q	MVAr	

Caveats

Nothing - at least not known. If you found something, please contact us!

Photovoltaic Power Plant

Result of a photovoltaic power plant.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	uuid for the result entity
timeStamp	–	date and time for the produced result
inputModel	–	uuid for the associated input model
p	MW	
q	MVAr	

Caveats

Nothing - at least not known. If you found something, please contact us!

Electrical Energy Storage

Result of an electrochemical storage

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	uuid for the result entity
timeStamp	–	date and time for the produced result
inputModel	–	uuid for the associated input model
p	MW	
q	MVAr	
soc	–	

Caveats

Nothing - at least not known. If you found something, please contact us!

Wind Energy Converter

Result of a wind turbine.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	uuid for the result entity
timeStamp	–	date and time for the produced result
inputModel	–	uuid for the associated input model
p	MW	
q	MVAr	

Caveats

Nothing - at least not known. If you found something, please contact us!

Thermal Sink

Result of a thermal sink.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	uuid for the result entity
timeStamp	–	date and time for the produced result
inputModel	–	uuid for the associated input model
qDot	MW	thermal heat demand

Caveats

Nothing - at least not known. If you found something, please contact us!

Thermal Storage

Result of a thermal storage.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	uuid for the result entity
timeStamp	–	date and time for the produced result
inputModel	–	uuid for the associated input model
energy	MWh	
qDot	MW	heat flowing in

Caveats

Nothing - at least not known. If you found something, please contact us!

Thermal Unit

Result of a thermal unit.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	uuid for the result entity
timeStamp	–	date and time for the produced result
inputModel	–	uuid for the associated input model
qDot	MW	thermal power exchanged

Caveats

Nothing - at least not known. If you found something, please contact us!

Thermal House

Model for the thermal behaviour of a building. This reflects a simple shoe box with transmission losses

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	uuid for the result entity
timeStamp	–	date and time for the produced result
inputModel	–	uuid for the associated input model
qDot	MW	thermal heat demand of the sink
indoorTemperature	°C	

Caveats

Nothing - at least not known. If you found something, please contact us!

Cylindrical Thermal Storage

Result of a cylindrical thermal storage using a fluent to store thermal energy.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	uuid for the result entity
timeStamp	–	date and time for the produced result
inputModel	–	uuid for the associated input model
energy	MWh	
qDot	MW	heat demand of the sink
fillLevel	–	

Caveats

Nothing - at least not known. If you found something, please contact us!

System Participant

Groups together all system participants such as PV, Storage etc.

Attributes, Units and Remarks

Attribute	Unit	Remarks
uuid	–	uuid for the result entity
timeStamp	–	date and time for the produced result
inputModel	–	uuid for the associated input model
p	MW	
q	MVAr	

Caveats

Nothing - at least not known. If you found something, please contact us!

2.3 Time Series

Time series are meant to represent a timely ordered series of values. Those can either be electrical or non-electrical depending on what one may need for power system simulations. Our time series models are divided into two subtypes:

Individual Time Series Each time instance in this time series has its own value (random duplicates may occur obviously). They are only applicable for the time frame that is defined by the content of the time series.

Repetitive Time Series Those time series do have repetitive values, e.g. each day or at any other period. Therefore, they can be applied to any time frame, as the mapping from time instant to value is made by information reduction. In addition to actual data, a mapping function has to be known.

To be as flexible, as possible, the actual content of the time series is given as children of the `Value` class. The following different values are available:

Value Class	Purpose
`PValue`	Electrical active power
`SValue`	Electrical active and reactive power
`HeatAndPValue`	Combination of thermal power (e.g. in kW) and electrical active power (e.g. in kW)
`HeatAndSValue`	Combination of thermal power (e.g. in kW) and electrical active and reactive power (e.g. in kW and kVAr)
`EnergyPriceValue`	Wholesale market price (e.g. in € / MWh)
`IrradiationValue`	Combination of diffuse and direct solar irradiation
`TemperatureValue`	Temperature information
`WindValue`	Combination of wind direction and wind velocity
`WeatherValue`	Combination of irradiation, temperature and wind information

CHAPTER 3

I/O

The PowerSystemDataModel library additionally offers I/O-capabilities. In the long run, it is our aim to provide many different source and sink technologies. Therefore, the I/O-package is structured as highly modular.

3.1 Implemented Data Connectors

- influxdb
- csvfiles

3.2 De-Serialization (loading models)

At the end, having an instance of *Grid Container* is the goal. It consists of the three main blocks:

1. *Raw grid elements*
2. *System participants*
3. *Graphics*

Those blocks are also reflected in the structure of data source interface definitions. There is one source for each of the containers, respectively.

As a full data set has references among the models (e.g. a line model points to its' nodes it connects), there is a hierarchical structure, in which models have to be loaded. Therefore, the different sources have also references among themselves. An application example to load an *exampleGrid* from csv files located in `./exampleGrid` could look like this:

```
/* Parameterization */
String csvSep = ";"
String folderPath = "./exampleGrid"
FileNamingStrategy namingStrategy = new FileNamingStrategy() // Default naming␣
↪strategy
```

(continues on next page)

io/_static/figures/uml/DataSourceClassDiagramm.png

Fig. 1: Class diagram of data sources

(continued from previous page)

```
/* Instantiating sources */
TypeSource typeSource = new CsvTypeSource(csvSep, folderPath, namingStrategy)
RawGridSource rawGridSource = new CsvRawGridSource(csvSep, folderPath, namingStrategy,
↪ typeSource)
ThermalSource thermalSource = new CsvThermalSource(csvSep, folderPath, namingStrategy,
↪ typeSource)
ParticipantSource participantSource = new CsvSystemParticipantSource(
  csvSep,
  folderPath,
  namingStrategy,
  typeSource,
  thermalSource,
  rawGridSource
)
GraphicSource graphicsSource = new CsvGraphicSource(
  csvSep,
  folderPath,
  namingStrategy,
  typeSource,
  rawGridSource
)

/* Loading models */
RawGridElements rawGridElements = rawGridSource.getGridData.orElseThrow(
      () -> new SourceException("Error during reading of raw grid data.")
   )
SystemParticipants systemParticipants = systemParticipantSource.getGridData.
↪orElseThrow(
      () -> new SourceException("Error during reading of raw grid data.")
   )
GraphicElements graphicElements = graphicsSource.getGraphicElements.orElseThrow(
      () -> new SourceException("Error during reading of graphic elements data.")
   )
JointGridContainer fullGrid = new JointGridContainer(
  gridName,
  rawGridElements,
  systemParticipants,
  graphicElements
)
```

As observable from the code, it doesn't play a role, where the different parts come from. It is also a valid solution, to receive types from file, but participants and raw grid elements from a data base. Only prerequisite: An implementation of the different interfaces for the desired data sink.

3.3 Serialization (writing models)

Serializing models is a bit easier:

```
/* Parameterization */
String csvSep = ";"
String folderPath = "./exampleGrid"
FileNamingStrategy namingStrategy = new FileNamingStrategy()
boolean initEmptyFiles = false
```

(continues on next page)

(continued from previous page)

```
/* Instantiating the sink */
DataSink sink = new CsvFileSink(folderPath, namingStrategy, initEmptyFiles, csvSep)
sink.persistAll(grid.allEntitiesAsList())
```

The sink takes a collection of model suitable for serialization and handles the rest (e.g. unboxing of nested models) on its own. But caveat: As the (csv) writers are implemented in a concurrent, non-blocking way, duplicates of nested models could occur.

3.4 Default naming strategy

There is a default mapping from model class to file naming in the case you would like to use csv files for (de) serialization of models. You may extend / alter the naming with pre- or suffix by calling `new FileNamingStrategy("prefix", "suffix")`.

3.4.1 Input

Model	File Name
operator	*prefix*_operator_input_*suffix*
node	*prefix*_node_input_*suffix*
line	*prefix*_line_input_*suffix* *prefix*_line_type_input_*suffix*
switch	*prefix*_switch_input_*suffix*
two winding transformer	*prefix*_transformer2w_input_*suffix* *prefix*_transformer2w_type_input_*suffix*
three winding transformer	*prefix*_transformer3w_input_*suffix* *prefix*_transformer3w_type_input_*suffix*
measurement unit	*prefix*_measurement_unit_input_*suffix*
biomass plant	*prefix*_bm_input_*suffix* *prefix*_bm_type_input_*suffix*
combined heat and power plant	*prefix*_chp_input_*suffix* *prefix*_chp_type_input_*suffix*
electric vehicle	*prefix*_ev_input_*suffix* *prefix*_ev_type_input_*suffix*
electric vehicle charging station	*prefix*_evcs_input_*suffix*
fixed feed in facility	*prefix*_fixed_feed_in_input_*suffix*
heat pump	*prefix*_hp_input_*suffix* *prefix*_hp_type_input_*suffix*
load	*prefix*_load_input_*suffix*
photovoltaic power plant	*prefix*_pc_input_*suffix*
electrical energy storage	*prefix*_storage_input_*suffix* *prefix*_storage_type_input_*suffix*
wind energy converter	*prefix*_wec_input_*suffix* *prefix*_wec_type_input_*suffix*
schematic node graphic	*prefix*_node_graphic_input_*suffix*
schematic line graphic	*prefix*_line_graphic_input_*suffix*

3.4.2 Results

Model	File Name
node	*prefix*_node_res_*suffix*
line	*prefix*_line_res_*suffix*
switch	*prefix*_switch_res_*suffix*
two winding transformer	*prefix*_transformer2w_res_*suffix*
three winding transformer | *prefix*_transformer3w_res_*suffix*	
biomass plant	*prefix*_bm_res_*suffix*
combined heat and power plant	*prefix*_chp_res_*suffix*
electric vehicle	*prefix*_ev_res_*suffix*
electric vehicle charging station	*prefix*_evcs_res_*suffix*
fixed feed in	*prefix*_fixed_feed_in_res_*suffix*
heat pump	*prefix*_hp_res_*suffix*
load	*prefix*_load_res_*suffix*
photovoltaic power plant	*prefix*_pv_res_*suffix*
storage	*prefix*_storage_res_*suffix*
wind energy converter	*prefix*_wec_res_*suffix*
thermal house model	*prefix*_thermal_house_res_*suffix*
cylindrical thermal storage	*prefix*_cylindrical_storage_res_*suffix*

3.4.3 Time Series

Model	File Name
individual time series	*prefix*_individual_time_series_*UUID*_*suffix*
load profile input	*prefix*_load_profile_time_series_*profileKey*_*UUID*_*suffix*

CHAPTER 4

Contact the (Main) Maintainers

If you feel, something this missing, wrong or misleading, please contact one of our main contributors:

- @sensarmad
- @johanneshiry
- @ckittl

Hat tip to all other contributors!

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

Bibliography

[Gremmel1999] Gremmel, H., Ed., Schaltanlagen. Cornelsen Verlag, 1999, Vol. 10, isbn: 3-464-48235-9.